

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	5
4	The Alternatives	5
5	The Copyright	6
6	The Installation	6
7	The Driver	6
8	The Implementation	7
8.1	The run-in lists	7
8.2	The short-lists	9

1 The Basics

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

- a lot
- of

*This file describes version 1.1, 1998/11/27, which is a bug-fix release containing no new functionality since version 1.0.

- white
- space.

`shortitemize` The obvious alternative is to use a `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 `itemize` input.

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

The code above yields output like¹ this:

- the `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 `shortitemize` environment specifying the width of the default allotment of space.

```
\begin{shortitemize}[the \textsf{itemize} environment]
```

Now each item gets as much space as the first.

- the `itemize` environment
- leaves
- a lot
- of
- white space

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

`shortenumerate` The `shortlst` package also provides a `shortenumerate` environment. You may use `shortenumerate` in the same way as `shortitemize`, *mutatis mutandis*.

```
\begin{enumerate}
\item An item.
\item This item contains a sub-list:
  \begin{shortenumerate}
    \item One
    \item[\%] One and a half
```

¹All “output” in this document is simulated since you probably did not install the `shortlst` package before \LaTeX ing the documentation. The *real* output is nicer...

```

\item Two
\item \label{here}Three
\end{shortenumerate}
\item Back in the outer list.
\end{enumerate}

```

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.

runenumerate 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!

runitemize 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

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

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

\labelsep All four environments, `shortitemize`, `shortenumerate`, `runitemize`, and `runenumerate` use `\labelsep` (from the L^AT_EX 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^AT_EX 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).

`\setlength{\shortitemwidth}{65pt}`

Inside a `shortitemize` or `shortenumerate` environment `\shortitemwidth` is the width of the optional argument to the environment. Thus you may use `\parbox[t]{\shortitemwidth}{\langle paragraph \rangle}` to typeset a long item.

```
\begin{shortenumerate}[A little paragraph]
\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
- that² will be too
- long to fit on one
- line no matter
- what.
4. A little paragraph that^a will be too long
5. Bla
- to fit on one line no matter what.

^aTo minipage

6. Bla

²To parbox

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`³, `tabbing`, `trivlist`, `list`, and all environments made with `\newtheorem`.
- All four environments handle footnotes⁴; 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.
- The `shortlist` 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 `shortlist` 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 [7] 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 [5] provides an environment similar to `shortenumerate`. However, the input syntax is more similar to `tabular` than to `enumerate`.

paralist The `paralist` package [8] provides environments similar to `runitemize` and `runenumerate` except that nesting inside `itemize` or `enumerate` is not considered.

³But `\verb` is ok.

⁴In the case of `shortitemize` and `shortenumerate` due to a technique found in the `tabularx` package by David Carlisle [2, code lines 121–127].

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 references⁵).

```
latex shortlst.dtx
```

- Run the file `shortlst.ins` which now exists through \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 \LaTeX 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 `shortlst.*` files.

7 The Driver

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

```
1 \docstrip{
2 \begin{filecontents}{shortlst.ins}
```

⁵If you want an index you must run `MakeIndex (makeindex -s gind.ist shortlst)` between the second and third \LaTeX run.

```

3 \input docstrip.tex
4 \preamble
5   The shortlst Package
6   Copyright 1998 by Mogens Lemvig Hansen
7   mlhansen@uniserve.com
8 \endpreamble

```

I think that `shortlst.sty` should go into the directory `tex/latex/misc`; however, `DOCSTRIP` will not heed my advise unless your configuration file, `docstrip.cfg`, allows it to.

```

9 \usedir{tex/latex/misc}
10 \generate{\file{shortlst.sty}{\from{shortlst.dtx}{package}}}
11 \endbatchfile
12 \end{filecontents}
13 \</docstrip>

```

Then the driver to typeset the documentation.

```

14 \<docdriver>
15 \documentclass{ltxdoc}[1996/01/11]
16 \CodelineIndex%\EnableCrossrefs
17 \begin{document} \DocInput{shortlst.dtx} \end{document}
18 \</docdriver>

```

8 The Implementation

Declare the package name. I made this package with the December 1997 version of \LaTeX ; it may work with earlier versions.

```

19 \<package>
20 \NeedsTeXFormat{LaTeX2e}[1997/12/01]
21 \ProvidesPackage{shortlst}
22     [1998/11/27 version 1.1]

```

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.

```

23 \newlength\runitemsep
24 \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.

```

25 \newenvironment{runitemize}{%
26   \ifnum \@itemdepth >\thr@@\@toodeep\else
27   \advance\@itemdepth\@ne

```

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

```

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

```

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

```
30 \let\@item\run@item
```

Disable `\par` inside the `runitemize` environment, ignore spaces following the `\begin{runitemize}`, and close the `\ifnum` at the beginning of this definition.

```
31 \let\par\relax
32 \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}`.

