moreenum

Seamus Bradley

November 3, 2011

This package provides more enumeration styles for \texttt{enumerate} environments. The styles are supposed to work with \texttt{enumitem}. This is moreenum version 1.03.

1 Basic procedure

At the heart of each new enumeration is the following procedure:

\begin{verbatim}
\newcommand*{\macro}[1]{% 
  \expandafter\@macro\csname c@#1\endcsname}
\newcommand*{\@macro}[1]{% 
  \translate{#1}}
\AddEnumerateCounter{\macro}{\@macro}{\distance}
\end{verbatim}

From a user perspective, \texttt{\macro} takes a counter as its argument and outputs, say, a binary number or whatever you want. Actually, what it really does is turn a counter into a number and pass the number to \texttt{\@macro} which does the real work. It takes a number and translates it into the final representation.

Most of the cleverness is done by \texttt{\translate} and these are mostly macros I’ve borrowed from other packages.

The distance is the widest entry in the enumeration. moreenum hasn’t been tested much with this parameter: I’ve just guessed a bit at what’s the widest enumerations are likely to get. Enumerations can \emph{theoretically} get up to 2147483647 items long. Which would be rather a long number.\footnote{\texttt{fmcount} doesn’t seem to work with numbers that big, actually. But even 131071 is 1111111111111111}.

The \texttt{\greek} macro is a little more involved because it involves first defining a macro that turns numbers into Greek letters.

\begin{verbatim}
\newcommand*{\single@greek}[1]{% 
  \expandafter\@single@greek\csname c@#1\endcsname}
\end{verbatim}
Then you need to define what to do when you run out of letters. You start again at $\alpha \alpha$. The clever work there is done by the alphalph package.

Some sophistication is required to get the \texttt{\LaTeX}-style macros to play nice with \texttt{\label} and \texttt{\ref} facilities. This can be seen in the following example.

The \texttt{\protect} makes sure the \texttt{Hexadecimalnum} get written to the \texttt{.aux} file, rather than expanded first. The \texttt{\number} makes sure the number \textit{is} written to the \texttt{.aux} file.\footnote{I'm actually guessing here. I have no idea. I got the clue from egreg here: http://tex.stackexchange.com/q/22234/215}

## 2 Limitations

The biggest number TeX can handle is 2147483647. I can't imagine this ever being a serious limitation to your enumerating.

There are, however, some further limitations. Certain \texttt{fmtcount} macros seem to break before they hit this fundamental limit. In brackets are the \texttt{moreenum}-defined enumerations affected.

- \texttt{\binary} and friends break at 131072 [\texttt{\enumbinary}]
- \texttt{\hexadecimal} and friends break at 1048576 [\texttt{\enumhex} and \texttt{\enumHex}]
- \texttt{\numberstring} and friends break at 100000 [\texttt{\nwords}, \texttt{\nthwords} and friends]
None of these is a serious limitation. If you desperately need bigger enumerations, they are fairly straightforward to define yourself using `binhex` for the numbers and `numname` for the words: these packages don’t have these limitations.³

3 Examples of the enumerations

Here are examples of all the kinds of enumeration that the package defines. The first item contains a reference to the third. This is to test if the referencing is working. The labels have dots after them, to check whether errant spaces are being added after the labels.⁴

\texttt{\greek}

\begin{itemize}
  \item \textit{\greek} Liberty: \textit{\greek}.
  \item \textit{\greek} Equality
  \item \textit{\greek} Fraternity
  \item \textit{\greek} Meanin\textit{\greek} of life
\end{itemize}

\texttt{\Greek}

\begin{itemize}
  \item \textit{\Greek} Liberty: \textit{\Greek}.
  \item \textit{\Greek} Equality
  \item \textit{\Greek} Fraternity
  \item \textit{\Greek} Meanin\textit{\Greek} of life
\end{itemize}

\texttt{\enumHex}

\begin{itemize}
  \item \texttt{\enumHex} 1. Liberty: 3.
  \item \texttt{\enumHex} 2. Equality
  \item \texttt{\enumHex} 3. Fraternity
  \item \texttt{\enumHex} 2A. Meaning of life
\end{itemize}

\texttt{\enumhex}

\begin{itemize}
  \item \texttt{\enumhex} 1. Liberty: 3.
  \item \texttt{\enumhex} 2. Equality
  \item \texttt{\enumhex} 3. Fraternity
  \item \texttt{\enumhex} 2a. Meaning of life
\end{itemize}

\texttt{\enumBinary}

\begin{itemize}
  \item \texttt{\enumBinary} 1. Liberty: 11.
  \item \texttt{\enumBinary} 10. Equality
  \item \texttt{\enumBinary} 11. Fraternity
  \item \texttt{\enumBinary} 101010. Meaning of life
\end{itemize}

\texttt{\enumOctal}

\begin{itemize}
  \item \texttt{\enumOctal} 1. Liberty: 3.
  \item \texttt{\enumOctal} 10. Equality
  \item \texttt{\enumOctal} 11. Fraternity
  \item \texttt{\enumOctal} 52. Meaning of life
\end{itemize}

³Why don’t I just use those packages instead? Because having \texttt{fmtcount} do most of the work means only loading one package instead of 3 (\texttt{numname}, \texttt{binhex} and \texttt{nth} or \texttt{engord}). Also, \texttt{fmtcount} can speak different languages, and in future releases I’m tempted to try to get that working here.

⁴Thanks to Kevin Klement for pointing this issue out to me.
raisenth
1st. Liberty: 3rd.
2nd. Equality
3rd. Fraternity
42nd. Meaning of life

Nthwords
First. Liberty: Third.
Second. Equality
Third. Fraternity
Forty-Second. Meaning of life

NTHWORDS
FIRST. Liberty: THIRD.
SECOND. Equality
THIRD. Fraternity
FORTY-SECOND. Meaning of life

nthwords
first. Liberty: third.
second. Equality
third. Fraternity
forty-second. Meaning of life

levelnth
1st. Liberty: 3rd.
2nd. Equality
3rd. Fraternity
42nd. Meaning of life

Nwords
One. Liberty: Three.
Two. Equality
Three. Fraternity
Forty-Two. Meaning of life

NWORDS
ONE. Liberty: THREE.
TWO. Equality
THREE. Fraternity
FORTY-TWO. Meaning of life

nwords
one. Liberty: three.
two. Equality
three. Fraternity
forty-two. Meaning of life