Macros for Jill

Donald E. Knuth
Stanford University

At the TUG meeting in July, 1986, I mentioned in conversation that one of my new household duties was to write macros for my wife Jill, who had just installed TeX on her PC. Later, when Jill came to the dinner party, many people asked her for copies of the macros; and this led eventually to the idea that I should publish them in *TUGboat*. So here they are, slightly cleaned up from the way I originally wrote them.

The first task Jill assigned me was perhaps the most interesting. She had started to keep an electronic journal, and she wanted to make a nice hardcopy book. The format she had in mind was somewhat tricky because she wanted to be able to generate marginal notes in the middle of any paragraph. Furthermore, she wanted these notes to go in the left-hand margin on left-hand pages and in the right-hand margin on right-hand pages.

This task is difficult for TeX, because TeX generates paragraphs before it knows what page they will go on. Indeed, the decision about what to put on page 100 may not be made until TeX has generated a good deal of page 101.

One way to solve the problem would be to cheat, by putting the notes in both margins and masking off the undesired ones. Jill didn't like that idea very much.

A legitimate solution can be obtained by asking TeX to make two passes over the input: The first pass writes a file that tells the page numbers of each marginal note; the second pass reads this file and puts the notes into the desired margin.

The second solution isn't terribly difficult, but I decided to use a third approach, which is surprisingly simple. TeX can easily be programmed to put all the notes in the left margin, or all in the right margin. Then we simply tell TeX to output only the left-hand pages, or only the right-hand pages. With two runs, we've got everything.

The text of the marginal notes was specified in Jill's journal by using a special case of an idea that appears in Appendix E of *The TeXbook*, where a similar notation is used for index entries. Namely, `^{note}` yields 'note' in the margin and also in the paragraph; `^^{note}` yields 'note' in the margin only.

Here is the macro file `jmac.tex`:

```
% format for Jill's Journal

% sample input:
% \input jmac
% \title A New Chapter That Starts a New Page
%
% \date Umbruary 29
%
% When I woke up this morning, I decided to make this
% journal into a book, using \TeX. I like to put ^{notes}
% into the margin, so that it's easy to find things later.
% My husband^^{Don} figured out a tricky way to put these
% notes into the left margin on left-hand pages, and into
% the right margin on right-hand pages.
%
% In order to do this, he claims that it's necessary to run
% \TeX\ on the file {\it twice\/}!^^{two runs needed} One
% run gives the odd-numbered pages, the other gives
% even-numbered pages. Fortunately, this doesn't take
% much longer, because printing is the slow part.
%
% This journal contains {\it ^{no math}}.
%
% \bye
```

```
% Each run begins with a little dialog:
\newif\ifleft
\def\lefthand{l }
\message{*********************** Which pages do you want (l or r)? }
\read-1 to\next          % get user's response (l or r)
\ifx\next\lefthand\lefttrue\else\leftfalse\fi
\message{OK, I'll produce only the
 \ifleft left\else right\fi-hand pages.      }