```
33 {\unskip\hspace\runitemsep\ignorespacesafterend}
```

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

```
34 \newenvironment{runenumerate}{%
35 \ifnum \@enumdepth >\thr@@\toodeep\else
36 \advance\@enumdepth\@ne
```

The call to `\usecounter{enum<i>}` sets `\ifnmbrlist` true and defines `\@listctr` as `enum<i>` which is the counter used by the `enumerate` environment at this level.

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

Define the default label, `\@itemlabel`, as `\labelenum<i>`.

```
39 \def\@itemlabel{\csname label\@enumctr \endcsname}%
40 \let\@item\run@item
41 \let\par\@empty
42 \ignorespaces\fi}
43 {\unskip\hspace\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.

```
44 \def\run@item[#1]{%
45 \ifhmode\unskip\hspace\runitemsep\else\leavevmode\fi
```

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

```
46 \if@noitemarg
47 \noitemargfalse
48 \ifnmbrlist\refstepcounter{\@listctr}\fi
49 \fi
```

Set the label.

```
50 \mbox{#1}\kern\labelsep\ignorespaces}
```


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.

```
51 \newlength\shortitemwidth
52 \setlength\shortitemwidth{65pt}
```

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

```
53 \newenvironment{shortitemize}[1][\hbox to \shortitemwidth{\hfil}]{%
```

Check the level of nesting and define the default label as for `runitemize` above.

```
54 \ifnum \@itemdepth > \thr@@ \toodeep \let \endsh@rtitem \relax \else
55 \ifx \item \sh@rtitem \sh@rtnesterr \let \endsh@rtitem \relax \else
56 \advance \@itemdepth \@ne
57 \edef \@itemitem {labelitem \romannumeral \the \@itemdepth}%
```

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

```
58 \settowidth\shortitemwidth{#1}%
```

Use the `list` environment to create a displayed paragraph for the short-list. If the end-user has mismatched environments, \LaTeX 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.

```
59 \list{}{\addtolength\leftmargin\itemindent
60 \addtolength\leftmargin{-\labelwidth}%
61 \addtolength\leftmargin{-\labelsep}%
```

You cannot have any indentation inside the `shortitemize` environment.

```
62 \setlength\itemindent\z@}%
```

The `\item` command gets the `list` environment started. Then it is safe to set up a custom version of `\item` and a `\raggedright` style right-hand margin. The `\endsh@rtitem` command is explained below.

```
63 \item \relax
64 \@rightskip \@flushglue \rightskip \@rightskip
65 \let \endsh@rtitem \noindent
66 \let \item \sh@rtitem
67 \def \@itemlabel {\csname \@itemitem \endcsname}%
68 \fi \fi \ignorespaces}%
```

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

```
69 {\endsh@rtitem \endlist}
```

`shortenumberate` The definition of the `shortenumberate` environment is straight forward once you comprehend the `runenumberate` and `shortitemize` environments.

```

70 \newenvironment{shortenumberate}[1][\hbox to \shortitemwidth{\hfil}]{%
71   \ifnum \@enumdepth >\thr@@\toodeep\let\endsh@rtitem\relax\else
72   \ifx\item\sh@rtitem\sh@rtnesterr\let\endsh@rtitem\relax\else
73   \advance\@enumdepth\@ne
74   \edef\@enumctr{enum\romannumeral\the\@enumdepth}%
75   \settowidth\shortitemwidth{#1}%
76   \list{}\{\addtolength\leftmargin\itemindent
77             \addtolength\leftmargin{-\labelwidth}%
78             \addtolength\leftmargin{-\labelsep}%
79             \setlength\itemindent\z@}%
80   \item\relax
81   \@rightskip\@flushglue \rightskip\@rightskip
82   \let\endsh@rtitem\noindent
83   \let\item\sh@rtitem
84   \usecounter{\@enumctr}%
85   \def\@itemlabel{\csname label\@enumctr \endcsname}%
86   \fi\fi\ignorespaces}%
87   {\endsh@rtitem\endlist}

```

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

```

88 \ifx\TX@ftn\undefined
89   \newtoks\TX@ftn
90   \long\def\TX@ftntext#1{%
91     \edef\@tempa{\the\TX@ftn\noexpand\footnotetext
92                  [\the\csname c@\@mpfn\endcsname]}}%
93   \global\TX@ftn\expandafter{\@tempa{#1}}}%
94   \long\def\TX@xftntext[#1]#2{%
95     \global\TX@ftn\expandafter{\the\TX@ftn\footnotetext[#1]{#2}}}
96 \fi

```

`\sh@rtitem` 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 `\endsh@rtitem` to finish off the last item, then checks for
an optional argument, stores the answer in `\@noitemarg`, and calls `\sh@rt@item`
or `\sh@rt@item[\@itemlabel]` as appropriate. Here `\@itemlabel` is the default
label.

```

97 \def\sh@rtitem{%
98   \endsh@rtitem
99   \@inmatherr\item
100   \ifnextchar [\sh@rt@item{\@noitemargtrue \sh@rt@item[\@itemlabel]}

```

`\sh@rt@item` The `\sh@rt@item` command first (re-)defines `\endsh@rtitem`. An `lrbox` environment captures the label and the text of the item in the box `\@tempboxa`, so close that environment—only use `\begin@group\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]).

```
101 \def\sh@rt@item[#1]{%
102   \def\endsh@rtitem{\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).

