Stream lists and related list types for LaTeX

Reinhard Wonneberger*
Universität Hamburg

ABSTRACT. As a first step towards a more general concept of lists in LaTeX, 'stream lists' are introduced. While sharing explicit labeling with normal lists, they are set without linebreaks.

On Lists in General. Lists in general have the following advantages: 1. they help with organizing thought; 2. they provide ordered access to a group of items; 3. lists are especially useful with automatic formatting: (a) items can be easily accessed in the file by find or point commands; (b) producing text is supported in the following ways: i. automatic counting helps with modifications; ii. items may be referred to by crossreference tags; iii. the appearance of lists can be changed without touching the items; iv. different list styles can be chosen for proofreading and the final draft.

Normally the internal structure of lists is represented by linebreaks and indents.[1] Since this style improves overall orientation and access at the cost of consuming space and leaving ugly white zones on the page, it is closely related with the field of technical documentation.

In other areas, however, it may be preferable to have *stream lists*. An example of `enumerated` nested stream lists is given in the previous paragraph. While stream lists also represent the logical structure of the list through their explicit label information, they avoid interrupts of the flow of reading and improve the aesthetic appearance of the text.

Closely related to stream lists are *empty lists*, which do not show up in print; producing unlabeled running text from a list environment, they allow the user to keep 'hidden' list structures in his source text.

Looking beyond the scope of this paper, users like me, apart from the possibility of changing individual parameters, should be pleased to have at their disposal, also in LaTeX, the following list types on a *keyword* basis: 1. *spaced*, which is now the default; 2. *compact*, which is like *spaced*, but with normal `\baselineskip` also between items; 3. *stream* and 4. *empty* both described in this article; 5. *par* placing the label at the beginning of an indented paragraph and 6. *hang* doing the same for paragraphs with hanging indentation.

Though a first step is done with the macros presented here, implementing such a general scheme should rather be a challenge for the wizards than a brave but hopeless endeavor for an amateur.

Introducing Stream Lists. Streamlists are derived here from the LaTeX list macros.[2] Though in the standard list macros much can be done by changing parameters in the second argument, streaming is impossible, since `\everypar` is used.

There is a `streamlist`-environment which corresponds to the normal `list`-environment. This environment may be used to specify labeling explicitly, e.g.

```
\newcounter{list}
\begin{streamlist}{(\alph{list})}
          {\usecounter{list}}
\item First element on this level.
\item Second element on this level.
\end{streamlist}.
```

It is also used implicitly by the `streamenumerate`-environment, which corresponds to the normal `enumerate`-environment and can be nested in the known way:

```
\begin{streamenumerate} ...
      \end{streamenumerate}
```

In normal lists, the skip before and after the item label is controlled by the *dimension parameters* `\itemindent` and `\labelsep` resp. Here we use two *commands* instead: `\labelpref` for the label prefix, which is initialized to `\quad`, normally an em-space, and `\labelsuff` for the label suffix, which is initialized to `\enspace`, normally an en-space. Since both of them are commands, they may also be used to surround labels with generated punctuation like square brackets, colons or the like.

---

* *Alttestamentliches Seminar, Sedanstr.19, D 2000 Hamburg 13, FRG.* The macros presented here were developed at *DESY, Notkestr.85, D 2000 Hamburg 54, FRG,* with the aid of P. K. SCHILLING. Comments are welcome to him (R02SCH) or the author (B03WBG) at DHHDESY3 via EARN (which is connected to BITNET).

[1] Correspondence of the logical structure of the text and its graphic representation has a long history of its own and is by no means trivial; in some cases, a lot of linguistic theory is involved. On this subject cf. R. WONNEBERGER: *Normaltext und Normalsynopse. Neue Wege bei der Darstellung alttestamentlicher Texte.* Zeitschrift für Sprachwissenschaft 3 (1984) 203-233.

[2] L. LAMPORT: The LaTeX Document Preparation System. Reading, Massachusetts / Menlo Park, California / London / Amsterdam / Don Mills, Ontario / Sydney: Addison-Wesley (forthcoming).

Macro-Definitions. Stream and empty lists are obtained with the aid of the following definitions:

```
%       \labelpref      :  macro executed right BEFORE an item label.
%       \labelsuff      :  macro executed right AFTER  an item label.

\def\streamlist#1#2%
  {\ifnum \@listdepth >5\relax \@toodeep
       \else \global\advance\@listdepth\@ne \fi
  \csname @list\romannumeral\the\@listdepth\endcsname
  \def\labelpref{\quad}\def\labelsuff{\enspace}
  \def\@itemlabel{#1}\let\makelabel\@stlab \@nmbrlistfalse #2\relax
  \let\@item=\@streamitem
  \ignorespaces}

\def\@stlab#1{{#1}}

\def\@streamitem[#1]{\if@nmbrlist \refstepcounter{\@listctr}\fi
  \labelpref{\makelabel{#1}}\nobreak \labelsuff \ignorespaces}

\def\endstreamlist{\global\advance\@listdepth\m@ne}

\def\streamenumerate{\ifnum \@enumdepth >3 \@toodeep\else
  \advance\@enumdepth \@ne
  \edef\@enumctr{enum\romannumeral\the\@enumdepth}\streamlist
    {\csname label\@enumctr\endcsname}{\usecounter
      {\@enumctr}\def\makelabel##1{{##1}}}\fi}

\let\endstreamenumerate =\endstreamlist

\def\streamempty{\ifnum \@enumdepth >3 \@toodeep\else
  \advance\@enumdepth \@ne
  \streamlist{\relax}{\def\makelabel##1{\relax}\let
  \labelpref=\relax
  \let\labelsuff=\relax}\fi}

\let\endstreamempty =\endstreamlist
```