% Here are conventions for text layout
\frenchspacing             % no extra space after punctuation
\hsize=5.25in              % lines to be 5.25 inches wide
\baselineskip=14pt         % and 14 points apart
\parindent=0pt             % no paragraph indentation
\parskip=\baselineskip     % leave a blank line between paragraphs
\topskip=5\baselineskip    % leave four blank lines at top of page
\vsize=40\baselineskip     % forty lines on a page
\setbox\strutbox=\hbox{\vrule height.75\baselineskip
 depth.25\baselineskip width0pt} % this is a one-line strut

\newdimen\titleoffset    \titleoffset=1.5in     % titles move into margin
\newdimen\notespace      \notespace=.375in      % space between notes, text
\newdimen\maxnote        \maxnote=2in           % maximum width of a note

\font\titlefont=cmbx10 scaled\magstep2      % font for titles at page top
\font\datefont=cmbx10 scaled\magstephalf    % font for dates in margin
\font\notefont=cmbx10                       % font for notes in margin
\font\textrm=cmr10 scaled\magstephalf       % font for normal text
\font\textit=cmti10 scaled\magstephalf      % font for emphasized text
\font\foliofont=cmbx10 scaled\magstephalf % font for page numbers

\let\rm=\textrm \let\it=\textit \rm         % all text is either \rm or \it
\textfont2=\nullfont                        % disallow math mode

% Here ^ and ^^ are changed to \mnote, visible or invisible
\newif\ifvisible
\catcode`\^=\active
\def^{\futurelet\next\testdoublehat}
\def\testdoublehat{\ifx\next^\let\next=\silentnote
 \else\visibletrue\let\next=\mnote\fi \next}
\def\silentnote^{\visiblefalse\mnote}

\ifleft                    % do the next only if assuming left margins
 \def\title#1\par{\vfill\eject\message{#1:}
  \null\vskip-4\baselineskip
  \moveleft\titleoffset\hbox{\titlefont\uppercase{#1}}
  \vskip\baselineskip}
 \def\date#1\par{\vskip\parskip
  \moveleft\notespace
   \llap{\hbox to\maxnote{\hfil\datefont#1\unskip}}
  \nobreak\vskip-\baselineskip\vskip-\parskip}
 \def\mnote#1{\strut\vadjust{\kern-\dp\strutbox
   \vtop to\dp\strutbox{\vss \baselineskip=\dp\strutbox
    \moveleft\notespace
     \llap{\hbox to\maxnote{\hfil\notefont#1}}\null}}%
  \ifvisible#1\fi}
 \hoffset=\titleoffset
```

```
\else              % do the next only if assuming right margins
 \def\title#1\par{\vfill\eject\message{#1:}
  \null\vskip-4\baselineskip
  \moveright\titleoffset\rightline{\titlefont\uppercase{#1}}
  \vskip\baselineskip}
 \def\date#1\par{\vskip\parskip
  \moveright\notespace\rightline{%
   \rlap{\hbox to\maxnote{\datefont#1\unskip\hfil}}}
  \nobreak\vskip-\baselineskip\vskip-\parskip}
 \def\mnote#1{\strut\vadjust{\kern-\dp\strutbox
   \vtop to\dp\strutbox{\vss \baselineskip=\dp\strutbox
    \moveright\notespace\rightline{%
     \rlap{\hbox to\maxnote{\notefont#1\hfil}}}\null}}%
  \ifvisible#1\fi}
\fi                % in both cases, TeX will choose the same page breaks
% We output either left-hand or right-hand pages (only)
\output{\ifleft
 \ifodd\pageno\discard\else
  \shipout\vbox{\box255 \baselineskip=30pt \hbox{\foliofont\folio}}\fi
 \else\ifodd\pageno
  \shipout\vbox{\box255 \baselineskip=30pt \rightline{\foliofont\folio}}
  \else\discard\fi\fi
 \advancepageno}
\newbox\voidbox
\def\discard{\global\setbox255=\box\voidbox}
\outer\def\bye{\vfill\eject\deadcycles=0\end}
```

