Endnotes for \LaTeX 2ε

Clemens Niederberger

https://bitbucket.org/cgnieder/enotez/

contact@mychemistry.eu

Table of Contents

1 Licence and Requirements  5 Collect Notes Section-wise and Print List Stepwise
2 Motivation
3 Usage  6 Language Support
   3.1 Placing the Notes . . . . . .  2 7 hyperref Support
   3.2 Printing the Notes . . . . . .  3
4 Options  4 Notes
   4.1 Package Options . . . . . .  4 10 References
   4.2 Customizing the List . . . . .  5
      4.2.1 The paragraph Template  5
      4.2.2 The list Template . .  7
6 Language Support
7 hyperref Support
8 Notes
9 References
10 Index

1 Licence and Requirements

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

\texttt{enotez} needs and loads the following packages: l3kernel [The13a], xparse, xtemplate and l3keys2e from the l3packages bundle [The13b], xpatch [Gre12], scrlfile from the KOMA-Script bundle [KN12] and translations [Nie13b].

2 Motivation

\texttt{enotez} is a new implementation of endnotes for \LaTeX 2ε since the endnotes package [Lav03] has some deficiencies. Nested endnotes, for example, are not supported, neither is hyperref [OR12]. The sepfootnotes package [dLim13] also provides means for endnotes but actually
has a different purpose: to separate input and usage both of footnotes and endnotes. So it might
not be the best solution in every case.\footnote{It also does not allow nested endnotes.}

While \texttt{ENOTEZ} worked in tests nicely with the memoir\cite{WM11} class please keep in mind
that memoir provides its own endnote mechanism.

\texttt{ENOTEZ} enables nested endnotes properly and has another mechanism of customizing the
list of endnotes which is easily extendable. One of the main features of \texttt{ENOTEZ} is a split list of
endnotes in which the notes are automatically separated by the sections or chapters they were
set in, see section 5 for more information.

As an aside: \texttt{ENOTEZ} is nicely compatible with the fnpct package\cite{Nie13a}. Version 0.2j or
newer of fnpct automatically detects \texttt{ENOTEZ} and adapts the \texttt{\endnote} command.

3 Usage

3.1 Placing the Notes

The usage is simple: use \texttt{\endnote} in the text where you want to place the note mark.

\begin{verbatim}
\endnote{⟨mark⟩}{⟨text⟩}
\end{verbatim}

Add an endnote in the text.

\begin{verbatim}
\text{This is some text.}\endnote{With an endnote.}
\end{verbatim}

This is some text.\footnote{There’s not really much more to it. It is possible to add a custom mark by using the optional
argument but that shouldn’t be needed too often. \texttt{\endnote} works fine inside tables, minipages,
floats and captions.\footnote{Endnotes can also be nested.}

Since this functionality seemed making a pair \texttt{\endnotemark/\endnotetext} superfluous they are not
defined by \texttt{ENOTEZ}.}

\begin{verbatim}
\text{This is some text.}\endnote{With another endnote.}\endnote{This is a

\hspace{1cm}\text{nested endnote!}}\endnote{And another level deeper\ldots}
\end{verbatim}

% uses package ‘kantlipsum’ to produce dummy text:
\begin{verbatim}
\text{Of course you can have several paragraphs}\endnote{\kant[1-3]} in an endnote.
\end{verbatim}

This is some text.\footnote{Of course you can have several paragraphs\footnote{You have to write the actual notes in the preamble or a separate file and reference them in the text.} in an endnote.}
The marks of the endnotes in the running text are printed through the command \enotez\writemark which defaults to \textsuperscript. Its argument contains the current mark which is preceded by \enmarkstyle. Both of these commands can be redefined of course to adapt to custom settings. This can also be done using options, see section 4. The mark of the endnote that has been set last is stored in \theendnote and in \@currentlabel.

