The shortlst Package*

© Mogens Lemvig Hansen
mlhansen@uniserve.com

1998/11/27

Abstract

The shortlst package provides environments similar to itemize and enumerate designed especially for lists of short items.

Contents

1 The Basics 1
2 The Parameters 3
3 The Limitations 4
4 The Alternatives 5
5 The Copyright 5
6 The Installation 5
7 The Driver 6
8 The Implementation 6
  8.1 The run-in lists ........................................... 7
  8.2 The short-lists ........................................... 8

1 The Basics

When you have a list of short items, the regular itemize environment leaves

• a lot
• of
• white
• space.
The obvious alternative is to use a \texttt{tabular}, but tables are difficult to type and harder to edit if you decide to interchange two items. This package allows you to type input very similar to \texttt{itemize} input.

\begin{shortitemize}
\item the \texttt{itemize} environment
\item leaves
\item a lot
\item of
\item white space
\end{shortitemize}

The code above yields output like\footnote{This file describes version 1.1, 1998/11/27, which is a bug-fix release containing no new functionality since version 1.0.} this:

- the \texttt{itemize} environment
- leaves
- a lot
- of
- white space

Above, \LaTeX{} decided that the first item needs twice as much space as the other items. You may give an optional argument to the \texttt{shortitemize} environment specifying the width of the default allotment of space.

\begin{shortitemize}[the \texttt{itemize} environment]
\item the \texttt{itemize} environment
\item leaves
\item a lot
\item of
\item white space
\end{shortitemize}

Now each item gets as much space as the first.

- the \texttt{itemize} environment
- leaves
- a lot
- white space

You may not put any list environment inside a \texttt{shortitemize} environment—but a \texttt{shortitemize} environment can be (part of) an item of a regular list environment. If you put a \texttt{shortitemize} environment inside a regular \texttt{itemize} environment, \texttt{shortitemize} will use the next level of labels. You may use the optional argument to \item to override the label—just as for the regular \texttt{itemize} environment.

The \texttt{shortlist} package also provides a \texttt{shortenumerate} environment. You may use \texttt{shortenumerate} in the same way as \texttt{shortitemize}, \textit{mutatis mutandis}.

\begin{enumerate}
\item An item.
\item This item contains a sub-list:
  \begin{shortenumerate}
  \item One
  \item[\%] One and a half
  \item Two
  \item \label{here}Three
  \end{shortenumerate}
\item Back in the outer list.
\end{enumerate}

\footnote{All “output” in this document is simulated since you probably did not install the \texttt{shortlist} package before \LaTeX{}ing the documentation. The \textit{real} output is nicer.}
The code above yields this output:

1. An item.

2. This item contains a sub-list:

   (a) One % One and a half (b) Two
   (c) Three

3. Back in the outer list.

Now you may use \ref{here} to refer to item 2(c) in the usual manner.

Some lists do not deserve a displayed paragraph. For such you may use the runenumerate environment.

You have three choices:
\begin{runenumerate}
\item wash you hands,
\item postpone it until tomorrow, or
\item \label{choice}stay dirty.
\end{runenumerate}
I choose \ref{choice}!

Here is then the output:

You have three choices: 1. wash you hands, 2. postpone it until tomorrow, or 3. stay dirty. I choose 3!

The main advantage of the runenumerate environment is the automatic (and nested) numbering and cross references. For completeness, the shortlist package also provides a runitemize environment.

2 The Parameters

The length \runitemsep stores the space between items of a runenumerate or runitemize environment. Here is the default value:

\begin{verbatim}
\setlength{\runitemsep}{1em plus .5em minus .5em}
\end{verbatim}

\labelsep All four environments, shortitemize, shortenumerate, runitemize, and runenumerate use \labelsep (from the L\TeX{} kernel, see [6, p. 113], [4, p. 62], or [1]) for the space between the label and the item itself. The two short-list environments use the same space as the minimal space between one item and the next label.

\labelwidth The labels of the shortitemize and shortenumerate environments may overlap the previous item if the length \labelwidth (from the L\TeX{} kernel, see [6, p. 113], [4, p. 62], or [1]) is too small.

