

Queries

Automatic Page Balancing Macros Wanted

One of the shortcomings of T_EX (all right, all right, the only shortcoming) is its inability to handle page makeup simply. If one wishes (as one should) to find the most elegant pagebreaks, one must go into one's file and manually insert `\breaks` and `{\looseness=1\tolerance=2000...\par}`s or add lines of the form

```
\ifnum\pageno=68 \global
  \advance\vsz by-1\baselineskip\fi
\ifnum\pageno=72 \global
  \advance\vsz by1\baselineskip\fi
```

to the end of `\plainoutput`.

Altering the `\vsz` by one line in either direction (but not more) is a standard trick of good typesetting, but both pages of a spread must be the same size.

If one is unfortunate enough to have an article or chapter that has long unbreakable displays and more insertions than pages, one can end up spending altogether too much time determining pagebreaks. While, admittedly, it may be perfectly legitimate to expect an author to struggle with his file over half a dozen or so runs to make it perfect, a typesetter's time is much too valuable to indulge in such foolishness.

I had thought that it would be a simple matter to alter `\plainoutput` to `{\ifodd\pageno \oddpagelayout \else \evenpagelayout \fi}`, where `\evenpagelayout` would define a `\vbox` called `\evenpagebox` while the analogous `\oddpagelayout` was created. Once one has the two boxes, one would add the badnesses and, if the sum is greater than, say, 2000, lengthen or shorten the `\vsz` by 1 `\baselineskip` and `\unvbox` both boxes.

What a simple idea, I thought. All I was worried about at first was insertions: I had no very clear idea how they would go back on the list of recent contributions. Then I discovered that there was *no way* for T_EX to tell me how bad a `\vbox` was before it was shipped out!

Is there a Grandmaster or Wizard out there who can show me how to discover the badness of a box before it's too late?

Frederick H. Bartlett
The Bartlett Press, Inc.

Inverted Pyramidal Titles

Stephen C. Lipp's request (8(3): 326) for a title macro that (i) requires only spaces between words, (ii) capitalizes, (iii) double-spaces, (iv) centers, (v) has an inverted pyramidal shape, (vi) preferentially breaks at commas, and (vii) smoothly varies line length is hereby granted:

```
\newcount\initiallinesdone
\newcount\finallinesdone
\newcount\endhere
{\catcode'\,=\active
\gdef\invpyramid#1#2{{\catcode'\,=\active
  \def,{\char"2C\penalty-5000\}%
  \multiply\normalbaselineskip by2
  \normalbaselines
  \parfillskip=0pt\parindent=0pt
  \leftskip=0pt plus9pt minus9pt
  \rightskip=\leftskip
  \endhere=0\initiallinesdone=#2
  \loop
    \everypar={\prevgraf=\initiallinesdone}
    \global\setbox0=\vbox{
      \parshape=12 1pc25pc 2pc23pc 3pc21pc
        4pc19pc 5pc17pc 6pc15pc 7pc13pc
        8pc11pc 9pc9pc 10pc7pc 11pc5pc
        12pc3pc
      \noindent\uppercase{#1}\endgraf
      \global\finallinesdone=\prevgraf}
    \ifnum\endhere=1
      \global\advance\endhere by1
    \else
      \ifnum\finallinesdone>12
        \global\advance\initiallinesdone
          by-2\global\finallinesdone=11
        \global\advance\endhere by1
      \fi
      \ifnum\finallinesdone<12
        \global\advance\initiallinesdone
          by1
      \else
        \global\advance\endhere by2
      \fi
    \fi
  \ifnum\endhere<2
    \repeat
  \box0}}}
```

Example (I fudged the example with a `\kern -4.16667pc` since this column is $8\frac{1}{3}$ picas narrower than the measure assumed for `\invpyramid`):

```
\invpyramid{This is a long, ses\~qui\~
pe\~da\~lian, verbose title}7
```

THIS IS A LONG, SES-
QUIPEDALIAN,
VERBOSE
TITLE

The second group reflects the user's guess of how many lines the title should take up. I thought that this one would take four, so I set `\initiallinesdone` to 7. If the guess is too high, the result may be rather ugly, so use `11 - <guess>` instead of `12 - <guess>`. (This is not strictly necessary, but it speeds the process up at a small cost in human thought.)

The `\parshape` is not variable; that is, it must be determined afresh for each `\hsize` (unless the user wishes to use a variable dimension, as Knuth did in his answer to Exercise 14.18 on p. 315 of *The T_EXbook*, and further complicate the `\loop`ing).

If the user wishes to break at commas even more often, he could increase the `\penalty` to `-9999`.

If the user wishes to discourage (or encourage) hyphenation, he could give `\pretolerance`, `\exhyphenpenalty`, and `\hyphenpenalty` new values.

If the user wishes to allow more variation from a perfect pyramid, he could increase the `\leftskip` and `\rightskip`.

The subtle part of this macro is the `\loop` and its use of `\prevgraf`.

Without the `\loop`, it is obvious that only titles that are exactly 12 lines long will have appropriately narrow last lines. So, at the end of the first pass, T_EX checks `\finallinesdone`. If `\finallinesdone` is 12, we stop and set the `\vbox`. If (as is more likely) `\finallinesdone` is less than 12, we add 1 to `\initiallinesdone`, which fools `\parshape` into setting the first line shorter, and try again.