```
103   \setlength\@tempdima{\wd\@tempboxa}%
104   \addtolength\@tempdima\labelsep
```

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

```
105   \setlength\@tempdimb\shortitemwidth
106   \addtolength\@tempdimb{2\labelsep}%
107   \addtolength\@tempdimb{\labelwidth}%
```

Then calculate how many multiples of b you need to contain a ; that is, calculate $\lceil \frac{a}{b} \rceil$. However, L^AT_EX does not provide a ceiling function, so use T_EX’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.

```
108   \addtolength\@tempdima{-1sp}% subtract a Very Small Length
109   \divide\@tempdima by \@tempdimb%
110   \addtolength\@tempdima{1sp}% add one
111   \multiply\@tempdimb by \@tempdima%
```

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).

```
112   \addtolength\@tempdimb{-\wd\@tempboxa}%
113   \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 T_EX 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 `\TX@ftn` if the short-list is nested inside a `tabularx` environment (so that `tabularx` can typeset them in due time).

```
114   \global\TX@ftn\expandafter{\expandafter}\the\TX@ftn
115   \hspace\@tempdimb}%
```

That was the end—now back to the beginning. First ref-step the counter if appropriate.

```

116 \if@noitemarg
117 \noitemargfalse
118 \if@nmbrrlist\refstepcounter{\@listctr}\fi
119 \fi

```

Issue a warning if the label is too wide—it may overlap the previous item.

```

120 \settowidth\@tempdima{#1}%
121 \ifdim\@tempdima>\labelwidth\PackageWarning{shortlst}%
122 {label too wide
123 (set \string\labelwidth\space to at least \the\@tempdima)}%
124 \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.

```

125 \begingroup\lrbox{\@tempboxa}%
126 \let\@footnotetext\TX@fnttext\let\@xfootnotenext\TX@xftntext
127 \makebox[\labelwidth][r]{#1}%
128 \hspace{\labelsep}\ignorespaces

```

`\sh@rtnesterr` That's it! You just need a single custom error message.

```

129 \newcommand{\sh@rtnesterr}{\PackageError{shortlst}
130 {nested short-lists}
131 {don't nest short-list environments within each other}}
132 </package>

```

References

- [1] Johannes Braams, David Carlisle, Alan Jeffrey, Leslie Lamport, Frank Mittelbach, Tobias Oetiker, Chris Rowley, and Rainer Schöpf. *ltxlists.dtx*. CTAN, 1997/02/21 edition, 1997.
- [2] David Carlisle. *The tabularx Package*. CTAN, 1997/09/18 v2.05 edition, 1997.
- [3] David Carlisle, Leslie Lamport, Frank Mittelbach, and Chris Rowley. *ltxboxes.dtx*. CTAN, 1998/06/05 edition, 1998.
- [4] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1994. ISBN 0-201-54199-8.
- [5] Dennis Kletzing. *A Multienumerate Package*. CTAN. The author does not provide a version number.
- [6] Leslie Lamport. *L^AT_EX, A Document Preparation System, User's Guide and Reference Manual*. Addison-Wesley, second edition, 1994. ISBN 0-201-52983-1.
- [7] Frank Mittelbach. *An environment for multicolumn output*. CTAN, 1997/12/16 v1.5p edition, 1997.
- [8] Bernd Schandl. *paralist, Enumerate and Itemize as and within Paragraphs*. CTAN, 1998/10/11 v1.0 edition.

- [9] TUG Working Group on a TeX Directory Structure (TWG-TDS). *A Directory Structure for TeX Files*. CTAN, version 0.9995, january 26, 1998 edition, 1998.

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.

```

rtitem=      \subitem rtitem+, \main{101} \subitem **\labelwidth+, \subitem rtnesterr+, \main{12}
environments:runenumerate=runenumerate,          shortenumerate=shortenumerate
3,          34 item=      \subitem item+, \main{44} environ-
environments:runitemize=runitemize, runenumerate=runenumerate      ment), 2, 70
3,          25 (environ-          shortenumerate=shortenumerate
environments:shortenumerate=shortenumerate, 3, 34 (environ-          shortitemize=shortitemize
2,          70 runitemize=runitemize      ment), 2, 53
environments:shortitemize=shortitemize,          shortenumerate=shortenumerate
2,          53 (environ-          shortitemwidth=      \subitem **\shortitemwid
3,          25      ment), 3, 25
item=      \subitem **\item+, \usage{2}      \subitem **\runitemsep+, \main{82}
runitemsep=      \subitem **\runitemsep+, \usage{3} \subitem **\subitemsep+, \main{83}
ftntext=      \subitem ftntext+, \main{88}
labelsep=      \subitem **\labelsep+, \usage{3} \subitem rtitem+, \main{97} \subitem xftntext+, \main{88}

```