nomencl[*]
A Package to Create a Nomenclature

Boris Veytsman[†]
Bernd Schandl[‡]
Lee Netherton & CV Radhakrishnan[§]

http://sarovar.org/projects/nomencl

Printed on September 22, 2005

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 nomencl 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.

1.1 Important Notes for Users of Previous Versions

The latest update to the nomencl package has included some major changes to some of the more important commands. In particular, the `\makeglossary` and `\printglossary` commands have now been renamed to `\makenomenclature` and `\printnomenclature` respectively. The reason for this change is to increase the package's compatibility with other MakeIndex using packages. With this increased compatibility, users will be able to have nomenclatures, glossaries and indexes all in one document.

[*] Package version v4.2 of 2005/09/22.

[†] Up to version v2.2 (1996/11/25)

[‡] Up to version v3.1c (2001/10/02)

[§] Up to version v4.0 (2005/03/31)

There is a compatibility option that will allow you to still use your `\makeglossary` and `\printglossary` commands (see section 2.3), but it is advised that you change your `\...glossary` commands to the new `\...nomenclature` commands in your LaTeX files. For more information on the compatibility mode see section 6.

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*⟩`]{nomencl}` in the preamble of your document.

`\makenomenclature`

- Put `\makenomenclature` 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.

`\printnomenclature`

- Put `\printnomenclature` at the place you want to have your nomenclature list.

Now put your file through LaTeX. The command `\makenomenclature` will instruct LaTeX to open the nomenclature file ⟨*filename*⟩`.nlo` corresponding to your LaTeX file ⟨*filename*⟩`.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 ⟨*filename*⟩`.nlo` as your input file, use `nomencl.ist` as your style file[1], and write output to the file ⟨*filename*⟩`.nls`. How to do this depends on your implementation of *MakeIndex*. For most UNIX implementations you should write something like

```
makeindex ⟨filename⟩.nlo -s nomencl.ist -o ⟨filename⟩.nls
```

[1] German 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*.

Now you have the file ⟨*filename*⟩`.nls` that contains your nomenclature list properly ordered. The last step is to invoke LaTeX on your master file ⟨*filename*⟩`.tex` once more. It will input your `.nls` 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[`⟨*prefix*⟩`]{`⟨*symbol*⟩`}{`⟨*description*⟩`}`

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

Put this command immediately after the equation or text that introduces ⟨*symbol*⟩. 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 ⟨*symbol*⟩ in `$` signs.

Let's have a look at a simple example. If your input file looks like the one in Figure 1 then your nomenclature[2] 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 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:

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

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

```
\documentclass{article}
\usepackage{nomencl}
\makenomenclature
\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.
\printnomenclature
\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

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.

intoc Inserts the nomenclature in the Table of Contents.

notintoc No entry for the nomenclature in the Table of Contents. (default)

compatible Run in compatibility mode. Older tex files may need this option selected to be able to compile. In the latest version of nomencl the commands `\makeglossary` and `\printglossary` were replaced with `\makenomenclature` and `\printnomenclature`. Selecting this option will redefine the old commands, but will loose the compatibility with other glossary packages.

noncompatible Do not run in compatibility mode. (default)

croatian, danish, english, french, german, italian, polish, portuguese, 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. **x̊indy**. Please help me out with other languages. (default: english)

2.4 Referencing

\nomrefeq
\nomrefpage
\nomrefeqpage
\nomnorefeq
\nomnorefpage
\nomnorefeqpage

As explained in Section 2.3, you can turn referencing to equations and pages on/off globally using the package options. But sometimes you might want to change the referencing behavior for single entries. The following six macros can be used inside a `\nomenclature` macro: `\nomrefeq`, `\nomnorefeq`, `\nomrefpage`, `\nomnorefpage`, `\nomrefeqpage`, `\nomnorefeqpage`. The first four work similarly to the package options, only local to the entry; the last two are shortcuts, so saying `\nomrefeqpage` is equivalent to `\nomrefeq\nomrefpage`.

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

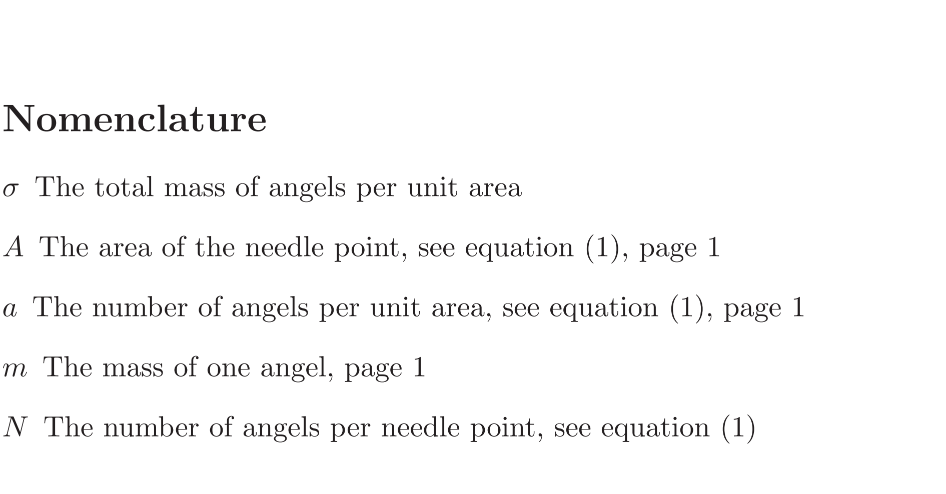
```
\begin{equation}
  a=\frac{N}{A}
\end{equation}%
\nomenclature{$a$}{The number of angels per unit area\nomrefeqpage}%
\nomenclature{$N$}{The number of angels per needle point\nomrefeq}%
\nomenclature{$A$}{The area of the needle point\nomrefeq\nomrefpage}%
The equation $\sigma = m a$%
\nomenclature{$\sigma$}{The total mass of angels per unit area}%
\nomenclature{$m$}{The mass of one angel\nomrefpage}
follows easily.
\printnomenclature
\end{document}
```

Figure 3: Input 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.



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

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, strings[4]. For the groups the following algorithm[5] 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<~`).[6] If two non-letters are compared (which can not happen

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

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

[6] An 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?

at the first position of a string), ASCII code is used (`+<1<:<\`). Additionally there is the issue of word ordering (treat spaces as letters 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.

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

`\printnomenclature` Probably the most common change to the nomenclature is a different

`\nomlabelwidth` 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 `\printnomenclature`. If you want to have a little more space for the labels (and you don't live in a metric world) you can use

```
\printnomenclature[0.5in]
```

instead of the simple

```
\printnomenclature
```

`thenomenclature` If you don't like the format of the nomenclature at all, you will have to redefine the `thenomenclature` 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
\printnomenclature
```

I thought about putting this in the definition of `\printnomenclature` 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 `nomencl.cfg` that executes these three lines all at once and can be used instead of `\printnomenclature`.

Putting an entry for the nomenclature in the table of contents can be done by adding an **intoc** to the package options.

`\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 nomencl 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]{%
```

[7] I hope you did read it ;-)

```
\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` `\nompostamble` Maybe you want to explain something just between the title of the 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 `nomencl.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.

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

```
\renewcommand{\nomentryend}{.}
```

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

`\eqdeclaration` `\pagedeclaration` If you don’t like the text that is used for the references to equations and pages, you can define `\eqdeclaration` and `\pagedeclaration`. Both should accept one argument, namely the equation and page number, respectively. 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 nomencl package.

5 Tips and Tricks

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

For most examples, sample configuration files will be generated if you run LaTeX on the file `nomencl.ins`. There will for example be a file `sample01.cfg` for the subgroups example in Section 5.1. Rename it to `nomencl.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.

```
%<*sample01>
\RequirePackage{ifthen}
\renewcommand{\nomgroup}[1]{%
  \ifthenelse{\equal{#1}{R}}{\item[\textbf{Variables}]}{%
    \ifthenelse{\equal{#1}{G}}{\item[\textbf{Constants}]}{}}}
⟨/sample01⟩
```

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

```
%<*sample02>
\newcommand{\nomunit}[1]{%
\renewcommand{\nomentryend}{\hspace*{\fill}#1}}
⟨/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\@nomenclaturefile{}%
  {\string\nomenclatureentry{#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\thenomenclature{%
  \@ifundefined{chapter}{\section*}{\chapter*}{\nomname}%
  \nompreamble
  \begin{longtable}[l]{@{}ll@{}}}
\def\endthenomenclature{%
  \end{longtable}%
  \nompostamble}
```

Finally we add the following two lines at the end of nomencl.ist[8].

