LIST MACROS

Lynne A. Price

BNR INC.,
subsidiary of Bell-Northern Research, Ltd.

In many document systems, there is a need for lists of indented paragraphs marked by numbers, bullets, letters, or other symbols. This article describes the design of a TEX macro package that provides such a facility with examples of the use of the macros and a listing of their source.

- An author indicates in a TEX input file that a list is to begin with a call to a start-of-list macro. The start-of-list macros indicate the type of list.

  (1) Lists begun with the following macros are marked with numbers or letters enclosed in parentheses:

      (a) `\numberlist` — Arabic numerals
      (b) `\romanlist` — lower-case Roman numerals
      (c) `\ROMANLIST` — upper-case Roman numerals
      (d) `\alphalist` — lower-case letters
      (e) `\ALPHALIST` — upper-case letters

  (2) Lists begun with the following macros are marked with numbers or letters followed by a period:

      a. `\dotnumberlist` — Arabic numerals
      b. `\dotromanlist` — lower-case Roman numerals
      c. `\DOTROMANLIST` — upper-case Roman numerals
      d. `\dotalphalist` — lower-case letters
      e. `\DOTALPHALIST` — upper-case letters

  (3) Lists begun with the following macros are marked with numbers or letters that have no surrounding punctuation:

      a `\nopuncnumberlist` — Arabic numerals
      b `\nopuncromanlist` — lower-case Roman numerals
      c `\NOPUNCROMANLIST` — upper-case Roman numerals
      d `\nopuncalphalist` — lower-case letters
      e `\NOPUNCALPHALIST` — upper-case letters

  (4) Lists begun with the following macros are marked with numbers or letters followed by a

right parenthesis:

a) \closenumberlist — Arabic numerals
b) \closeromanlist — lower-case Roman numerals
c) \CLOSEROMANLIST — upper-case Roman numerals
d) \closealphalist — lower-case letters
e) \CLOSEALPHALIST — upper-case letters

(5) The macro \bulletlist starts a list of items marked by bullets.

(6) The macro \dashlist starts a list of items marked by em-dashes.

(7) The macro \marklist {<mark>} starts a list in which each item is marked by the string specified as <mark>.

- A call to the macro \listitem should precede each item (including the first one) on a list except when the list items occur within tables. This macro inserts the list mark specified by the start-of-list macro, causes the following text to appear in indented blocked paragraphs, and increments numeric and alphabetic marks.

- The user may specify a mark for a particular item by using the macro \markitem {<mark>} instead of \listitem. Lists in which every mark is specified in this fashion may be begun with \startlist instead of one of the start-of-list macros enumerated above. **Warning**: Spaces and carriage returns following macro names are insignificant; in fact, when a macro such as \listitem has no parameters, unless the following text begins with a special character, a space or carriage return is required to delimit the macro name. However, the TEX spacing conventions cause a space or carriage return after the right brace that ends the mark specified with \markitem to result in additional space between the mark and the beginning of the list item. As a result of these conventions the input

```
\listitem
Note the position of the first word here.
```

is equivalent to

```
\listitem Note the position of the first word here.
```

while

```
\listitemNote the position of the first word here.
```

would result in an error, because the macro \listitemNote has not been defined. However, while

```
\markitem{A---}
Note the position of the first word here.
```

is equivalent to

```
\markitem{A---} Note the position of the first word here.
```

the two latter probably do **not** produce what the user intended and are **not** equivalent to the correct form

```
\markitem{A---}Note the position of the first word here.
```

- Lists may occur within other lists—up to four levels of nesting are permitted.

- The macro \listmark can be used within tables to insert the current list mark. Like \listitem, listmark increments numeric and alphabetic marks. It does not, however, affect the current paragraph structure.

- The end of a list is indicated by the macro \endlist.
- The TEX control sequence \par used to indicate the end of paragraphs should not be included before the beginning of a list or before calling \listitem. However, individual items may contain several paragraphs separated by calls to \par.
- **Warning:** A blank line or a line containing only a comment is treated by TEX identically to the control sequence \par. Normally, several adjacent calls to \par are equivalent to a single call, so that one or more blank lines or comment lines can appear between paragraphs. Within lists, however, \par is redefined so that successive paragraphs will have the appropriate indentation. Multiple calls to \par within lists can create unexpected results. Therefore, blank lines and comment lines should be avoided within lists.