\shortitemwidth Instead of giving optional arguments to each shortitemize and shortitemize environment, you may change the length \shortitemwidth, which stores the default width of each item (without the label and a bit of space on each side).

\begin{verbatim}
\setlength{\shortitemwidth}{65pt}
\end{verbatim}
Inside a `shortitemize` or `shortenumerate` environment \shortitemwidth is the width of the optional argument to the environment. Thus you may use \parbox[t]{\shortitemwidth}{(paragraph)} to typeset a long item.

\begin{shortenumerate}
\item A short item
\item \parbox[t]{\shortitemwidth}{A little paragraph that\footnote{To parbox} will be too long to fit on one line no matter what\strut.}
\item Another short
\item \newlength{\mylength} \setlength{\mylength}{2\shortitemwidth}\% \
addtolength{\mylength}{2\labelsep}\% \
addtolength{\mylength}{\labelwidth}\% \
\begin{minipage}[t]{\mylength}
A little paragraph that\footnote{To minipage} will be too long to fit on one line no matter what.
\end{minipage}
\item Bla
\item Bla
\end{shortenumerate}

The \strut on the last line of the \parbox improves the spacing between the \parbox and the following line.

1. A short item 2. A little paragraph 3. Another short that\footnote{To minipage} will be too long to fit on one line no matter what.

4. A little paragraph that\footnote{To minipage} will be too long to fit on one line no matter what.

5. Bla

6. Bla

### 3 The Limitations

- The `shortitemize` and `shortenumerate` environments cannot contain any list environment. These include `shortitemize`, `shortenumerate`, `itemize`, `enumerate`, `description`, `quote`, `quotation`, `verse`, `center`, `flushleft`, `flushright`, `verbatim`, `tabbing`, `trivlist`, `list`, and all environments made with \newtheorem.

- All four environments handle footnotes\footnote{To parbox}; however, the `shortitemize` and `shortenumerate` environments do not handle other types of floats.

- The `shortitemize` and `shortenumerate` environments cannot handle items that are longer than one line; use \parbox or the `minipage` environment—with the implications that has on footnotes.

\footnote{To parbox}
\footnote{But \verb is ok.}
\footnote{In the case of `shortitemize` and `shortenumerate` due to a technique found in the `tabularx` package by David Carlisle [2, code lines 121–127].}
• The \shortlst package does not define a version of the description environment because I do not know what it should do.

• You cannot use the \label command if your shortenumerate or runenumerate environment is nested inside a \tabularx environment.

• All four environments defined by the \shortlst package make (parts of) paragraphs not boxes. If you want your list centred, you must therefore include it in a \parbox (or minipage) and centre the box.

4 The Alternatives

multiple columns You may use the regular itemize and enumerate environments in a multi-column format. The \multicol package \cite{multicol} creates multiple columns very nicely. The items end up ordered vertically where this package orders them horizontally.

tabular You can put your items in a table—but such are difficult to edit and you do not get automatic numbering and cross references.

multienum The \multienum package \cite{multienum} provides an environment similar to shortenumerate. However, the input syntax is more similar to tabular than to enumerate.

paralist The \paralist package \cite{paralist} provides environments similar to runitemize and runenumerate except that nesting inside itemize or enumerate is not considered.

5 The Copyright

• You may use this software.

• You may copy this software and the documentation for your own use.

• You may distribute the file \shortlst.dtx to others provided you do not make a profit doing so.

• You may modify my code and the documentation—including incorporating it into your own work—provided you do not make a profit on my work or allow others to.

6 The Installation

• Run the file \shortlst.dtx through \LaTeX (thrice to resolve cross references\footnote{If you want an index you must run MakeIndex (makeindex \texttt{-s gind.ist} \texttt{shortlst}) between the second and third \LaTeX run.}).

  \texttt{latex shortlst.dtx}
• Run the file `shortlst.ins` which now exists through \TeX.

```latex
latex shortlst.ins
```

• You now have to decide what to do with several files.

  – You may now have to move the file `shortlst.sty` to some directory where \TeX{} can find it; `texmf/tex/latex/misc` would be the natural choice [9].
  – Move the documentation, `shortlst.dvi`, to `texmf/doc/latex/misc`.
  – You may discard the source file, `shortlst.dtx`, or store it in the directory `texmf/source/latex/misc`.
  – Discard all remaining `shortlist.*` files.

