Table of Contents

1 Licence and Requirements 4.2.2 The list Template ... 7
2 Motivation 1 5 Collect Notes Section-wise and Print List Stepwise 7
3 Usage 2 6 Language Support 9
  3.1 Placing the Notes 2 7 hyperref Support 10
  3.2 Printing the Notes 3
4 Options 4 10 Notes
  4.1 Package Options 4 11 References
  4.2 Customizing the List 5 12 Index
     4.2.1 The paragraph Template 5

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.”

\enotez{} needs and loads the following packages: l3kernel [L3Pa], xparse, xtemplate and l3keys2e from the l3packages bundle [L3Pb], xpatch [Gre12], scrfile from the KOMA-Script bundle [Koh15] and translations [Nie15].

2 Motivation

\enotez{} is a new implementation of endnotes for \LaTeX{} since the endnotes package [Lavo3] has some deficiencies. Nested endnotes, for example, are not supported, neither is hyper-
ref [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 [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 [Nie13]. Version 0.2j or newer of fnpct automatically detects \texttt{enotez} and adapts the \texttt{endnote} command.

\section{Usage}

\subsection{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)}
Add an endnote in the text.
\end{verbatim}

\begin{verbatim}
\endnotemark{(mark)}
Add an endnotemark.
\end{verbatim}

\begin{verbatim}
\endnotetext{(text)}
Add text to an endnote placed with \texttt{\endnotemark}.
\end{verbatim}

\begin{verbatim}
\endnote{With an endnote.}
This is some text.\endnote{With another endnote.}\endnote{This is a nested endnote.}\endnote{And another level deeper\ldots} endnote!}
\end{verbatim}

\begin{Turing}
\% uses package 'kantlipsum' to produce dummy text:
\end{Turing}

\begin{Turing}
\% of course you can have several paragraphs:\texttt{\endnote{\texttt{kant}[1-3]}} in an endnote.
\end{Turing}

\footnote{You have to write the actual notes in the preamble or a separate file and reference them in the text.}
3 Usage

This is some text. Of course you can have several paragraphs in an endnote.

The marks of the endnotes in the running text are printed through the command `\enotezwrite{mark}` which defaults to `\textsuperscript{}`. Its argument contains the current mark which is preceded by `\enmark`. 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 `\@currentlabel`.

Endnotes can also be labelled and later be referred to:

\begin{enumerate}
\item The next endnote `\endnote{This endnote gets a label.}\label{en:test}` has the number \ref{en:test}.
\item Let's now test `\cs{endnotemark}\endnotemark[\ref{en:test}]`.\end{enumerate}

The next endnote has the number 4. Let's now test `\endnotemark`.\end{center}

3.2 Printing the Notes

The notes are printed by using the command `\printendnotes`.

`\printendnotes*[⟨style⟩]`
Print the list of endnotes. ⟨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’re 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. `\enote` tries to warn you in these cases by invoking the warning

Endnotes may have changed. Rerun to get them right.

but may not catch all cases.

`\enote` 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:

`\AtEveryEndnotesList{⟨text⟩}`

inserts ⟨text⟩ between heading and the actual notes every time `\printendnotes` is used.
4 Options

\AtNextEndnotesList\{\text\}

interts \text\ between heading and the actual notes the next time \printendnotes\ is used. This overwrites a possible preamble set with \AtEveryEndnotesList\ for this instance of \printendnotes.

\AfterEveryEndnotesList\{\text\}

inserts \text\ after the notes list every time \printendnotes\ is used.

\AfterNextEndnotesList\{\text\}

inserts \text\ after the notes list the next time \printendnotes\ is used. This overwrites a possible postamble set with \AfterEveryEndnotesList\ for this instance of \printendnotes. If something is inserted with one of these commands the inserted \text\ will be followed by a \par\ and a vertical skip for the preamble. The postambles follow a \par\ and a vertical skip. The skips can be set using an option, see section 4.

4 Options

4.1 Package Options

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

\setenotez\{\texttt{options}\}

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

\texttt{list-name} = \{\texttt{list name}\}

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

\texttt{reset} = \texttt{true}|\texttt{false}

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

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

Default: \texttt{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} = \{\texttt{code}\}

Redefine \texttt{enotezwritemark} to execute \texttt{code}. This command is placed directly before the endnote mark in the text.

\texttt{mark-cs} = \{\texttt{command}\}

Lets \texttt{enotez writemark} to be equal to \texttt{command}. This command is used to typeset the endnote marks in the text and should take one argument.

\texttt{backref} = \texttt{true}|\texttt{false}

If set to \texttt{true} and \hyperref\ has been loaded backlinks from the notes in the list to the marks in the text are added.
4 Options

\texttt{totoc = section|chapter|false}  \hspace{1in} \text{Default: false}

Add an entry to the table of contents.

\texttt{list-heading = \{\textit{sectioning command including argument}\}}

You can use this option to manually set the list heading command, e.g., \texttt{list-heading = \{\chapter\{#1\}\}} for a numbered heading. The default depends upon if the class you’re using provides \chapter or not. It either uses \chapter* or \section*. You can see that you have to refer to the actual heading with \#1.

\texttt{list-style = \{\langle style\rangle\}}  \hspace{1in} \text{Default: plain}

Sets the default list style, see section 4.2 for details.

\texttt{list-preamble-skip = \{\langle skip\rangle\}}  \hspace{1in} \text{Default: \medskipamount}

Introduced in version 0.5

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 \medskipamount.