These features are illustrated in the following example taken from Acts II and III of *Hamlet*:

```
What a piece of work is a man! How noble in reason!
How infinite in faculty, in form and moving! How express and
admirable in action! How like an angel in apprehension!
How like a god! The beauty of the world! The paragon of animals!
And yet, to me, what is this quintessence of
dust?$\ldots$\numberlist\listitem To be,
or not to be: that is the question. Whether:
\alphalist\listitem
`Tis nobler in the mind to suffer the slings and arrows of
outrageous fortune; or
\listitem To take arms against a sea of troubles,
and by opposing, end them.\endlist
\listitem
To die; to sleep; no more;
\alphalist\listitem
And by a sleep to say we end the heart-ache and the thousand natural shocks
that flesh is heir to.
\listitem
`Tis a consumation devoutly to be wish`d.\endlist\listitem
To die; to sleep;---to sleep?
Perchance to dream! Ay, there`s the rub.
For in that sleep of death what dreams may come, when we have shuffl`d off
this mortal coil, must give us pause. There`s the respect
that makes calamity of so long life.\endlist
$\ldots$ Thus conscience does make cowards of us all; and thus the native hue
of resolution is sicklied o`er with the pale cast of thought, and enterprises
of great faith and moment with this regard their currents turn awry, and lose
the name of action.
```

Figure 1. Nested Lists—Sample Input

What a piece of work is a man! How noble in reason! How infinite in faculty, in form and moving! How express and admirable in action! How like an angel in apprehension! How like a god! The beauty of the world! The paragon of animals! And yet, to me, what is this quintessence of dust?...

(1) To be, or not to be: that is the question. Whether:

> (a) 'Tis nobler in the mind to suffer the slings and arrows of outrageous fortune; or
>
> (b) To take arms against a sea of troubles, and by opposing, end them.

(2) To die; to sleep; no more;

> (a) And by a sleep to say we end the heart-ache and the thousand natural shocks that flesh is heir to.
>
> (b) 'Tis a consumation devoutly to be wish'd.

(3) To die; to sleep;—to sleep? Perchance to dream! Ay, there's the rub. For in that sleep of death what dreams may come, when we have shuffl'd off this mortal coil, must give us pause. There's the respect that makes calamity of so long life.

... Thus conscience does make cowards of us all; and thus the native hue of resolution is sicklied o'er with the pale cast of thought, and enterprises of great faith and moment with this regard their currents turn awry, and lose the name of action.

Figure 2. Nested Lists—Sample Output

The example below shows the use of the list macros in conjunction with TEX's alignment feature (the latter is used to produce tables). Alignment is described in Chapter 22 of the TEX manual.

```
\numberlist
\ctrline{Menu}
\ctrline{June 25, 1980}
\vskip 4ex
\ctrline{\vbox{\halign{\hfill#\quad|#\hfill\cr
\listmark|Tomato Bisque Soup\cr
\listmark|Spinach Salad\cr
\listmark|Tuscan Pot Roast\cr
\listmark|Zucchini Soufl\'e\cr
\listmark|Rice Pilaf\cr
\listmark|Braided Onion Bread\cr
\listmark|Cream Puff Swans\cr
}}}
\endlist
```

Figure 3. Lists Within Tables—Sample Input

Menu
June 25, 1980

(1) Tomato Bisque Soup
(2) Spinach Salad
(3) Tuscan Pot Roast
(4) Zucchini Soufflé
(5) Rice Pilaf
(6) Braided Onion Bread
(7) Cream Puff Swans

Figure 4. Lists Within Tables—Sample Output

The remainder of this article describes details about the list macros that many readers may wish to skip.