7 The Driver

The following few lines of code allow \TeX{} to generate the `.sty` file and the documentation from this source. First, a `filecontents` environment creates the installation script, `shortlst.ins`, for DocStrip.

```latex
\begin{filecontents}{shortlst.ins}
\input docstrip.tex
\preamble
The shortlst Package
Copyright 1998 by Mogens Lemvig Hansen
mlhansen@uniserve.com
\endpreamble
\usedir{tex/latex/misc}
\generate{\file{shortlst.sty}\{from{shortlst.dtx}{package}}}
\endbatchfile
\end{filecontents}
```

Then the driver to typeset the documentation.

```latex
\documentclass{ltxdoc}[1996/01/11]
\CodelineIndex\EnableCrossrefs
\begin{document} \DocInput{shortlst.dtx} \end{document}
\end{filecontents}
```

8 The Implementation

Declare the package name. I made this package with the December 1997 version of \TeX{}; it may work with earlier versions.
8.1 The run-in lists

\runitemsep
Run-in items are separated by the length \runitemsep. The same amount of space is also placed before the first item and after the last.

\newlength{runitemsep}
\setlength{runitemsep}{1em plus .5em minus .5em}

\runitemize
The runitemize environment first checks if the current level of nesting of itemize-like environments is shallow enough to proceed.

\newenvironment{runitemize}{%}
\ifnum \@itemdepth >\thr@@ \@toodeep \else
\advance\@itemdepth \@one

Define the default label, \@itemlabel, as \labelitem\langle i \rangle which expand to the bullet, dash, etc. used by itemize.

\edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
\def\@itemlabel\csname\@itemitem\endcsname%

The regular \item command calls \@item, so make it use a custom version.

\let\@item\run@item
\let\par\relax
\ignorespaces\fi}

At the end of the runitemize environment you must remove the space (if any) between the last item and the \end{runitemize}; replace it by \runitemsep amount of space. Then ignore spaces following the \end{runitemize}.

\unskip\@skip\runitemsep\ignorespacesafterend

\runenumerate
The definition of the runenumerate environment is similar to that of the runitemize environment.

\newenvironment{runenumerate}{%}
\ifnum \@enumdepth >\thr@@ \@toodeep \else
\advance\@enumdepth \@one

The call to \usecounter{enum\langle i \rangle} sets \if@nmbrlist\listctr\@listctr\ifnum\listctr\@listctr\false and defines \@listctr as \enum\langle i \rangle which is the counter used by the enumerate environment at this level.

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

Define the default label, \@itemlabel, as \labelenum\langle i \rangle.

\edef\@itemlabel\csname label\\@enumctr \endsname\endcsname%
\let\@item\run@item
\let\par\empty
\ignorespaces\fi}
\unskip\@skip\runitemsep\ignorespacesafterend
\run@item  The regular \item command checks if it has an optional argument, stores the answer in \if@noitemarg, and calls \item or \item[@itemlabel] as appropriate. The \run@item command is the custom version of \item for the run-in list environments.

First remove the space (if any) between the last item and this one; replace it by \runitemsep amount of space. However, this could be the first item of a list which is the first object in a paragraph. In that case, just begin a new paragraph.

\begin{verbatim}
def\run@item[#1]{% 
  \ifhmode\unskip\hspace{runitemsep}\else\leavevmode\fi \if@noitemarg \@noitemargfalse \if@nmbrlist\refstepcounter{\@listctr}\fi \fi \setbox{\@itemitem}{#1}\kern{\labelsep}\ignorespaces}
\end{verbatim}

If the \item had no optional argument and if this is a numbered list, increment the counter.

\begin{verbatim}
  \if@noitemarg \@noitemargfalse \if@nmbrlist\refstepcounter{\@listctr}\fi \fi \setbox{\@itemitem}{#1}\kern{\labelsep}\ignorespaces\end{verbatim}

8.2 The short-lists

\shortitemwidth  The idea is to typeset each item in a box of width “a multiple of a fixed length.” \TeX{} will then stack the boxes so that they line up nicely. The default “fixed length” is \shortitemwidth + \labelwidth + 2\labelsep (one \labelsep between the label and the item; another \labelsep between this item and the next label). The default value is rather arbitrary.

\begin{verbatim}
def\shortitemwidth{65pt}
\end{verbatim}

\shortitemize  The default value of the optional argument to the \shortitemize environment is a box of width \shortitemwidth.