\texttt{list-postamble-skip = \{\langle skip\rangle\}}  \hspace{1in} \text{Default: \medskipamount}

Introduced in version 0.5

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 \medskipamount.

4.2 Customizing the List

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

4.2.1 The \texttt{paragraph} Template

The \texttt{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:

\texttt{heading}

The command with which the heading is typeset.
4 Options

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:

\DeclareInstance{enotez-list}{plain}{paragraph}{}

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

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

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:

\printendnotes[custom]

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

\enmark
this command is initially defined like this: \newcommand*\enmark{1}{#1}
4.2.2 The list Template

The list template’s interface is defined as follows:

```
\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 ,
}
```

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

ENOTEZ defines two instances of this template:

```
\DeclareInstance{enotez-list}{description}{list}{}
\DeclareInstance{enotez-list}{itemize}{list}{{list-type = itemize}}
```

They’re available through `\printendnotes[description]` and `\printendnotes[itemize]`, respectively.

Again you can define your own instances using whatever list you want, possibly one defined with the power of 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 `\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 ENOTEZ.

First of all ENOTEZ will rely on the fact that you use `\printendnotes` only once! If you call it more times nobody knows what will happen…

You’ll need to tell ENOTEZ that you want to split the notes into groups.
5 Collect Notes Section-wise and Print List Stepwise

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

Enable the automatic splitting.

\texttt{split-sectioning = \{\langle csname\rangle\}}  
\texttt{\{ 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 \texttt{enotez} will choose subsection* for \texttt{split = \{section\}} and section* for \texttt{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, \textit{e.g.}, \texttt{split-heading = \{\texttt{\subsection*{\#1}}\}}. If the option is not used \texttt{enotez} will choose \texttt{\subsection*{\#1}} for \texttt{split = \{section\}} and \texttt{\section*{\#1}} for \texttt{split = \{chapter\}}.

\texttt{split-title = \{\langle tokenlist\rangle\}}  
Default: Notes for \texttt{\langle name\rangle \langle ref\rangle}

The title that will be inserted between the splits. \texttt{\langle name\rangle} is replaced by section for \texttt{split = \{section\}} and chapter for \texttt{split = \{chapter\}}. \texttt{\langle ref\rangle} is replaced by the corresponding \texttt{\thesec tion} or \texttt{\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 numbering will be reset for each section or chapter, depending on the choice you made for \texttt{split}.

There are additional commands:
6 Language Support

\AtEveryListSplit{(code)}
Inserts (code) before each sub-heading in a splitted list.

\AfterEveryListSplit{(code)}
Inserts (code) after each sub-heading in a splitted list.

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

\NewSplitTitleTag{⟨tag⟩}{⟨replacement⟩}
In the sub-headings of a splitted list, the tags <ref>, <name> and <split-level-id> can be used per default. This command allows to define additional tags.

The tags that can be used for setting the sub-headings of a splitted list have the following meaning:

<ref> The counter representation of the current section or chapter, i.e., either the corresponding expansion of \thesection or \thechapter, depending on the split-level.

<name> The name of the split-level, i.e., either “section” or “chapter”, depending on the split-level. This is language sensitive (see section 6). The corresponding translation ids are enotez-section and enotez-chapter, respectively.

<split-level-id> The number of the current section or chapter, i.e., either the corresponding expansion of \arabic{section} or \arabic{chapter}, depending on the split-level.

The command \NewSplitTitleTag let’s you define your own. You can use the three existing tags inside the replacement. Note that the ⟨tag⟩ argument is input without opening < and closing >! \NewSplitTitleTag{own}{...} would define the tag <own>.

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

6 Language Support

ENOTEZ uses the translations package [Nie15] to translate language dependent strings. The advantage of this is that language settings with babel [Bra16] or polyglossia [Cha15] 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:

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

7 hyperref Support

If hyperref is loaded and you are using the option \totoc (see page 5) the list title is linked via a \phantomsection.

If hyperref is used with \hyperfootnotes set to true the endnote marks are linked to the respective entries in the list. If you also set ENOTEZ’ option \backref (see page 4) 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 \AtNextEndnotesList.

1. With an endnote.

2. With another endnote.5

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!
References

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

References

        \url{http://mirror.ctan.org/macros/latex/required/babel/}.

        polyglossia. version 1.42.0, Aug. 6, 2015 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/polyglossia/}.

        \url{http://mirror.ctan.org/macros/latex/contrib/sepfootnotes/}.

        \url{http://mirror.ctan.org/macros/latex/contrib/xpatch/}.

        \url{http://mirror.ctan.org/macros/latex/contrib/koma-script/}.

[L3Pa] The \LaTeX3\ Project Team. \l3kernel. version SVN 6377, Jan. 19, 2016 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/l3kernel/}.

[L3Pb] The \LaTeX3\ Project Team.
        \l3packages. version SVN 6377, Jan. 19, 2016 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/l3packages/}.

        endnotes. version NA, Jan. 15, 2003 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/endnotes/}.

[Nie13] Clemens Niederberger. fnpct. version 0.2j, Feb. 28, 2013 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/fnpct/}.

[Nie15] Clemens Niederberger. translations. version 1.2e, Nov. 7, 2015 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/translations/}.

        hyperref. version 6.83m, Nov. 6, 2012 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/hyperref/}.

[WM11] Peter Wilson and Lars Madsen. memoir. version 3.6j, Mar. 6, 2011 (or newer).
        \url{http://mirror.ctan.org/macros/latex/contrib/memoir/}. 

11