- The indentation for all levels of nested lists is determined by the font in effect when the *outermost* list is started. Each level of list is indented by an amount of space equivalent to 4 ems in the initial font. This default may be changed by setting `\varunit <dimen>` where `<dimen>` is the amount of indentation desired. (See Figures 5 and 6 below.)

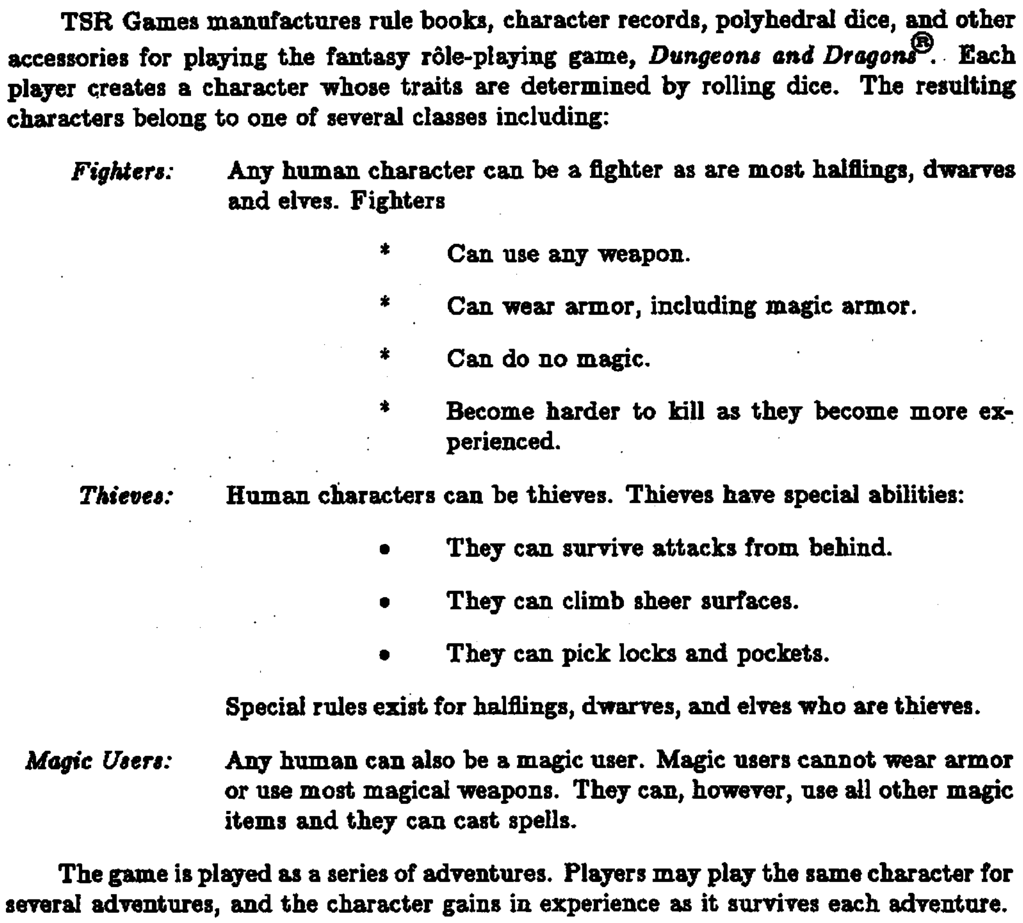
- The list macros use the single "variable unit" provided by TEX. Therefore, the user must not redefine `\varunit` except to change the indentation of list items as described above.