\begin{verbatim}
def\ifnum|\itemdepth|>\thr@@|\toodeep|{\setlength\shortitemwidth{|\esh@rtitem\relax|\else|\setlength\shortitemwidth{|\advance\@itemdepth\@ne|\edef\@itemitem{\romannumeral\the\@itemdepth}|\fi|\edef\@itemitem{|\labelitem\romannumeral\the\@itemdepth}\fi}\settowidth\shortitemwidth{#1}\list{}{|\advance\leftmargin\itemindent|\addtolength\leftmargin{-\labelwidth}|\addtolength\leftmargin{-\labelsep}\\}
\end{verbatim}

Measure the width of the optional argument. Since short-list environments cannot be nested anyway, you may store the value locally in \shortitemwidth.

\begin{verbatim}
def\shortitemwidth{#1}\\
\end{verbatim}

Use the list environment to create a displayed paragraph for the short-list. If the end-user has mismatched environments, \TeX{} should not mention a list environment from the workings of this code, so use \list and \endlist instead of \begin{list} and \end{list}.

In the itemize environment labels stick out in the margin; to get things to line up here, it is easier to move the left margin instead.

\begin{verbatim}
\list{}{|\addtolength\leftmargin{\itemindent}|\addtolength\leftmargin{-\labelwidth}\\
\addtolength\leftmargin{-\labelsep}\\
\end{verbatim}
You cannot have any indentation inside the `shortitemize` environment.

\item The \texttt{item} command gets the list environment started. Then it is safe to set up a custom version of \texttt{item} and a \texttt{raggedright} style right-hand margin. The \texttt{endshortitem} command is explained below.

\item \noindent
\item \let \item \sh@rtitem
\item \let \enditem \sh@rtitem
\item \def \@itemlabel \{
\item \relax
\item \@rightskip \@flushglue \rightskip \@rightskip
\item \let \endshortitem \noindent
\item \let \item \sh@rtitem
\item \usecounter \{
\item \def \@itemlabel \{
\item \fi \fi \ignorespaces }%
\item \fi \fi \ignorespaces }%
\item \endshortitem \endlist

At the end of the `shortitemize` environment you must process the last item and close the list environment.

\item \endshortitem \endlist

\texttt{shortenumerate} The definition of the `shortenumerate` environment is straightforward once you comprehend the `runenumerate` and `shortitemize` environments.

\item \newenvironment \{shortenumerate\}[1] \{
\hbox to \shortitemwidth \{ \hfil \}
\} \{ \item \relax
\item \@rightskip \@flushglue \rightskip \@rightskip
\item \let \endshortitem \noindent
\item \let \item \sh@rtitem
\item \usecounter \{
\item \def \@itemlabel \{
\item \fi \fi \ignorespaces }%
\item \fi \fi \ignorespaces }%
\item \endshortitem \endlist

Since each item is typeset inside a box, \TeX{} gobbles up all footnotes. David Carlisle has solved that problem in his `tabularx` package [2, code lines 122–127]. To require all of that package every time seems a bit much, so just copy those lines of code if needed. These special versions of \texttt{@footnotetext} and \texttt{@xfootnotetext} store appropriate \texttt{footnotetext} commands in the token list \texttt{TX@ftn} for later processing.

\item \ifx \TX@ftn \undefined
\item \newtoks \TX@ftn
\item \long \def \TX@ftntext \#1 \{%
\item \edef \@tempa \{ \the \TX@ftn \noexpand \footnotetext
\item \[ \the \csname c@\@mpfn \endcsname \]
\item \global \TX@ftn \expandafter \{
\item \the \TX@ftn \footnotetext \ (#1) \}
\item \global \TX@ftntext \#2 \{
\item \global \TX@ftn \expandafter \{
\item \the \TX@ftn \footnotetext \[#1\]
\item \hfill \ignorespaces }%
\item \fi
\item \fi
\shortitem You need to capture each item in a box. Thus each \item must close and process the previous box before it begins capturing the next item. The custom version of \item therefore first calls \endshortitem to finish off the last item, then checks for an optional argument, stores the answer in \@noitemmargin, and calls \shortitem or \shortitem[@itemlabel] as appropriate. Here \@itemlabel is the default label.