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

[8] Don'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 8.

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 nomencl 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 `\makenomenclature` macro has to be changed.

```
%<*sample04>
\def\makenomenclature{%
  \newwrite\@nomenclaturefile
  \immediate\openout\@nomenclaturefile=\jobname\@outputfileextension
  \def\@nomenclature{%
    \@ifnextchar[%
      {\@@@@nomenclature}{\@@@@nomenclature[\nomprefix]}}%
  \typeout{Writing nomenclature file \jobname\@outputfileextension}%
  \let\makenomenclature\@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.

```
\def\@@@@nomenclature[#1]#2#3{%
  \protected@write\@nomenclaturefile{}%
  {\string\nomenclatureentry{#1#2@[{#2}]%
      \begingroup#3\protect\nomeqref{\theequation}%
        |nompageref}{\thepage}}}%
⟨/sample04⟩
```

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

5.5 Glossary in “Kopka Style”

I was told that the glossary in the LaTeX book by Kopka looks roughly like in Figure 5. In order to get a glossary like this, 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.

```
%<*sample05>
\def\@@@nomenclature[#1]#2#3{%
  \def\@tempa{#2}\def\@tempb{#3}%
  \protected@write\@nomenclaturefile{}%
   {\string\nomenclatureentry{#1\nom@verb\@tempa @[{\nom@verb\@tempa}]%
     |nompageref{\begingroup\nom@verb\@tempb\protect\nomeqref{\theequation}}}%
     {\thepage}}%
  \endgroup
  \@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?

```
\def\nompageref#1#2{%
  \if@printpageref\pagedeclaration{#2}\else\null\fi
  \linebreak#1\nomentryend\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).

```
\def\pagedeclaration#1{\dotfill\nobreakspace#1}
```

```
\def\nomentryend{.}
\def\nomlabel#1{\textbf{#1}\hfil}
⟨/sample05⟩
```

6 Compatibility Mode

With previous versions of the nomencl, the commands `\makeglossary` and `\pringlossary` were used to generate and display the nomenclature. These commands have now been depreciated, and replaced with the `\makenomenclature` and `\printnomenclature` commands. The new commands do exactly the same as the old commands, but because of the name changes, the package is now compatible with other packages which use the `\makeglossary` commands. The previous versions of nomencl also used the file extensions `.glo` and `.gls` for the generated output and input files. These extensions have now been changed to `.nlo` and `.nls` respectively—again, for increased compatibility.

For all of the legacy LaTeX files out there which use the old commands there is a compatibility option available so that the old commands will still work without having to change any of the existing code. To enable the compatibility mode simply supply the **compatible** option when using the package. For example:

```
\usepackage[compatible]{nomencl}
```

Under compatibility mode, the package will generate and use files with the old-style file extensions (i.e. `.glo` and `.gls`).

It is worth noting that even though the compatibility mode is available, it is highly recommended to update your LaTeX files to use the new nomenclature commands.

7 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).

8 Releases and Legal Issues

This package can be redistributed and/or modified under the terms of the LaTeX Project Public License distributed from CTAN archives in the directory macros/latex/base/lppl.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 nomencl package can always be found at http://sarovar.org/projects/nomencl. Usually, the same version is also available at CTAN/macros/latex/contrib/supported/nomencl/.

9 Implementation

9.1 The LaTeX Package File

At the beginning of this file, the \ProvidesPackage macro was executed. So we only need to to state that we need LaTeX 2_{ε}.

```
%<*package>
\NeedsTeXFormat{LaTeX2e}
```

\if@printeqref
\if@printpageref

We need two switches to decide whether references to equations and pages should be printed.

```
\newif\if@printeqref
\newif\if@printpageref
```

\if@intoc

Another switch to decide whether to add an entry to the TOC.

```
\newif\if@intoc
```

\if@compatibilitymode

Another switch to decide whether to run in compatibility mode.

```
\newif\if@compatibilitymode
```

And the options to set these switches globally.

```
\DeclareOption{refeq}{\@printeqreftrue}
\DeclareOption{norefeq}{\@printeqreffalse}
\DeclareOption{refpage}{\@printpagereftrue}
\DeclareOption{norefpage}{\@printpagereffalse}
\DeclareOption{intoc}{\@intoctrue}
\DeclareOption{notintoc}{\@intocfalse}
\DeclareOption{compatible}{\@compatibilitymodetrue}
\DeclareOption{noncompatible}{\@compatibilitymodefalse}
```

\nomprefix

It might make sense to add the prefix “a” to every sortkey, see Section 3.