If `\finallinesdone` is greater than 12, 1 is deducted from `\initiallinesdone` and 1 is added to `\endhere`. Being interpreted, this means, "There's no way to make the last line 3 picas wide, so I'm going to do the next best thing."

It is assumed that the user's title is less than approximately 168 picas long. If it is longer, then the `\parshape` will have to be respecified for more than 12 lines. The attentive reader will note that an error (and a terrible result) would ensue if this condition is not met, for that would ensure that the very first pass would produce a `\finallinesdone` greater than 12, which would cause `\initiallinesdone` to be reduced to `-1`,

which T_EX will not allow. Thus, instead of 12 lines, you'd get about 20, the last 9 of which would all be 3 picas wide. In ten-point roman, however, 168 picas is about forty-five words, which is too long for any reasonable title anyway.

Frederick H. Bartlett
The Bartlett Press, Inc.

Logarithmic Time Scales

I should like to make a first (approximate) stab at responding to James Alexander's request in (8(2): 216) for a time scale macro.

I should admit immediately that there is a lot of grunt work to be done: I haven't (1) provided the necessary code to read dates and events from a separate file, (2) given a method for producing a linear time scale, (3) provided a method for determining the length, starting point, and finishing point of the scale, (4) met Alexander's specifications for typing the entries, or (5) addressed the problem of clustered entries.

Items (1) and (2) seem to me quite straightforward; I'm just too lazy to complete them. Item (3) is slightly more difficult: if you absolutely must have only two parameters to the `\entry` macro, you could have T_EX determine the value of what I call `\exponent` by dividing the length of a `\hbox` containing the date string by the width of a number in the current font. A new command, `\parse`, say, could then determine `\integer` and `\decimal` (`\def\parse#1#2//{\integer=#1\decimal=#2}`).

Item (4) is more difficult still: you must either have the user specify the three (dimen)s or run T_EX on the file twice: once to get the logarithms of the first and last dates and once to set the scale (if the length of the scale is not equal to `\vsize`, the user would have to specify it—unless you want T_EX to determine an optimum scale length).

Item (5) is a real bear. I don't see how T_EX could remember an arbitrary number of dates to see if they are "too close" to one another (if it can, then T_EX could also give the optimum scale length). Given a presorted list, however, it might be possible. The solution seems simple for a pair of close entries; if there are three or more close entries, though, I don't see an immediate solution (besides increasing the scale length).

What attracted me to this query was the challenge of coercing T_EX into doing logarithms. In solving this puzzle, I discovered (and if this is documented in *The T_EXbook*, I, at least, couldn't find it) that, while T_EX will add, subtract, and divide in the range ± 2147483647 , it will multiply only in the range ± 1073741823 .

Thus, my approximation algorithm was limited by $\max(\log(n+1) - \log n) \times \max(\backslashddecimal) = 1073741823$.

```

\newcount\clogi      \clogi=-32767
\newcount\clogii     \clogii=0
\newcount\clogiii    \clogiii=19167
\newcount\clogiv     \clogiv=32767
\newcount\clogv      \clogv=43316
\newcount\clogvi     \clogvi=51934
\newcount\clogvii    \clogvii=59222
\newcount\clogviii   \clogviii=65534
\newcount\clogix     \clogix=71102
\newcount\clogx      \clogx=76083
\newdimen\alog       \newcount\integer
\newdimen\decimal    \newcount\ddecimal
\newcount\exponent   \newcount\base
\newcount\basea      \newcount\tare
\tare=32768
\def\entry#1.#2E#3#4{\integer=#1
  \decimal=#2pt\exponent=#3
  \findlog
  \vskip\alog
  \vbox to0pt{\vss\line{\vrule height
    3.59267pt depth-3.35177pt width.3in
    \quad#4\hfil}\vss}
  \vskip-\alog\vskip-\baselineskip}
\def\findlog{%
  \ifnum\integer=0
    \base=0\basea=\clogi
  \else
    \base=\csname clog\romannumerals\integer
      \endcsname{\advance\integer by1
      \global\basea=\csname clog%
      \romannumerals\integer\endcsname}%
  \fi
  \advance\basea by-\base
  \advance\decimal by-.5pt
  \ddecimal=\decimal
  \multiply\ddecimal by\basea
  \multiply\basea by\tare
  \advance\ddecimal by\basea
  \divide\ddecimal by108850
  \advance\base by32767
  \multiply\base by10337
  \divide\base by17169
  \multiply\exponent by65536

```

```

\alog=\ddecimal sp
\advance\alog by\base sp
\advance\alog by\exponent sp
\multiply\alog by50}

```

This is only a rough-and-ready version; I wanted to get the basic ideas down for others to use and improve (especially since poor Prof. Alexander has been waiting for this for some months!). I suspect that a mathematician (or a more skilled T_EXpert, or both) could squeeze some more accurate logarithms out of T_EX; at a scaling factor of 50, this is only accurate to within four dots, or .9636 pts, on my laser printer.

Frederick H. Bartlett
The Bartlett Press, Inc.