- Since TEX's `\par` control sequence is used to end a paragraph rather than to start a new one, `\par` should not precede the text of list items. Nevertheless, to ensure proper indentation within lists, a macro call is necessary at the beginning of each paragraph. The required spacing information is provided at the beginning of list items by the macros `\listitem` and `\markitem {<mark>}` and by `\par` before succeeding paragraphs. A special case occurs, however, after the `\endlist` that follows a list that appears within another list when the following text is part of the same list item in the outer list as the one that contained the inner list. In this situation, the `\endlist` that ends the inner list should be immediately followed by `\continue`. However, the `\endlist` that ends the outermost list can be immediately followed by the text of the following paragraph. These situations (and the `\varunit` construct discussed above) are illustrated in the following example:

```
\def\trademark{\raise1.4ex\hbox{\spose{\raise 1ex\hbox{\curfont @\char`142}}\!
\hbox{\curfont fR}}}

TSR Games manufactures rule books, character records, polyhedral dice, and
other accessories for playing the fantasy r\A ole-playing game,
{\it Dungeons and Dragons}\trademark.
Each player creates a character whose traits are determined by rolling dice.
The resulting characters belong to one of several classes including:
\startlist
\save9\hbox{\it Magic Users:\quad\qquad} % Set list indentation to the width of
\varunit 1wd9                            % the string "Magic Users: "
\markitem{\it Fighters:}Any human character can be a fighter as are most
halflings, dwarves and elves.  Fighters
\marklist{*}\listitem
Can use any weapon.\listitem Can wear armor, including magic armor.
\listitem Can do no magic.\listitem Become harder
to kill as they become more experienced.\endlist
\markitem{\it Thieves:}Human characters can be thieves. Thieves have
special abilities:\bulletlist\listitem
They can survive attacks from behind.
\listitem They can climb sheer surfaces.
\listitem They can pick locks and pockets.
\endlist % the following is part of list item containing entire preceding list
\continue
Special rules exist for halflings, dwarves, and elves who are thieves.
\markitem{\it Magic Users:}Any human can also be a magic user.
Magic users cannot wear armor or use most magical weapons.
They can, however, use all other magic items and they can cast spells.
\endlist
The game is played as a series of adventures.  Players may play the same
character for several adventures, and the character gains in experience
as it survives each adventure.
```

Figure 5. Use of `\continue`—Sample Input



TSR Games manufactures rule books, character records, polyhedral dice, and other accessories for playing the fantasy rôle-playing game, *Dungeons and Dragons*®. Each player creates a character whose traits are determined by rolling dice. The resulting characters belong to one of several classes including:

Fighters: Any human character can be a fighter as are most halflings, dwarves and elves. Fighters

* Can use any weapon.

* Can wear armor, including magic armor.

* Can do no magic.

* Become harder to kill as they become more experienced.

Thieves: Human characters can be thieves. Thieves have special abilities:

- They can survive attacks from behind.
- They can climb sheer surfaces.
- They can pick locks and pockets.

Special rules exist for halflings, dwarves, and elves who are thieves.

Magic Users: Any human can also be a magic user. Magic users cannot wear armor or use most magical weapons. They can, however, use all other magic items and they can cast spells.

The game is played as a series of adventures. Players may play the same character for several adventures, and the character gains in experience as it survives each adventure.

Figure 6. Use of `\continue`—Sample Output

- Block paragraph structure is used inside lists. At the end of a list, paragraph indentation is reset according to the value specified in the macro `\saveparindent`. Users of standard macro packages need not concern themselves with the latter. Users, however, who design their own formats should know that the default value is 2 ems in the current font. To select another value, use

```
\def\saveparindent{<dimen>}
```

The slide macros, for example, which assume block paragraph structure throughout, include the command

```
\def\saveparindent{0 pt}
```

(here `0` is the number zero not the letter `O`).

- Since each list item is assumed to start a new paragraph, the same amount of vertical space that TEX inserts between paragraphs is inserted between list items. When the list items are very short, this amount of white space is excessive. The space between paragraphs is controlled with the TEX command \parskip <dimen>, where, following Knuth's notation, <dimen> may be any dimension or size. For example, \parskip 0pt completely suppresses extra spacing between list items. Figure 7illustrates how this command was used at the beginning of this article. Note the use of TEX's grouping feature (braces) to indicate that the change in \parskip should have effect only within the inner lists.

```
\numberlist\listitem
Lists begun with the following macros are marked with numbers or letters
enclosed in parentheses:
\alphalist
\listitem{\ty \\numberlist} --- Arabic numerals
{\parskip 0pt
\listitem{\ty \\romanlist} --- lower-case Roman numerals
\listitem{\ty \\ROMANLIST} --- upper-case Roman numerals
\listitem{\ty \\alphalist} --- lower-case letters
\listitem{\ty \\ALPHALIST} --- upper-case letters
\endlist}
\listitem
Lists begun with the following macros are marked with numbers or letters
followed by a period:
\dotalphalist
\listitem{\ty \\dotnumberlist} --- Arabic numerals
{\parskip 0pt
\listitem{\ty \\dotromanlist} --- lower-case Roman numerals
\listitem{\ty \\DOTROMANLIST} --- upper-case Roman numerals
\listitem{\ty \\dotalphalist} --- lower-case letters
\listitem{\ty \\DOTALPHALIST} --- upper-case letters
\endlist}
\endlist
```