This could be used to define a command which writes a mark:

\begin{verbatim}
% we cannot use \newcommand -- the command starts with \end...
\makeatletter
\def\endnotemark{%
    \ifnextchar[\end@endnotemark\end@endnotemark[\theendnote]}
\def\end@endnotemark[#1]{\enotez\writemark{\enmarkstyle#1}}
\makeatother
\end{verbatim}

Text\endnotemark

Text

3

Please note that this definition does not step the endnote counter but either refers to the last number (no optional argument) or uses the one provided in the optional argument. Endnotes can also be labelled and later be referred to:

\begin{verbatim}
\begin{verbatim}
\end{verbatim}
\end{verbatim}

The next endnote\endnote{This endnote gets a label.}\label{en:test} has the number~\ref{en:test}. We can use this with the previous definition of \cs{endnotemark}\endnotemark[\ref{en:test}].

The next endnote\endnotemark has the number 4. We can use this with the previous definition of \endnotemark.

\subsection{Printing the Notes}

The notes are printed by using the command \printendnotes. \printendnotes*[⟨\texttt{style}⟩]

Print the list of endnotes. ⟨\texttt{style}⟩ is one of the instances explained in section 4.2.

If used without argument it prints all notes set so far with \endnote. The current list will then be cleared. All endnotes set after it are stored again for the next usage of \printendnotes. The starred version will print all endnotes but shouldn’t be used more than once if you have nested endnotes.

It may take several compilation runs until all notes are printed correctly. In a first run they are written to the aux file. In the second run they are available to \printendnotes. If you have nested endnotes they will be written to the aux file the first time they are printed with \printendnotes which means you might have to compile your file once more. If you change any of the endnotes or add another one you again will need at least two runs, maybe more. \enotez tries to warn you in these cases by invoking the warning
### 4 Options

Endnotes may have changed. Rerun to get them right.

but may not catch all cases.

\texttt{enotez} provides two commands that allow to set some kinds of preamble and postamble to a list, either to every list or only to the next one:

\texttt{\AtEveryEndnotesList\{⟨text⟩\}}

inserts \texttt{⟨text⟩} between heading and the actual notes every time \texttt{\printendnotes} is used.

\texttt{\AtNextEndnotesList\{⟨text⟩\}}

inserts \texttt{⟨text⟩} between heading and the actual notes the next time \texttt{\printendnotes} is used. This overwrites a possible preamble set with \texttt{\AtEveryEndnotesList} for this instance of \texttt{\printendnotes}.

\texttt{\AfterEveryEndnotesList\{⟨text⟩\}}

inserts \texttt{⟨text⟩} after the notes list every time \texttt{\printendnotes} is used.

\texttt{\AfterNextEndnotesList\{⟨text⟩\}}

inserts \texttt{⟨text⟩} after the notes list the next time \texttt{\printendnotes} is used. This overwrites a possible postamble set with \texttt{\AfterEveryEndnotesList} for this instance of \texttt{\printendnotes}.

If something is inserted with one of these commands the inserted \texttt{⟨text⟩} will be followed by a \texttt{\par} and a vertical skip for the preamble. The postambles follow a \texttt{\par} and a vertical skip. The skips can be set using an option, see section 4.

## 4 Options

### 4.1 Package Options

\texttt{enotez} has a few package options which should be pretty self-explanatory. They can be set either as package options with \texttt{\usepackage\[\langle options\rangle\]{enotez}} or with the setup command.

\texttt{\setenotez\{\langle options\rangle\}}

Setup command for setting \texttt{enotez}' options.

\texttt{list-name = \{⟨list name⟩\}}

The name of the notes list. This name is used for the heading of the list.

\texttt{reset = true|false}  

Default: false

If set to true the notes numbers will start from 1 again after \texttt{\printendnotes} has been invoked.

\texttt{counter-format = arabic|alph|Alph|roman|Roman|symbols}

Default: arabic

Change the format of the endnote counter. Please be aware that there are only 26 alphabetic counter symbols (options \texttt{alph} and \texttt{Alph} and only 9 symbols (option \texttt{symbols}).

