The etaremune package *

Hendri Adriaens
http://stuwww.uvt.nl/~hendri

v1.2 (2005/06/01)

Abstract

This package implements the `etaremune` environment which is similar to the `enumerate` environment, except that labels are decreasing instead of increasing. This package provides an alternative to the revnum package which uses a lot of counters. The style of lists can be controlled locally and globally.

Contents

1 Introduction

This package implements the `etaremune` environment which labels items with decreasing numbers, starting from the number of items in the environment. Since this number is not known at the start of the environment, we use a two step procedure. On the first LaTeX run, we determine the number of items in each `etaremune` environment and number the items in a default way and on the second run, we number all items correctly.

The revnum package [2] already implements this environment. It provides the `revnumerate` environment which labels the items with decreasing numbers. This package has as main drawback that it consumes numerous counters. It uses 4 counters plus 1 for every `revnumerate` environment in the document. In big documents, this can cause problems. The package at hand uses only 2 counters, irrespective of the number of `etaremune` environments used in the document.

I created the first version of the main code of this package to solve a question on the TeX-NL mailing list. I figured that it might be a good idea to put my code in a small package on CTAN as an alternative for revnum.

*This package can be downloaded from CTAN mirrors: `/macros/latex/contrib/etaremune`. See `etaremune.dtx` for information on installing etaremune into your LaTeX distribution and for the license of this package.

2 The `etaremune` environment

environment
`etaremune`

The `etaremune` environment works just as the `enumerate` environment and has the same control possibilities using macros like `\labelenumi`, `\theenumi` and `\@listi`. See for a detailed description of customizing `enumerate` and `etaremune` environments a LaTeX manual, for instance [1], pages 128–131 and 144–151.

The `etaremune` environment is different from the `enumerate` environment in labeling the items. This environment labels items with decreasing numbers.

```
\begin{etaremune}[⟨options⟩]
⟨items⟩
\end{etaremune}
```

As LaTeX cannot know at the beginning of the environment how many items it will have to typeset, this environment computes the starting point of the list in two LaTeX runs, using the auxiliary file. On the first run, the number of items is counted and the items will be numbered with increasing numbers (as in the `enumerate` environment) and on the second run, the item labels will be set correctly. See an example below.

3. Not important.
2. Quite relevant.
1. Paramount.

```
\begin{etaremune}
 \item Not important.
 \item Quite relevant.
 \item Paramount.
\end{etaremune}
```

The `etaremune` environment will avoid creating labels with negative numbers as this can cause problems when using alphabets for labels.[1] So when adding items to an existing list, these will be numbered with 0 on the next run. Another LaTeX run will settle the item labels again.

The optional argument ⟨*options*⟩ can be used to control the style of the environment (see below) and the starting point (if one wants to replace the starting point computed by etaremune).

5. First.
4. Second.
3. Third.

```
\begin{etaremune}[start=5]
 \item First.
 \item Second.
 \item Third.
\end{etaremune}
```

The `etaremune` environment can be nested up to four levels deep, like the standard `enumerate` environment. The example below also shows an example of customizing item labels and referring to items. Notice that it takes three LaTeX runs for references to

[1] Positions -1, -2,... in an alphabet are not defined by the LaTeX macro `\@alph` and friends.

items in the etaremune environment to settle.

1) First.

   (iii) third.

   (ii) second.

   (i) first.

2) Second.

3) Third.

Notice item 1iii.

```
\renewcommand{\labelenumi}{\theenumi)}
\renewcommand{\theenumii}{\roman{enumii}}
\begin{enumerate}
 \item First.
  \begin{etaremune}
   \item third.\label{notice}
   \item second.
   \item first.
  \end{etaremune}
 \item Second.
 \item Third.
\end{enumerate}
Notice item~\ref{notice}.
```

options
topsep
partopsep
itemsep
parsep
leftmargin
rightmargin
listparindent
itemindent
labelwidth
labelsep

The etaremune package allows for specifying some style parameters to control the markup of lists. These are the vertical lengths \topsep, \partopsep, \itemsep and \parsep and the horizontal lengths \leftmargin, \rightmargin, \listparindent, \itemindent, \labelwidth and \labelsep.[2] When set through the optional argument ⟨*options*⟩, these lengths are set locally (so holding only for the environment at hand).

3. third.
2. second.

   (b) second.

   (a) first.

1. first.

```
\begin{etaremune}[itemsep=0pt,parsep=0pt]
 \item third.
 \item second.
  \begin{etaremune}
   \item second.
   \item first.
  \end{etaremune}
 \item first.
\end{etaremune}
```

One can also change the style of all etaremune environments throughout the document by specifying these options in the \usepackage command.

3. third.

2. second.

(b) second.

(a) first.

1. first.

```
\usepackage[leftmargin=0pt,labelsep=20pt]{etaremune}
...
\begin{etaremune}
\item third.
\item second.
 \begin{etaremune}
  \item second.
  \item first.
 \end{etaremune}
\item first.
\end{etaremune}
```

3 Implementation

```
1 %<*etaremune>
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3 \ProvidesPackage{etaremune}[2005/06/01 v1.2 Reversed enumerate (HA)]
```

[2] See for instance [1], pages 144–151, for more information.

