

# ENOTEZ

vo.3a 2012/07/24

Endnotes for  $\text{\LaTeX}$  2<sub>ε</sub>

Clemens NIEDERBERGER

<https://bitbucket.org/cgnieder/enotez/>  
contact@mychemistry.eu

English documentation

## Contents

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

## 1 Licence and Requirements

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

**ENOTEZ** needs and loads the following packages: `expl3`,<sup>1</sup> `xparse`,<sup>2</sup> `xtemplate`<sup>3</sup> and `l3keys2e`.<sup>4</sup> If you’re using `memoir`,<sup>5</sup> a KOMA-Script<sup>6</sup> class or the package `caption`<sup>7</sup> **ENOTEZ** also needs and loads `etoolbox`.<sup>8</sup>

---

<sup>1</sup> CTAN: `expl3`   <sup>2</sup> CTAN: `xparse`   <sup>3</sup> CTAN: `xtemplate`   <sup>4</sup> CTAN: `l3keys2e`   <sup>5</sup> CTAN: `memoir`   <sup>6</sup> CTAN: KOMA-Script   <sup>7</sup> CTAN: `caption`   <sup>8</sup> CTAN: `etoolbox`

## 2 Motivation

**ENOTEZ** is a new implementation of endnotes for  $\text{\LaTeX}$  2 $\epsilon$  since the endnotes<sup>9</sup> package [Faio3] has some deficiencies. Nested endnotes, for example, are not supported, neither is hyperref.<sup>10</sup> The sepfootnotes<sup>11</sup> package [Lim12] 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.<sup>12</sup> It also does not allow nested endnotes.

**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<sup>13</sup> package.<sup>14</sup>

## 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.

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

This is some text.<sup>1</sup>

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. `\endnote` works fine inside tables, minipages, floats and captions.<sup>15</sup> Endnotes can also be nested. Since that seemed making a pair `\endnotemark/\endnotetext` superfluous they are *not* defined by **ENOTEZ**.

```
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':
4 Of course you can have several paragraphs\endnote{\kant[1-3]} in an
   endnote.
```

This is some text.<sup>2</sup> Of course you can have several paragraphs<sup>3</sup> in an endnote.

---

<sup>9</sup> CTAN: endnotes   <sup>10</sup> CTAN: hyperref   <sup>11</sup> CTAN: sepfootnotes   <sup>12</sup> You have to write the actual notes in the preamble or a separate file and reference them in the text.   <sup>13</sup> CTAN: fnpct   <sup>14</sup> You'll have to use `\AdaptNote` for the time being, though.   <sup>15</sup> This has been tested with the standard classes, memoir, and the KOMA-Script classes, with and without the caption package. If you're using another package that redefines `\caption` it might not work.

## 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. `ENOTEZ` tries to warn you in these cases by invoking L<sup>A</sup>T<sub>E</sub>X's warning

Label(s) may have changed. Rerun to get cross-references right.

but may not catch all cases.

## 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` Default: arabic  
Change the format of the endnote counter.
- `totoc = section|chapter|false` Default: false  
Add an entry to the table of contents.
- `list-style = <style>` Default: plain  
Sets the default list style, see section 4.2 for details.

## 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` → is 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`.<sup>16</sup>

## 5 Collect Notes Section-wise and Print List Stepwise

*This feature is experimental and has some limitations.*

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)  
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-title` = <tokenlist> Default: Notes from <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
4   {#1}}
5 \setenotez{list-style=section,split=section}
6 % document:
```

---

<sup>16</sup> CTAN: `enumitem`

```
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`.

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

## 6 hyperref Support

If `hyperref` is loaded and you are using the option `totoc` (see p 3) 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.

## Notes

1. With an endnote.
2. With another endnote.<sup>4</sup>
3. This is the Kant lipsum generator. 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.
4. This is a nested<sup>5</sup> endnote!
5. And another level deeper...

## References

- [Faio3] Robin Fairbairns (current maintainer). *endnotes*. Version NA. Jan. 15, 2003. URL: <http://www.ctan.org/pkg/endnotes> (visited on 07/03/2012).
- [Lim12] Eduardo C. Lourenço de Lima. *sepfootnotes*. Version 0.1. Mar. 6, 2012. URL: <http://www.ctan.org/pkg/sepfootnotes> (visited on 07/03/2012).

## Index

Section titles are indicated **bold**, packages sans serif, commands `\brown` and options `yellow`.

### C

`caption` ..... 1 f.  
`counter-format` ..... 3

### E

`\endnote` ..... 2 f.  
`\endnotemark` ..... 2  
`endnotes` ..... 2  
`\endnotetext` ..... 2  
`\enmark` ..... 4 f.  
`enumitem` ..... 6  
`etoolbox` ..... 1  
`expl3` ..... 1

### F

`fnpct` ..... 2

### H

`hyperref` ..... 2, 7

### K

`KOMA-Script` ..... 1 f.

### L

`l3keys2e` ..... 1  
`list-name` ..... 3  
`list-style` ..... 3

### M

`memoir` ..... 1 f.

### P

`\printendnotes` ..... 3, 5 f.

### R

`reset` ..... 3, 7

### S

`sepfootnotes` ..... 2  
`\setenotez` ..... 3

`split` ..... 6 f.  
`split-sectioning` ..... 6  
`split-title` ..... 6

### T

`totoc` ..... 3, 7

### X

`xparse` ..... 1  
`xtemplate` ..... 1, 4