\texttt{mark-format = \{⟨code⟩\}}

Redefine \texttt{\enmarkstyle} to execute \texttt{⟨code⟩}. This command is placed directly before the endnote mark in the text.
4 Options

mark-cs = \{\langle\text{command}\rangle\} 
Lets \enotez{writemark} to be equal to \langle\text{command}\rangle. This command is used to typeset the endnote marks in the text and should take one argument. Default: \textsuperscript

backref = true|false 
If set to true and \text{hyperref} has been loaded backlinks from the notes in the list to the marks in the text are added. Default: false

totoc = section|chapter|false 
Add an entry to the table of contents. Default: false

list-heading = \{\langle\text{sectioning command including argument}\rangle\} 
You can use this option to manually set the list heading command, e.g., \text{list-heading} = \{\\text{\chapter}\{#1\}\} for a numbered heading. The default depends upon if the class you’re using provides \text{\chapter*} or \text{\section*}. You can see that you have to refer to the actual heading with #1.

list-style = \{\langle\text{style}\rangle\} 
Sets the default list style, see section 4.2 for details. Default: plain

list-preamble-skip = \{\langle\text{skip}\rangle\} 
Sets the vertical skip (a rubber length) that is inserted if a list preamble is inserted by using either \AtNextEndnotesList or \AtEveryEndnotesList. It’s default is set equal to \text{\medskipamount}.

list-postamble-skip = \{\langle\text{skip}\rangle\} 
Sets the vertical skip (a rubber length) that is inserted if a list postamble is inserted by using either \AfterNextEndnotesList or \AfterEveryEndnotesList. It’s default is set equal to \text{\medskipamount}.

4.2 Customizing the List

The list is typeset with \text{xtemplate}’s possibilities. \text{ENOTEZ} declares the object \text{enotez-list} and two templates for it, the template \text{paragraph} and the template \text{list}.

4.2.1 The paragraph Template

The \text{paragraph} template’s interface is defined as follows:

\begin{verbatim}
\DeclareTemplateInterface{enotez-list}{paragraph}{1}
{
  % parameter : type = default
  heading : function 1 = \section*{#1} ,
  format : tokenlist = \footnotesize ,
  number : function 1 = \enmark{#1} ,
  number-format : tokenlist = \normalfont ,
  notes-sep : length = .5\baselineskip ,
}
\end{verbatim}
The parameters functions are these:

**heading**
- The command with which the heading is typeset.

**format**
- The format of the whole list.

**number**
- The command that is used to typeset the numbers of the notes. The command \enmark is explained soon.

**numbers-format**
- The format of the numbers.

**notes-sep**
- Additional space between the notes.

ENOTEZ uses this template to define the instance plain:

\begin{verbatim}
\DeclareInstance{enotez-list}{plain}{paragraph}{}
\end{verbatim}

This is the default style of the list. You can easily define your own instances, though:

\begin{verbatim}
\DeclareInstance{enotez-list}{custom}{paragraph}
{
 heading = \chapter*{#1} ,
 notes-sep = \baselineskip ,
 format = \normalfont ,
 number = \textsuperscript{#1}
}
\end{verbatim}

This would use a chapter heading for the title, separate the notes with \baselineskip and typeset them with \normalfont. The numbers would be typeset with \textsuperscript. You could now use it like this:

\begin{verbatim}
\printendnotes[custom]
\end{verbatim}

If you wanted superscripted numbers, you could also redefine \enmark.

\begin{verbatim}
\newcommand*{\enmark}{#1.}
\end{verbatim}
4.2.2 The list Template

The list template’s interface is defined as follows:

\begin{verbatim}
\DeclareTemplateInterface{enotez-list}{list}{1}
{
  % parameter : type = default
  heading : function 1 = \section*{#1} ,
  format : tokenlist = \footnotesize ,
  number : function 1 = \enmark{#1} ,
  number-format : tokenlist = \normalfont ,
  list-type : tokenlist = description ,
}
\end{verbatim}

This template uses a list to typeset the notes. As you can see the default list is a description list.

\texttt{enotez} defines two instances of this template:

\begin{verbatim}
\DeclareInstance{enotez-list}{description}{list}{},
\DeclareInstance{enotez-list}{itemize}{list}{list-type = itemize}
\end{verbatim}

They’re available through \texttt{\printendnotes[description]} and \texttt{\printendnotes[itemize]}, respectively.

Again you can define your own instances using whatever list you want, possibly one defined with the power of \texttt{enumitem}.

5 Collect Notes Section-wise and Print List Stepwise

This feature is experimental and surely has some limitations. Please let me know if something doesn’t work as expected.