Build global and local options. We need some delicate work with presets here as LaTeX resets dimens locally in each environment. So setting lengths globally couldn't be done without presets.

```
\RequirePackage{xkeyval}
\def\@tempa#1#2{%
  \DeclareOptionX{#1}{\presetkeys[EM]{template}{#1=##1}{}}%
  \define@key[EM]{template}{#1}{\setlength#2{##1}}%
}
\XKV@for@n{topsep,partopsep,itemsep,parsep,leftmargin,rightmargin,%
  listparindent,itemindent,labelwidth,labelsep}\@tempb{%
  \edef\@tempb{%
    {\@tempb}\expandafter\noexpand\csname\@tempb\endcsname
  }%
  \expandafter\@tempa\@tempb
}
\DeclareOptionX*{%
  \PackageWarning{etaremune}{Unknown option `\CurrentOption'}%
}
\ProcessOptionsX
\define@cmdkey[EM]{etaremune}[EM@]{start}{}
\presetkeys[EM]{etaremune}{start=-1}{}
```

Two counters needed in the package.

```
\newcounter{EM@itemctr}
\newcounter{EM@etaremunectr}
```

`etaremune` The main code.

```
\newenvironment{etaremune}[1][]{%
```

There is a (usual) maximum to nesting.

```
  \ifnum\@enumdepth>\thr@@\@toodeep\else
    \setkeys*[EM]{etaremune}{#1}%
    \advance\@enumdepth\@ne
```

Count the environments. Each environment gets a unique identification to be used to define macros containing the number of items and to set starting points on the second run.

```
    \stepcounter{EM@etaremunectr}%
```

Backup the number of items that we have seen so far in a higher level. We reinitialize the items counter when we get back from the nested environment. Unfortunately, locally counting items is not possible as LaTeX uses a box to typeset the item label. Otherwise, this macros would not be necessary.

```
    \edef\EM@currnum{\the\c@EM@itemctr}%
```

This is a backup of the current level list number. This is used to be able to identify the current list at the end of the list taking into account possibly nested lists in this list.

```
    \edef\EM@currlist{\romannumeral\c@EM@etaremunectr}%
```

Define the current level list counter.

```
    \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
```

Start counting items.

```
    \setcounter{EM@itemctr}\z@
```

Initialize the step size to count backwards.

```
    \let\EM@step\m@ne
    \ifnum\EM@start<\z@
```

If we have no optional input, check whether the macro for the current list has been defined.

```
      \@ifundefined{etaremune@\EM@currlist}{%
```

If the macro is not defined, we count forwards (like `enumerate`). Define the macro to produce a warning. This warning will be issued `\AtEndDocument` to notify the user that the document should be rerun.

```
        \let\EM@step\@ne
        \setcounter\@enumctr\m@ne
        \gdef\EM@rerun{%
          \@latex@warning@no@line{Etaremune labels have changed.^^J
          \@spaces\@spaces\@spaces\space\space Rerun to get them right}%
        }%
      }{%
```

If the macro is defined, it contains the number of items of the current list at this depth. Use that to initiate the label counter.

```
        \setcounter\@enumctr{\csname etaremune@\EM@currlist\endcsname}%
      }%
```

If there was input, use that to set the starting point.

```
    \else\setcounter\@enumctr\EM@start\fi
    \stepcounter\@enumctr
```

Start the list.

```
    \list{%
```

At every item, step the label counter, check whether it is smaller than 0, increase the items counter by 1 and define the current label for referring to items.

```
      \addtocounter\@enumctr\EM@step
      \ifnum\csname c@\@enumctr\endcsname<\z@\setcounter\@enumctr\z@\fi
      \stepcounter{EM@itemctr}%
      \xdef\@currentlabel{%
        \csname p@\@enumctr\endcsname\csname the\@enumctr\endcsname
      }%
```

Typesets the item label.

```
      \csname label\@enumctr\endcsname
```

Define how to align the label. This is the same as for the `enumerate` environment.

```
    }{\def\makelabel##1{\hss\llap{##1}}\setrmkeys[EM]{template}}%
  \fi
}{%
```

The end of the environment. End the list.

```
  \endlist
```

Write the number of items of the current list to the auxiliary file.

```
  \immediate\write\@auxout{\string\gdef\expandafter\string
    \csname etaremune@\EM@currlist\endcsname{\the\c@EM@itemctr}%
  }%
```

Restore the items counter to continue counting the number of items in the environment at this level. This is needed when lists were nested.

```
  \setcounter{EM@itemctr}\EM@currnum
}
```

Issue a warning when item labels haven't settled yet.

```
\AtEndDocument{\EM@rerun}
\let\EM@rerun\relax
⟨/etaremune⟩
```

References

[1] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX Companion, Second Edition.* Addison-Wesley, 2004.

[2] Jörn Wilms. revnum package, v1.0, 1997/05/10. CTAN:/macros/latex/contrib/revnum.

Version history

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.