The **endnotes-hy** Package

D. P. Story

Email: `dpstory@uakron.edu`

processed April 10, 2020

Contents

1 ⟨∗package⟩

1 Introduction

This package is an extension of endnotes.

Background: A colleague—a dedicated user of the endnotes package—complained to me (for some reason) that links created by hyperref do not point to the endnotes at the end of the file; for example, if we say

`\endnote{\label{⟨name⟩}⟨text⟩}`

then any link, such as `\hyperref[⟨name⟩]{endnote~\ref*{⟨name⟩}}` jumps to the most recent anchor (`Doc-Start`, for example); no hypertext anchor is created where the endnote is inserted into the document. This short package is an attempt to rectify this problem.

Syntax: The `\endnote` command has been modified to conform to the following syntax:

`\endnote*[⟨num⟩]{⟨text⟩}\label{⟨name⟩}`

Refer to the documentation of the endnotes package for a description of the `\endnote` command and its ⟨*num*⟩ and ⟨*text*⟩ arguments. If the ∗-option (new

syntax) is used, the endnote mark is not placed, but the endnote is written to the ENT file. Such a "secret" endnote can be referred to using standard hyperref commands, see the endnote 1 on page 4 for more information. The placement of the \label has changed for this package as well. To define an endnote label, place \label{name} following the \endnote command, a more traditional location. This was done to get access to the ⟨*name*⟩, if it exist. This enables the package to place a hypertext link around the endnote mark.[2] The optional ⟨*num*⟩ can be used to write special endnotes.[N1] As a side benefit to this new syntax, you can place verbatim text into an endnote.[3] If there is no \label following \endnote, no hypertext link is created.[4]

Note: This package will work correctly without the hyperref package (of course, no links). The advantage of using endnotes-hy without hyperref is to use the extended syntax: \endnote*[⟨*num*⟩]{⟨*text*⟩}\label{⟨*name*⟩}

2   Required packages

```
2 \RequirePackage{endnotes}
```

Require etoolbox because we redefine several commands using \patchcmd; this is not really necessary, we could have redefined the whole command instead.

```
3 \RequirePackage{etoolbox}
```

3   Package code

We allow the author not to use hyperref; this enables the use of the extended syntax of the \endnote command.

```
4 \newcommand\hyperref@nohy[2][]{#2}
5 \AtBeginDocument{\@ifundefined{hyperref}{\let\hyperref@en\hyperref@nohy
6   \let\phantomendnote\relax}
7   {\let\hyperref@en\hyperref}}
```

\phantomendnote The key is to create an hyperref anchor to reference; this definition is based on \phantomsection of hyperref. This is used internally.

```
8 \def\phantomendnote{%
9   \Hy@MakeCurrentHrefAuto{endnote}%
10   \Hy@raisedlink
11     {\hyper@anchorstart{endnote.\@theenmark}\hyper@anchorend}}
12 \def\endnoteautorefname{endnote}
```

\theendnotes Redefinition of \theendnotes

```
13 \patchcmd{\theendnotes}{\enoteformat}
14   {\phantomendnote\def\@currentHref{endnote.\@theenmark}%
15     \enoteformat}{}{}
```

\endnote Redefine \endnote to take an optional *, if this option is used, the mark does not appear in the text. This enables us to easily define an endnote and refer to it even with multiple paragraphs. I didn't like the syntax of inserting the \label within

the argument of `\endnote`, this seems to have problems. The new syntax is as follows:

Syntax: `\endnote*[⟨num⟩]{⟨text⟩}\label{⟨name⟩}`

This new syntax could be a problem if an author has already used endnotes and then converts to endnotes-hy. In endnotes the *-option is note defined and the `\label` command is typically placed within the argument of the `{⟨text⟩}`. Labels would have to be moved to a position following `{⟨text⟩}`.

To obtain this syntax, we need several stages of parsing. The major problem is to get the label name in time to build a hyperref link around `\@endnotemark`.

```
16 \newtoks\@entoks
17 \def\endnote{\let\@encurrlabelname\@empty
18    \@ifstar{\let\@noMrk1\endnote@i}{\let\@noMrk0\endnote@i}}
19 \def\endnote@i{\@ifnextchar[%
20    {\@xendnote}
21    {\stepcounter{endnote}%
22    \protected@xdef\@theenmark{\theendnote}%
```

We placed the `{text}` argument into a token register, this removes that argument from the input stream, and allows us to see of the next token is `\label`, if so, we get the label name argument.

```
23    \afterassignment\endnote@ii\@entoks}}
```

If the next token is `\label`, get the label name with `\endnote@iii`, otherwise, move on to `\endnote@iv`, the final step.

```
24 \def\endnote@ii{\@ifnextchar\label{\endnote@iii}{\endnote@iv}}
25 \def\endnote@iii\label#1{\def\@encurrlabelname{#1}\endnote@iv}
```

If the flag `\@noMrk` is 0 (mark is typeset), and if there is a label name, we make an hypertext link.

```
26 \def\endnote@iv{%
27    \if\@noMrk0\relax
28      \ifx\@encurrlabelname\@empty
29        \@endnotemark
30      \else
31        \hyperref@en[\@encurrlabelname]{\@endnotemark}%
32      \fi
33    \fi
```

Finish by expanding `\@endnotetext` with its argument that was saved in the `\@entoks` register.

```
34    \expandafter\@endnotetext\expandafter{\the\@entoks}}
```

`\@endnotetext` We patch into `\@endnotetext` just after `\def\next{#1}`. If there was a `\label`, we insert it back into the argument of `\endnote`.

```
35 \patchcmd{\@endnotetext}
36    {\def\next{#1}}
37    {\ifx\@encurrlabelname\@empty
38      \def\next{#1}\else
39      \edef\x{\noexpand\label{\@encurrlabelname}}%
40      \expandafter\def\expandafter\next\expandafter{\x#1}\fi}{}{}
```

\xendnote   If \endnote has an optional argument, the flow passes to \xendnote. We replace \@endnotemark\@endnotetext by redirecting flow to to \endnote@ii, after saving the argument in \@entoks.

```
41 \patchcmd{\@xendnote}
42   {\@endnotemark\@endnotetext}
43   {\afterassignment\endnote@ii\@entoks}
44   {}{}
```

45 ⟨/package⟩

Notes

[1]An endnote created with *-option
[2]This is normal endnote
[N1]This is special endnote
[3]Some verbatim text `%^$^&$%^&`
[4]No hypertext link created

4 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.

5 Change History

v0.1 (2020/04/08)
General: First published version of this package. . 1