Figure 7. Changing the Spacing Between List Items—Samp'

(1) Lists begun with the following macros are marked with numbers or letters enclosed in parentheses:

  (a) `\numberlist` — Arabic numerals
  (b) `\romanlist` — lower-case Roman numerals
  (c) `\ROMANLIST` — upper-case Roman numerals
  (d) `\alphalist` — lower-case letters
  (e) `\ALPHALIST` — upper-case letters

(2) Lists begun with the following macros are marked with numbers or letters followed by a period:

  a. `\dotnumberlist` — Arabic numerals
  b. `\dotromanlist` — lower-case Roman numerals
  c. `\DOTROMANLIST` — upper-case Roman numerals

Figure 8. Changing the Spacing Between List Items—Sample Output

The TEX source of a set of list macros to implement this design is shown below:

```
% List Macros

% \neg and \ifzero from Appendix X
\def\neg#1{\setcount#1-\count#1}
\def\ifzero#1#2\else#3{\ifpos#1{#3}\else{\neg#1
  \ifpos#1{\neg#1 #3}\else{\neg#1 #2}}}
% Arguments to \ifeq can be constants of counters
\def\ifeq#1#2#3\else#4{\setcount9 #1 \advcount9 by -#2
  \ifzero9{#3}\else{#4}}

% Count 9 for scratch, count 8 for list level, count 7 for current item number
% Box 9 for scratch, File 9 for error messages

% \Alph prints the value of the specified counter according to
% the alphabetic sequence A, B, C, ... I.e., if the value of the
% counter is 1, \Alph prints A, etc.  \alph does the same except
% it generates lower case instead of upper case letters.
\def\Alph#1{\setcount9 \count#1 \advcount9 by `100 \char\count9}
\def\alph#1{\setcount9 \count#1 \advcount9 by `140 \char\count9}

% Make \endpar a synonym for TeX`s standard \par control sequence.
% Within a list, \par is redefined to produce an indented paragraph.
\let \endpar=\par

% TeX counter 8 is used to count list indentation level.  \ifcounteight
% executes its ith argument if the current value of counter 8 is i.
% Note that only four levels of list are permitted.
\def \ifcounteight#1#2#3#4{
  \ifeq1{\count8}{#1}
  \else{\ifeq2{\count8}{#2}
     \else{\ifeq3{\count8}{#3}
        \else{\ifeq4{\count8}{#4}\else{}}}}}

% \startpar starts a paragaph inside a list
\def \startpar{\hangindent \count8vu$ $}

% Start a new paragraph in a current list item after the end of a nested
% list
\def \continue{\startpar\hbox to \count8vu{}}

% After a list, paragraph indentation is reset according to \saveparindent
% By default, this value is 2em
\def\saveparindent{2em}

\setcount8 0
\parskip 2ex

% At start of an indented list, save item number of outer list
\def \savecount{
  \ifcounteight{\xdef\savea{\count7}}{\xdef\saveb{\count7}}{\xdef
  \savec{\count7}}{\xdef\saved{\count7}}}

% At end of an indented list, restore item number from previous level
\def \restorecount{
  \ifcounteight{\setcount7 \savea}{\setcount7 \saveb}{\setcount7
  \savec}{\setcount7 \saved}}

% Initially, no current list actve
\def\listitemerror{\send9{LIST ITEM ENCOUNTERED BUT NO CURRENT LIST}}