Not to be misunderstood: you can use \texttt{\printendnotes} as often as you like, possibly after each section. That is not what is meant here. Let’s suppose you are writing a book and have many endnotes in many chapters. It would be nice if the list of endnotes at the end of the book could be split into parts for each chapter. This section describes how you can achieve that with \texttt{enotez}.

First of all \texttt{enotez} will rely on the fact that you use \texttt{\printendnotes} only once! If you call it more times nobody knows what will happen...

You’ll need to tell \texttt{enotez} that you want to split the notes into groups.

\texttt{split = section|chapter|false}  
Default: false

Enable the automatic splitting.

\texttt{split-sectioning = \{\texttt{csname}\}}  
(initially empty)

\texttt{This option is deprecated and may be dropped in future versions!} The command that is used to
display the titles between the splits. It needs to be a command that takes one argument and should be entered without the leading backslash. If the option is not used \enotez will choose \subsection* for split = {section} and \section* for split = {chapter}.

\texttt{split-heading = \{\langle sectioning command including argument \rangle\}}

The command that is used to display the titles between the splits. It is entered with argument and the actual title is referred to with \#1, e.g., \texttt{split-heading = \{\subsection*{\#1}\}}. If the option is not used \enotez will choose \subsection*{\#1} for split = {section} and \section*{\#1} for split = {chapter}.

\texttt{split-title = \{\langle tokenlist \rangle\}}

\texttt{Default: Notes for \langle name \rangle \langle ref \rangle}

The title that will be inserted between the splits. \langle name \rangle is replaced by section for split = \{section\} and chapter for split = \{chapter\}. \langle ref \rangle is replaced by the corresponding \thesection or \thechapter.

Set the \texttt{split} option:

\begin{verbatim}
\setenotez(split=section)
\end{verbatim}

Well – that’s it, basically. You’ll have to be careful, though: if you’re having nested endnotes the nested ones appear first in the “Notes” section (or chapter, respectively). In this case you should have a numbered section title for the notes, presumably in the appendix. You’ll need to create a new list style:

\begin{verbatim}
% preamble:
\usepackage{enotez}
\DeclareInstance{enotez-list}{section}{paragraph}{heading=\section{\#1}}
\setenotez{list-style=section,split=section}
% document:
\appendix
\printendnotes
\end{verbatim}

Please beware that the option \texttt{reset} also impacts here: the numbing will be reset for each section or chapter, depending on the choice you made for \texttt{split}.

There are two – or three, actually – additional commands:

\texttt{\AtEveryListSplit\{\langle code \rangle\}}

Inserts \langle code \rangle before each sub-heading in a splitted list.

\texttt{\AfterEveryListSplit\{\langle code \rangle\}}

Inserts \langle code \rangle after each sub-heading in a splitted list.

\texttt{\EnotezCurrentSplitTitle}

Holds the current sub-heading in a splitted list and may be used in \texttt{\AtEveryListSplit} and \texttt{\AfterEveryListSplit}.  

\newpage
6 Language Support

\textit{ENOTEZ} comes with an example document for a split list which you should find in the same folder as this documentation.

6 Language Support

\textit{ENOTEZ} uses the translations package [Nie13b] to translate language dependent strings. The advantage of this is that language settings with babel [Bra13] or polyglossia [Cha13] are detected automatically. However, the available translations are somewhat limited due to my limited language knowledge. If you find missing or wrong translations you can try to add or correct them by adding the corresponding versions of the following lines to your preamble:

\begin{verbatim}
1 \DeclareTranslation{English}{enotez-title}{Notes}
2 \% \%''<name> <ref>'' is a placeholder for e.g. `section 1'
3 \DeclareTranslation{English}{enotez-splitted-title}{Notes for <name> <ref>}
4 \DeclareTranslation{English}{enotez-section}{section}
5 \DeclareTranslation{English}{enotez-chapter}{chapter}
\end{verbatim}

If you like you can also drop me an email at contact@mychemistry.eu and I’ll add the correct translations to \textit{ENOTEZ}.

7 hyperref Support

If hyperref is loaded and you are using the option \texttt{toc = {⟨⟨⟩⟩}see p 5) the list title is linked via a \texttt{\phantomsection}.

If hyperref is used with \texttt{hyperfootnotes} set to \texttt{true} the endnote marks are linked to the respective entries in the list. If you also set \textit{ENOTEZ}’ option \texttt{backref} (see p 5) the notes in the list are themselves linked to the marks in the text.

