

ENOTEZ

vo.9a 2017/04/24

Endnotes for L^AT_EX 2_ε

Clemens NIEDERBERGER

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

contact@mychemistry.eu

Table of Contents

1	Licence and Requirements	1	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	Notes	10
4	Options	4	References	11
4.1	Package Options	4	Index	12
4.2	Customizing the List	5		
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 L^AT_EX Project Public License (LPPL), version 1.3 or later (<http://www.latex-project.org/lppl.txt>). The software has the status “maintained.”

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 L^AT_EX 2_ε since the endnotes package [Lav03] 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.¹ It also does not allow nested endnotes.

While `ENOTEZ` worked in tests nicely with the memoir [WM11] class please keep in mind that memoir provides its own endnote mechanism.

`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 `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: `ENOTEZ` is nicely compatible with the fnpct package [Nie13]. Version 0.2j or newer of fnpct automatically detects `ENOTEZ` and adapts the `\endnote` command.

3 Usage

3.1 Placing the Notes

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

`\endnote[⟨mark⟩]{⟨text⟩}`
Add an endnote in the text.

Introduced in
version 0.9

`\endnotemark[⟨mark⟩]`
Add an endnotemark.

Introduced in
version 0.9

`\endnotetext{⟨text⟩}`
Add text to an endnote placed with `\endnotemark`.

```
1 This is some text.\endnote{With an endnote.}
```

This is some text.¹

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. Endnotes can also be nested.

```
1 This is some text.\endnote{With another endnote.\endnote{This is a
2   nested\endnote{And another level deeper\ldots} endnote!}}
3 % uses package `kantlipsum' to produce dummy text:
4 Of course you can have several paragraphs\endnote{\kant[1-3]} in an endnote.
```

1. You have to write the actual notes in the preamble or a separate file and reference them in the text.

This is some text.² Of course you can have several paragraphs³ in an endnote.

The marks of the endnotes in the running text are printed through the command `\enotezwritemark` 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 `\@currentlabel`.

Endnotes can also be labelled and later be referred to:

```

1 The next endnote\endnote{This endnote gets a label.}\label{en:test} has
2 the number~\ref{en:test}. Let's now test
3 \cs{endnotemark}\endnotemark[\ref{en:test}].

```

The next endnote⁴ has the number 4. Let's now test `\endnotemark4`.

3.2 Printing the Notes

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

`\printendnotes*[\langle style \rangle]`

Print the list of endnotes. `\langle style \rangle` 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. **ENOTEZ** 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.

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:

`\AtEveryEndnotesList{\langle text \rangle}`

inserts `\langle text \rangle` between heading and the actual notes every time `\printendnotes` is used.

`\AtNextEndnotesList{⟨text⟩}`
 Introduced in version 0.5 inserts `⟨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⟩}`
 Introduced in version 0.5 inserts `⟨text⟩` after the notes list every time `\printendnotes` is used.

`\AfterNextEndnotesList{⟨text⟩}`
 Introduced in version 0.5 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

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

`\setenotez{⟨options⟩}`
 Setup command for setting `ENOTEZ`' options.

`list-name = {⟨list name⟩}` Default: Notes
 The name of the notes list. This name is used for the heading of the list.

`reset = true|false` Default: false
 If set to `true` the notes numbers will start from 1 again after `\printendnotes` has been invoked.

`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 `alph` and `Alph` and *only 9* symbols (option `symbols`)).

`mark-format = {⟨code⟩}` (initially empty)
 Redefine `\enmarkstyle` to execute `⟨code⟩`. This command is placed directly before the endnote mark in the text.

`mark-cs = {⟨command⟩}` Default: `\textsuperscript`
 Lets `\enotezwritemark` to be equal to `⟨command⟩`. This command is used to typeset the endnote marks in the text and should take one argument.

`backref = true|false` Default: false
 Introduced in version 0.7 If set to `true` and `hyperref` has been loaded backlinks from the notes in the list to the marks in the text are added.

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

`list-heading = {\sectioning command including argument}`
 You can use this option to manually set the list heading command, e.g., `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`.

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

Introduced in
version 0.5

`list-preamble-skip = {\skip}` Default: `\medskipamount`
 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`.

Introduced in
version 0.5

`list-postamble-skip = {\skip}` Default: `\medskipamount`
 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. `ENOTEZ` declares the object `enotez-list` and two templates for it, the template paragraph and the template list.

4.2.1 The paragraph Template

The paragraph template's interface is defined as follows:

```

1 \DeclareTemplateInterface{enotez-list}{paragraph}{1}
2 {
3   % parameter : type      = default
4   heading    : function 1 = \section*{#1} ,
5   format     : tokenlist = \footnotesize ,
6   number     : function 1 = \enmark{#1} ,
7   number-format : tokenlist = \normalfont ,
8   notes-sep  : length    = .5\baselineskip ,
9 }

```

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:

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

This is the default style of the list.

You can easily define your own instances, though:

```
1 \DeclareInstance{enotez-list}{custom}{paragraph}
2 {
3   heading    = \chapter*{#1}      ,
4   notes-sep = \baselineskip      ,
5   format     = \normalfont       ,
6   number     = \textsuperscript{#1}
7 }
```

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:

```
1 \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:

```

1 \DeclareTemplateInterface{enotez-list}{list}{1}
2 {
3   % parameter : type      = default
4   heading    : function 1 = \section*{#1} ,
5   format     : tokenlist = \footnotesize ,
6   number     : function 1 = \enmark{#1} ,
7   number-format : tokenlist = \normalfont ,
8   list-type  : tokenlist = description ,
9 }

```

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:

```

1 \DeclareInstance{enotez-list}{description}{list}{}
2 \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.