97 \def\shortitem{%
98 \endshortitem
99 \@inmatherr\item
100 \@ifnextchar \[\shortitem{\@noitemmargintrue \shortitem[@itemlabel]}}
101 \shortitem
102 \endshortitem

The \shortitem command first (re-)defines \endshortitem. An \lrbox environment captures the label and the text of the item in the box \@tempboxa, so close that environment—only use \begingroup\lrbox and \endlrbox\endgroup instead of \begin{lrbox} and \end{lrbox} as for \list above (the extra group is necessary due to the convoluted workings of \lrbox [3, code lines 69–80]).

103 \def\shortitem[#1]{%
104 \def\endshortitem{\endlrbox\endgroup%

Measure the width of the box—that is, the width of the label, the space between the label and the item, and the item itself. Let \( a \) be that width plus one \labelsep (for the separation between this item and the next label).

105 \setlength\@tempdima{\wd\@tempboxa}%
106 \addtolength\@tempdima\labelsep

Then let \( b \) be the “fixed length;” that is, \( \labelwidth + 2\labelsep + \shortitemwidth \).

108 \addtolength\@tempdimb\shortitemwidth
109 \addtolength\@tempdimb\labelsep
110 \addtolength\@tempdimb\labelwidth

Then calculate how many multiples of \( b \) you need to contain \( a \); that is, calculate \( \lceil \frac{a}{b} \rceil \). However, \TeX does not provide a ceiling function, so use \TeX’s integer division command, which (for positive integers) calculates \( \lfloor \frac{a}{b} \rfloor \), and add one. The result is fine except when \( b \) divides \( a \) perfectly—which could happen if the item is a \parbox{\shortitemwidth}, so cheat: decrease \( a \) by A Very Small Length first.

112 \addtolength\@tempdimb{-1sp}% subtract a Very Small Length
113 \divide\@tempdima by \@tempdimb%
114 \addtolength\@tempdimb\labelsep%
115 \multiply\@tempdimb by \@tempdimb%

Now \( b \) is the desired width of the box. Instead of actually making a box of that width, typeset the box as is followed by a suitable amount of space (which will then nicely disappear if it happens to land on the right-hand margin).

117 \addtolength\@tempdimb{-\wd\@tempboxa}%
118 \usebox\@tempboxa

Now \( b \) is the amount of space which you must remember to set once the footnotes have been processed.

The following line of code (from [2, code line 50]) inserts the footnotes (or, rather, \footnotetext-s) that were collected inside the box. The \expandafter trickery causes \TeX to first clear the token list \TX@ftn, then execute the \footnotetext commands that it contained. This convoluted order ensures that
the \footnotetext-s go back into \texttt{\expandafter\the\texttt{\expandafter}\the\texttt{\expandafter} of course \texttt{tabularx} can typeset them in due time).
\global\texttt{\expandafter{\expandafter}\the\texttt{\expandafter} if the short-list is nested inside a \texttt{tabularx} environment (so that \texttt{tabularx} can typeset them in due time).

That was the end—now back to the beginning. First ref-step the counter if appropriate.
\texttt{\if@noitemarg \@noitemargfalse \if@nmbrlist\refstepcounter{\@listctr}\fi \fi}

Issue a warning if the label is too wide—it may overlap the previous item.
\texttt{\settowidth{\tempdima}{#1}% \ifdim\tempdima>\labelwidth\PackageWarning{shortlst}{}{label too wide \set \string\labelwidth\space to at least \the\tempdima}}\fi

Then begin collecting an lrbox. Once inside the box, use the custom footnotes and set the label and some space before the item itself.
\texttt{\begingroup\lrbox{\@tempboxa}% \let\@footnotetext\texttt{\expandafter{\expandafter}\the\texttt{\expandafter} \let\@xfootnotenext\texttt{\expandafter{\expandafter}\the\texttt{\expandafter} \makebox[\labelwidth][r]{#1}% \hspace{\labelsep}\ignorespaces}

\texttt{\sh@rtnesterr That’s it! You just need a single custom error message. \newcommand{\sh@rtnesterr}{\PackageError{shortlst}{}{nested short-lists} \{don’t nest short-list environments within each other}} \endgroup}

References


[5] Dennis Kletzing. \textit{A Multienumerate Package}. CTAN. The author does not provide a version number.