\def\listmarkerror{\send9{LIST MARK ENCOUNTERED BUT NO CURRENT LIST}}
\def \listitem{\listitemerror}
\def \listmark{\listmarkerror}

% Provide for converting item numbers to upper or lower case roman
\def\roman#1{\setcount9 -\count#1\count9}
\def\Roman#1{\setcount 9
  -\count#1\xdef\uppercaseroman{\uppercase{\count9}}\uppercaseroman}

\def \markitem#1{\setmark{#1}\listitem}
\def \setmark#1{
  \ifcounteight{\gdef \marka{#1}}{\gdef \markb{#1}}{\gdef
    \markc{#1}}{\gdef \markd{#1}}}

% Start of list macros
\def \numberlist{
  \startlist
  \setmark{(\count7)}}
\def \alphalist{
  \startlist
  \setmark{(\alph7)}}
\def \Alphalist{
  \startlist
  \setmark{(\Alph7)}}
\def \romanlist{
  \startlist
  \setmark{(\roman7)}}
\def \Romanlist{
  \startlist
  \setmark{(\Roman7)}}
\def \dotnumberlist{
  \startlist
  \setmark{\count7.}}
\def \dotalphalist{
  \startlist
  \setmark{\alph7.}}
\def \DOTAlphalist{
  \startlist
  \setmark{\Alph7.}}
\def \dotromanlist{
  \startlist
  \setmark{\roman7.}}
\def \DOTRomanlist{
  \startlist
  \setmark{\Roman7.}}
\def \nopuncnumberlist{
  \startlist
  \setmark{\count7}}
\def \nopuncalphalist{
  \startlist
  \setmark{\alph7}}
\def \NOPUNCAlphalist{
  \startlist
  \setmark{\Alph7}}
\def \nopuncromanlist{
  \startlist
  \setmark{\roman7}}
\def \NOPUNCRomanlist{
  \startlist
  \setmark{\Roman7}}
\def \closenumberlist{
  \startlist
  \setmark{\count7)}}
\def \closealphalist{
  \startlist
  \setmark{\alph7)}}
\def \CloseAlphalist{
  \startlist
  \setmark{\Alph7)}}
\def \closeromanlist{
  \startlist
  \setmark{\roman7)}}
\def \CLOSERomanlist{
  \startlist
  \setmark{\Roman7)}}
\def \bulletlist{
  \startlist
  \setmark{$\bullet$}}
\def \dashlist{
  \startlist
  \setmark{\rm ---}}
\def \marklist#1{
  \startlist
  \setmark{#1}}

% Start a list
\def \startlist{
  \advcount8\setcount 9 \count8 \advcount9 by -4 % Compute level number
  \ifpos9{
     \xdef\listerror{ATTEMPT TO START \count8TH LEVEL OF NESTED LISTS ---
       ONLY 4 LEVELS ALLOWED}
     \send9{\listerror}}
  \else{     % Compute indent if first level
    \ifeq1{\count8}{\save9\hbox{M}\varunit 4wd9}\else{}
    \savecount
    \setcount7 0
    \parindent 0pt
    \gdef \listitem{\endpar
      \advcount7
      \startpar\hbox to \count8vu{
        \hfill\ifcounteight{\marka}{\markb}{\markc}{\markd}\unskip\hbox to
         25vu{}}}
    \gdef \listmark{\advcount7\ifcounteight{\marka}{\markb}{\markc}{\markd}}
    \gdef \par{\endpar\startpar\hbox to \count8vu{}}
    }
  }

% End a list
\def \endlist{
  \endpar
  \restorecount
  \advcount8 by -1
  \ifeq{\count8}0{   % If ending outermost list, reset paragraph structure and
    \gdef \listitem{\listitemerror}  % set error messages to be issued if an
    \gdef \listmark{\listmarkerror}  % attempt to specify a list item is made
    \parindent \saveparindent
    \gdef \par{\endpar}
    }
  \else{\setcount9\count9\neg9\ifpos9{\send9{Extra ENDLIST}}\else{}}
  }
```