Notes

This is an example of a preamble to the list set with \texttt{\AtNextEndnotesList}.

1. With an endnote.
2. With another endnote.³
3. As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline
of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

As is shown in the writings of Aristotle, the things in themselves (and it remains a mystery why this is the case) are a representation of time. Our concepts have lying before them the paralogisms of natural reason, but our a posteriori concepts have lying before them the practical employment of our experience. Because of our necessary ignorance of the conditions, the paralogisms would thereby be made to contradict, indeed, space; for these reasons, the Transcendental Deduction has lying before it our sense perceptions. (Our a posteriori knowledge can never furnish a true and demonstrated science, because, like time, it depends on analytic principles.) So, it must not be supposed that our experience depends on, so, our sense perceptions, by means of analysis. Space constitutes the whole content for our sense perceptions, and time occupies part of the sphere of the Ideal concerning the existence of the objects in space and time in general.

4. This endnote gets a label.

5. This is a nested* endnote!

6. And another level deeper...

This is an example of a postamble to the list set with \AfterEveryEndnotesList. Note that it would have started with a paragraph indent which was prevented here by using \noindent.

References


[The13b] The \LaTeX\ Project Team. \l3packages. version SVN 4582, July 28, 2013. url: http://mirror.ctan.org/macros/latex/contrib/l3packages/.  


Index

Symbols
\@endnotemark .................................. 3  
A \AfterEveryEndnotesList .......... 4 ff.  
\AfterEveryListItem .................. 8  
\AfterNextEndnotesList ........... 4 ff.  
\AtEveryEndnotesList ............ 4 ff.  
\AtEveryListItem .................. 8  
\AtNextEndnotesList ............ 4 ff., 9  
B babel (package) .................... 9  
backref .................................. 5 ff.  
Bezos, Javier ............................ 9  
Braams, Johannes .................... 9  
C \chapter .................................. 5  
Charette, François .................... 9  
counter-format .......................... 4  
D De Lima, Eduardo C. Lourenço .... 1  
E \endnote .................................. 2 ff.  
\endnotemark .................................. 2 ff.  
endnotes (package) .................... 1  
endnotetext ................................ 2  
enmark .................................. 5 ff.  
enmarkstyle .................................. 3 ff.  
\EnotezCurrentSplitTitle ....... 8  
\enotez writemark .................... 3 ff.  
enumitem (package) .................... 7  
F Fairbairns, Robin .................... 1  
fnpc (package) .......................... 2  
G Gregorio, Enrico .................... 1  
H hyperref (package) ................. 1, 5, 9  
K Kohm, Markus ....................... 1  
KOMA-Script (class) ................. 2  
KOMA-Script (bundle) .................. 1  
L \l3kernel (bundle) ................. 1  
\l3keys2e (package) .................. 1  
\l3packages (bundle) .................. 1  
Lavagnino, John .................... 1  
list-heading ......................... 5  
list-name .................................. 4  
list-postamble-skip .................. 5  
list-preamble-skip .................. 5  
list-style .................................. 5  
\LPL .................................. 1  
M Madsen, Lars ..................... 2  
mark-cs .................................. 5  
mark-format .......................... 4  
memor (class) .......................... 2  
N Neukahm, Frank .................... 1 ff.  
Niederberger, Clemens ............. 1 ff.  
O Oberdiek, Heiko .................... 1  
P polyglossia (package) ............... 9  
\printendnotes ....................... 3 ff., 6 ff.  
Q  
R Rahtz, Sebastian .................... 1  
reset .................................. 4, 8  
Reutenauer, Arthur .................. 9  
S scrifile (package) .................. 1  
sepfootnotes (package) .............. 1  
\setenotez .................. 4 ff., 8  
split .................................. 7 ff.  
split-heading .......................... 8  
split-sectioning .................... 7  
split-title .................................. 8  
T The \LaTeX\ Project Team .......... 1  
\thefootnotenumber .................. 3  
\totoc .................................. 5 ff.  
translations (package) ............. 1, 9  
W Wilson, Peter ..................... 2  
X xpatch (package) .................... 1  
\xtemplate (package) ............. 1 ff., 5  

11