```
\DeclareOption{prefix}{\def\nomprefix{a}}
\DeclareOption{noprefix}{\def\nomprefix{}}
```

\if@loadcfg

Another switch and the corresponding options to decide whether we should look for a configuration file.

```
\newif\if@loadcfg
\DeclareOption{cfg}{\@loadcfgtrue}
\DeclareOption{nocfg}{\@loadcfgfalse}
```

\eqdeclaration
\pagedeclaration
\nomname

If you can help out with translations for some other languages, let me know.

```
\DeclareOption{croatian}{%
  \def\eqdeclaration#1{, vidi jednad\v{z}bu\nobreakspace(#1)}%
  \def\pagedeclaration#1{, stranica\nobreakspace#1}%
  \def\nomname{Popis simbola}}
\DeclareOption{danish}{%
  \def\eqdeclaration#1{, se ligning\nobreakspace(#1)}%
  \def\pagedeclaration#1{, side\nobreakspace#1}%
  \def\nomname{Symbolliste}}
\DeclareOption{english}{%
  \def\eqdeclaration#1{, see equation\nobreakspace(#1)}%
  \def\pagedeclaration#1{, page\nobreakspace#1}%
  \def\nomname{Nomenclature}}
\DeclareOption{french}{%
  \def\eqdeclaration#1{, voir \'equation\nobreakspace(#1)}%
  \def\pagedeclaration#1{, page\nobreakspace#1}%
  \def\nomname{Liste des symboles}}
\DeclareOption{german}{%
  \def\eqdeclaration#1{, siehe Gleichung\nobreakspace(#1)}%
  \def\pagedeclaration#1{, Seite\nobreakspace#1}%
  \def\nomname{Symbolverzeichnis}}
\DeclareOption{italian}{%
\def\eqdeclaration#1{, vedi equazione\nobreakspace(#1)}%
\def\pagedeclaration#1{, pagina\nobreakspace#1}%
\def\nomname{Elenco dei simboli}}
\DeclareOption{polish}{%
  \def\eqdeclaration#1{, porownaj rownanie\nobreakspace(#1)}%
  \def\pagedeclaration#1{, strona\nobreakspace#1}%
  \def\nomname{Lista symboli}}
\DeclareOption{portuguese}{%
  \def\eqdeclaration#1{, veja equa\c{c}\~ao\nobreakspace(#1)}%
  \def\pagedeclaration#1{, p\'agina\nobreakspace#1}%
  \def\nomname{Nomenclatura}}
\DeclareOption{russian}{%
  \def\eqdeclaration#1{, \cyrs\cyrm.\nobreakspace(#1)}%
  \def\pagedeclaration#1{, \cyrs\cyrt\cyrr.\nobreakspace#1}%
  \def\nomname{\CYRS\cyrp\cyri\cyrs\cyro\cyrk%
    \ \cyro\cyrb\cyro\cyrz\cyrn\cyra\cyrch\cyre\cyrn\cyri%
    \cyrishrt}}
\DeclareOption{spanish}{%
  \def\eqdeclaration#1{, v\'ease la ecuaci\'on\nobreakspace(#1)}%
  \def\pagedeclaration#1{, p\'agina\nobreakspace#1}%
  \def\nomname{Nomenclatura}}
\DeclareOption{ukrainian}{%
  \def\eqdeclaration#1{, \cyrd\cyri\cyrv.\nobreakspace(#1)}%
  \def\pagedeclaration#1{, \cyrs\cyrt\cyro\cyrr.\nobreakspace#1}%
  \def\nomname{\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk%
         \ \cyrp\cyro\cyrz\cyrn\cyra\cyrch\cyre\cyrn\cyrsftsn}}
```

Finally set the default options and process everything.

```
\ExecuteOptions{noncompatible,notintoc,norefeq,norefpage,prefix,cfg,english}
\ProcessOptions\relax
```

\@outputfileextension
\@inputfileextension

The default file extension for the output and input nomenclature files are `.nlo` and `.nls` respectively. In compatibility mode, these are changes to `.glo` and `.gls`.