`split = section|chapter>false` Default: false
 Enable the automatic splitting.

`split-sectioning = {<csname>}` (initially empty)

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}`.

`split-heading = {<sectioning command including argument>}`

Introduced in
version 0.3

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., `split-heading = {\subsection*{#1}}`. If the option is not used `ENOTEZ` will choose `\subsection*{#1}` for `split = {section}` and `\section*{#1}` for `split = {chapter}`.

`split-title = {<tokenlist>}` Default: Notes for <name> <ref>

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

Set the `split` option:

```
1 \setenotez{split=section}
```

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:

```
1 % preamble:
2 \usepackage{enotez}
3 \DeclareInstance{enotez-list}{section}{paragraph}{heading=\section{#1}}
4 \setenotez{list-style=section,split=section}
5 % document:
6 \appendix
7 \printendnotes
```

Please beware that the option `reset` also impacts here: the numbering will be reset for each section or chapter, depending on the choice you made for `split`.

There are additional commands:

Introduced in version 0.7	<code>\AtEveryListSplit{<code>}</code> Inserts <code><code></code> before each sub-heading in a splitted list.
Introduced in version 0.7	<code>\AfterEveryListSplit{<code>}</code> Inserts <code><code></code> after each sub-heading in a splitted list.
Introduced in version 0.7	<code>\EnotezCurrentSplitTitle</code> Holds the current sub-heading in a splitted list and may be used in <code>\AtEveryListSplit</code> and <code>\AfterEveryListSplit</code> .
Introduced in version 0.8	<code>\NewSplitTitleTag{<tag>}{<replacement>}</code> In the sub-headings of a splitted list, the tags <code><ref></code> , <code><name></code> and <code><split-level-id></code> 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.

Changed in
version 0.7

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 nested⁶ endnote!

References

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

- [Bra16] Johannes BRAAMS, current maintainer: Javier BEZOS.
babel. version 3.9q, Feb. 24, 2016 (or newer).
URL: <http://mirror.ctan.org/macros/latex/required/babel/>.
- [Cha15] François CHARETTE, current maintainer: Arthur REUTENAUER.
polyglossia. version 1.42.0, Aug. 6, 2015 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/polyglossia/>.
- [dLim13] Eduardo C. Lourenço DE LIMA. sepfootnotes. version 0.2, Jan. 17, 2013 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/sepfootnotes/>.
- [Gre12] Enrico GREGORIO. xpatch. version 0.3, Oct. 2, 2012 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/xpatch/>.
- [Koh15] Markus KOHM. KOMA-Script. version 3.18, July 2, 2015 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/koma-script/>.
- [L3Pa] THE L^AT_EX₃ PROJECT TEAM. l3kernel. version SVN 6377, Jan. 19, 2016 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/l3kernel/>.
- [L3Pb] THE L^AT_EX₃ PROJECT TEAM.
l3packages. version SVN 6377, Jan. 19, 2016 (or newer).
URL: <http://mirror.ctan.org/macros/latex/contrib/l3packages/>.
- [Lav03] John LAVAGNINO, current maintainer: Robin FAIRBAIRNS.
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/>.
- [OR12] Heiko OBERDIEK and Sebastian RAHTZ.
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/>.

Index

A		
<code>\AfterEveryEndnotesList</code>	4 f., 11	
<code>\AfterEveryListSplit</code>	9	
<code>\AfterNextEndnotesList</code>	4 f.	
<code>\AtEveryEndnotesList</code>	3 ff.	
<code>\AtEveryListSplit</code>	9	
<code>\AtNextEndnotesList</code>	4 f., 10	
B		
babel (package)	9	
backref	4, 10	
BEZOS, Javier	9	
BRAAMS, Johannes	9	
C		
<code>\chapter</code>	5	
CHARETTE, François	9	
counter-format	4	
E		
<code>\endnote</code>	2 f.	
<code>\endnotemark</code>	2 f.	
endnotes (package)	1	
<code>\endnotetext</code>	2	
<code>\enmark</code>	5 ff.	
<code>\enmarkstyle</code>	3 f.	
<code>\EnotezCurrentSplitTitle</code>	9	
<code>\enotezwritemark</code>	3 f.	
enumitem (package)	7	
F		
FAIRBAIRNS, Robin	1	
fnpct (package)	2	
G		
GREGORIO, Enrico	1	
H		
hyperref (package)	2, 4, 10	
K		
KOHN, Markus	1	
KOMA-Script (bundle)	1	
L		
l3kernel (bundle)	1	
l3keys2e (package)	1	
l3packages (bundle)	1	
LAVAGNINO, John	1	
DE LIMA, Eduardo C. Lourenço	2	
list-heading	5	
list-name	4	
list-postamble-skip	5	
list-preamble-skip	5	
list-style	5	
LPPL	1	
M		
MADSEN, Lars	2	
mark-cs	4	
mark-format	4	
memoir (class)	2	
N		
<code>\NewSplitTitleTag</code>	9	
NIEDERBERGER, Clemens	1 f., 9	
O		
OBERDIEK, Heiko	2	
P		
polyglossia (package)	9	
<code>\printendnotes</code>	3 f., 6 ff.	
R		
RAHTZ, Sebastian	2	
reset	4, 8	
REUTENAUER, Arthur	9	
S		
scrfile (package)	1	
sepfootnotes (package)	2	
<code>\setenotez</code>	4, 8	
split	8	
split-heading	8	
split-sectioning	8	
split-title	8	
T		
THE L ^A T _E X ₃ PROJECT TEAM	1	
totoc	5, 10	
translations (package)	1, 9	
W		
WILSON, Peter	2	
X		
xparse (package)	1	
xpatch (package)	1	
xtemplate (package)	1, 5	