The sample file at the beginning of jmac.tex would be output as follows, on two pages (and in two passes), if the \vsize is reduced to 9\baselineskip:

A NEW CHAPTER THAT STARTS A NEW PAGE

When I woke up this morning, I decided to make this journal into a book, using TeX. I like to put notes into the margin, so that it's easy to find things later. My husband figured out a tricky way to put these notes into the left margin on left-hand pages, and into the right margin on right-hand pages.

Umbruary 29
notes
Don

1

two runs needed   In order to do this, he claims that it's necessary to run TeX on the file *twice!* One run gives the odd-numbered pages, the other gives even-numbered pages. Fortunately, this doesn't take much longer, because printing is the slow part.

no math   This journal contains *no math.*

2

The second task was rather different. Our collection of family recipes was kept on scraps of paper, and the pieces kept crumbling and/or getting lost. Jill decided to enter the recipes into her computer so that we could print them on file cards. This way we could keep everything in order, and we could also make sets for our son and daughter to use.

Jill worked out a system of codes that she found convenient for entering the data efficiently. The main interesting thing (to me) was the way it was possible to implement these codes as "active" characters in TEX. The trick was to define the macros first, before fooling around with active characters, so that the old character meanings wouldn't get mixed up with the new ones.

Here is the file rmac.tex, which should be almost "self explanatory":

```
% recipe format
% sample input:
% \input rmac
% #RELISH
% >Thanksgiving Cranberry Relish
% <Wilda Bates Carter
% $3 cups
% |chill overnight
% *
% @1 pound fresh cranberries
% 2 oranges, peeled and seeded
% rind of one orange, grated
% 1 ^1/2 c sugar
% *
% !Coarsely grind cranberries and oranges. Add rind
% and sugar. Refrigerate overnight.
% =
% #BREAD
% >Cheese Crisps
% |chill at least 2 hours, bake 20--25 minutes
% %300\0 F
% $5 dozen
% *
% @1 jar sharp cheese spread (5 ounces)
% ^1/2 c butter
% ^1/4 t salt
% dash pepper
% 1 ^1/2 c flour
% *
% !Beat together cheese and butter. Stir in remaining
% ingredients. Form into two rolls, 1~^1/4 inch in
% diameter. Wrap and chill at least 2~hours. Cut into
% ^1/4-inch slices, place on ungreased cookie sheet,
% bake 20--25 minutes at 300\0~F until slightly
% darker in color.
% =
% \bye
\hsize=4.25in
\vsize=7in
\parindent=0pt

\font\classfont=cmbx10 scaled\magstep2
\font\titlefont=cmbx10 scaled\magstep2
\font\specfont=cmsl10 scaled\magstephalf  % time, temp, qty
```

```
\font\ingredfont=cmr7 scaled\magstep2
\font\normalfont=cmr10

\newdimen\specbaseline
\specbaseline=14pt  % \baselineskip between time, temp, qty

\output{\shipout\vbox{\vbox to .75in{
  \rightline{\classfont\currentclass\hskip-.25in}\vss}
 \nointerlineskip\box255}
 \advancepageno \global\let\currentdonor=\empty}
\let\currentdonor=\empty

\def\0{$^\circ$}    % degrees

\obeylines
\def\class#1
 {\gdef\currentclass{#1}}
\def\title#1
 {{\message{#1}\titlefont#1\par}}
\def\donor#1
 {\gdef\currentdonor{#1}}
\def\time#1
 {{\baselineskip=\specbaseline \rightline{\specfont#1\/}}}
\def\temp#1\0 F
 {{\baselineskip=\specbaseline \rightline{\specfont#1\/\0 F\/}}}
\let\quantity=\time
\def\ingredients{\ingredfont\everypar{\hangindent=20pt}}
\def\method{\let^^M=\space \normalfont \everypar{}}
\def\endit{\par\vfill%
 \ifx\currentdonor\empty\else\rightline{---\currentdonor}\fi%
 \eject\obeylines}
\def\frac#1/#2{\leavevmode\raise.5ex\hbox{\the\scriptfont0 #1}%
 \kern-.1em/\kern-.15em\lower.25ex\hbox{\the\scriptfont0 #2}}
```

```
\catcode`\"=14                        " comment character
\catcode`\#=\active \let#=\class       " class of food, e.g. SOUP
\catcode`\>=\active \let>=\title       " name of recipe
\catcode`\<=\active \let<=\donor       " source of recipe
\catcode`\|=\active \let|=\time        " preparation time
\catcode`\%=\active \let%=\temp        " baking temperature
\catcode`\$=\active \let$=\quantity    " amount of output
\catcode`\@=\active \let@=\ingredients " begin list of inputs
\catcode`\!=\active \let!=\method      " begin cooking algorithm
\catcode`\*=\active \let*=\medskip     " spacer
\catcode`\^=\active \let^=\frac        " numerator of fraction
\catcode`\==\active \let=\endit        " end of recipe card
```

Notice the use of \obeylines here: Most of the data for a recipe appears on single lines, until you get to the "method" which consists of one or more paragraphs. Therefore \method converts the ends of lines to spaces. The method is followed by an '='; this finishes the card and restores \obeylines mode.

If the \vsize is reduced to 2.5 inches, the sample input produces the two cards of output shown on the next page.

Since we computerized our recipes in July, we've used the resulting cards quite often. Jill's format has worked well; it's easy to read the recipes while fixing the food, and it's easy to plan ahead because the quantities and preparation are highlighted.

Of course, the next step should be to connect the computer to our kitchen equipment, so that the cooking will be done automatically. But I think I'll work on *The Art of Computer Programming* first.

RELISH

Thanksgiving Cranberry Relish

3 cups
chill overnight

1 pound fresh cranberries
2 oranges, peeled and seeded
rind of one orange, grated
1 $1/2$ c sugar

Coarsely grind cranberries and oranges. Add rind and sugar. Refrigerate overnight.

—Wilda Bates Carter

BREAD

Cheese Crisps

chill at least 2 hours, bake 20–25 minutes
300° F
5 dozen

1 jar sharp cheese spread (5 ounces)
$1/2$ c butter
$1/4$ t salt
dash pepper
1 $1/2$ c flour

Beat together cheese and butter. Stir in remaining ingredients. Form into two rolls, 1 $1/4$ inch in diameter. Wrap and chill at least 2 hours. Cut into $1/4$-inch slices, place on ungreased cookie sheet, bake 20–25 minutes at 300° F until slightly darker in color.