```
\if@compatibilitymode%
    \def\@outputfileextension{.glo}%
    \def\@inputfileextension{.gls}%
\else%
    \def\@outputfileextension{.nlo}%
    \def\@inputfileextension{.nls}%
\fi%
```

\makenomenclature

The definition of `\makenomenclature` is pretty much the same as in the LaTeX kernel for `\makeglossary`, we only use `\@nomenclature` instead of `\glossary`.

```
\def\makenomenclature{%
  \newwrite\@nomenclaturefile
  \immediate\openout\@nomenclaturefile=\jobname\@outputfileextension
  \def\@nomenclature{%
    \@bsphack
    \begingroup
    \@sanitize
    \@ifnextchar[%
      {\@@@nomenclature}{\@@@nomenclature[\nomprefix]}}%
  \typeout{Writing nomenclature file \jobname\@outputfileextension}%
  \let\makenomenclature\@empty}
```

\makeglossary

The `\makeglossary` command has been depreciated, and is only available in compatibility mode.

```
\if@compatibilitymode\let\makeglossary\makenomenclature\fi%
```

\nom@verb The macro \nom@verb, which is copied from [4] and [5, p. 382], makes it possible to use \nomenclature in another macro.

```
\def\nom@verb{\expandafter\strip@prefix\meaning}
```

\nomenclature This macro just protects the “real” \@nomenclature macro. I am not sure whether this makes sense because you shouldn’t use \nomenclature in something like \section anyway, but it doesn’t hurt.

```
\def\nomenclature{\protect\@nomenclature}
```

\@nomenclature
\@@nomenclature Without an executed \makenomenclature, \@nomenclature will only change some catcodes and call the macro \@@nomenclature to gobble its arguments.

```
\def\@nomenclature{%
  \@bsphack
  \begingroup
  \@sanitize
  \@ifnextchar[%
    {\@@nomenclature}{\@@nomenclature[\nomprefix]}}
\def\@@nomenclature[#1]#2#3{\endgroup\@esphack}
```

\@@@nomenclature If \makenomenclature was already executed, then \@nomenclature calls the macro \@@@nomenclature which writes to the nomenclature file. It puts the prefix in front of the entry, adds brackets [] around the entry (because it will be the argument of an \item) and adds possible references at the end of the entry description. A group is started to keep changes to the reference switches local. The arguments are written using \nom@verb so they will not be expanded, even when \nomenclature is used within another macro. By the way, \@bsphack and \@esphack makes \nomenclature disappear between two spaces; unfortunately this doesn’t work if \nomenclature is the first thing in a line.

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

\nomgroup The next macro is executed between each character group in the nomenclature. By default it just gobbles its argument, but the user can redefine it to add white space or some fancy divider including the starting character of the new group.

```
\def\nomgroup#1{}
```

\nomlabelwidth

This is the default label width for the nomenclature. It can be changed e. g. in the `cfg` file.

```
\newdimen\nomlabelwidth
\nomlabelwidth1cm\relax
```

\nom@tempdim
\printnomenclature
\@printnomenclature

The optional argument is read and assigned to `\nom@tempdim`. Then the `gls` file is read.

```
\newdimen\nom@tempdim
\def\printnomenclature{%
  \@ifnextchar[%
    {\@printnomenclature}{\@printnomenclature[\nomlabelwidth]}}
\def\@printnomenclature[#1]{%
  \nom@tempdim#1\relax
  \@input@{\jobname\@inputfileextension}}
```

\printglossary

The `\printglossary` command has been depreciated, and is only available in compatibility mode.

```
\if@compatibilitymode\let\printglossary\printnomenclature\fi%
```

\nomlabel
\nompreamble
\nompostamble
\nomentryend

Now some bells and whistles to format the nomenclature: the definition of the label, the preamble, the postamble and the symbol that is added at the end of an entry. The last three are defined to do nothing by default.

```
\def\nomlabel#1{#1\hfil}
\def\nompreamble{}
\def\nompostamble{}
\def\nomentryend{}
```

\nomitemsep

The skip between two items is adjustable by changing `\nomitemsep`. It defaults to `\itemsep`.

```
\newskip\nomitemsep
\nomitemsep\itemsep
```

thenomenclature

The `thenomenclature` environment formats its title and optionally inserts an item in the TOC, both are dependant on whether the `\chapter` command is available or not. After printing the preamble, a list is started with the `\labelwidth` being set to the value defined in the optional argument of `\printnomenclature`.

```
\def\thenomenclature{%
```

```
  \@ifundefined{chapter}%
  {
    \section*{\nomname}
    \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
  }%
  {
    \chapter*{\nomname}
    \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
  }%

  \nompreamble
  \list{}{%
    \labelwidth\nom@tempdim
    \leftmargin\labelwidth
    \advance\leftmargin\labelsep
    \itemsep\nomitemsep
    \let\makelabel\nomlabel}}
\def\endthenomenclature{%
  \endlist
  \nompostamble}
```

\nomrefeq \refpage \refeqpage \norefeq \norefpage \norefeqpage

These are the switches to turn referencing on or off locally for a single entry.

```
\def\nomrefeq{\@printeqreftrue}
\def\nomrefpage{\@printpagereftrue}
\def\nomrefeqpage{\@printeqreftrue\@printpagereftrue}
\def\nomnorefeq{\@printeqreffalse}
\def\nomnorefpage{\@printpagereffalse}
\def\nomnorefeqpage{\@printeqreffalse\@printpagereffalse}
```

\nomeqref

The equation is only referenced if the corresponding switch is true. Since *MakeIndex* tends to insert a line break just before the page number, we have to add `\ignorespaces` at the end.

```
\def\nomeqref#1{\if@printeqref\eqdeclaration{#1}\fi\ignorespaces}
```

\nompageref

The page is also only referenced if requested. Then the end symbol is added and finally the group started in `\@@@nomenclature` is closed.

```
\def\nompageref#1{\if@printpageref\pagedeclaration{#1}\fi%
  \nomentryend\endgroup}
```

Read the config file if it exists and the corresponding option was given.

```
\if@loadcfg
  \InputIfFileExists{nomencl.cfg}{%
    \typeout{Using the configuration file nomencl.cfg}}{}
\fi
```

The end.

```
%</package>
```

9.2 The *MakeIndex* Style File

The “magic word” for *MakeIndex* in the input file is `\nomenclatureentry`. German user might need to redefine the quote character if they want to use `"a` instead of `\"a`. Choose whatever character you see fit except `|`, `@` and `!`.

```
%<*idxstyle>
%% ---- for input file ----
keyword    "\\nomenclatureentry"
%% Germans might want to change this and delete the two %%
%% quote '"'
```

Define what is printed at the beginning and the end of the file and the skip between groups. Since we already write `\nomgroup` between groups, we define `group_skip` to just input an empty line.

```
%% ---- for output file ----
preamble   "\\begin{thenomenclature} \n"%
postamble  "\n\n\\end{thenomenclature}\n" group_skip "\n"
```

Since we can’t handle multiple pages for an entry anyway, we also don’t need any delimiters.

```
delim_0    ""
delim_1    ""
delim_2    ""
```

Now the macro between the groups. Since the flag is positive, the character will be inserted as a capital letter. As the comment states, this will cause some warnings. If someone has a better solution, let me know.

```
%% The next lines will produce some warnings when
%% running Makeindex as they try to cover two different
%% versions of the program:
lethead_prefix "\n \\nomgroup{"
lethead_suffix "}\n"
lethead_flag   1
heading_prefix "\n \\nomgroup{"
heading_suffix "}\n"
headings_flag  1
⟨/idxstyle⟩
```

References

[1] Braams, Johannes; Carlisle, David; Jeffrey, Alan; Lamport, Leslie; Mittelbach, Frank; Rowley, Chris; Schöpf, Rainer (1996). `ltidxglo.dtx` – 1996/01/20 v1.1e LaTeX Kernel (Index and Glossary). CTAN/macros/latex/base/ltidxglo.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 TEX Archive Network CTAN. `ftp://ctan.tug.org/tex-archive/`.

[4] Jones, David M. (1995). A new implementation of LATEX's indexing commands, Version v4.1beta of 1995/09/28. CTAN/macros/latex/contrib/supported/camel/index.dtx.

[5] Knuth, Donald E. (1984). *The TEXbook*. Addison-Wesley Publishing Company, Reading, MA.

[6] Lamport, Leslie (1994). *LATEX: A Document Preparation System*. Addison-Wesley Publishing Company, Reading, MA.

[7] Veytsman, Boris (1996). Package nomencl, Version 4.0. `http://sarovar.org/projects/nomencl` (2000/09/15).

Index

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