* * * * * * * * * * *

HOW TO BUILD A \STRUT

Barbara N. Beeton
American Mathematical Society

Struts are things that keep objects a fixed distance apart, like the wings of a biplane. Because of the way that TEX puts boxes together vertically, struts are sometimes needed to maintain the desired distance. The concept was introduced in the definition and explanation of \| on pp. 108–109 of the TEX manual [*TEX and Metafont*]: "TEX doesn't use `\baselineskip` and `\lineskip` before and after horizontal rules." The failure of `\baselineskip` to apply the desired spacing also affects adjacent `\hbox` pars which contain more than one line, but in that case, a visible vertical rule would not be a satisfactory remedy. As it happens, an *invisible* vertical rule is just the thing, but first let us look at the problem in more detail.

In text, `\baselineskip` is typically set at 2 points greater than the text body size: 10-on-12, 9-on-11, 8-on-10. For some special work, though, it may be desirable to set material more densely, even "solid" — 10-on-10, etc. Only rarely are lines of text set any closer than that, and struts won't help with that problem in any case, so it will be ignored here. A strut for solid text should be the same height and depth as the tallest and deepest characters in the font; in METAFONT text fonts, a parenthesis ( ) or square bracket [ ] qualifies, so adjacent vertical boxes containing one of these on the last and first lines respectively will be separated by the desired distance. Consider the following example, which consists of three `\hbox` pars: the first junction lacks sufficient ascenders and descenders to force the baselines apart to the `\baselineskip` distance (this is `\tenpoint\rm \baselineskip 10pt`), but the second junction looks no different from two lines in the middle of a paragraph.

This paragraph has no
descenders in the last line.
one can scarce see an
answer to this (let's cheat).
(This example may be
contrived, but it works.)

Now, define a strut with the maximum height and depth of any character in the font, and insert it at the beginning of the first and end of the last line in each paragraph:

This paragraph has no
descenders in the last line.
one can scarce see an
answer to this (let's cheat).
(This example may be
contrived, but it works.)

Finally, reset `\baselineskip 12pt` and apply a strut that is 2 points longer:

This paragraph has no
descenders in the last line.
one can scarce see an
answer to this (let's cheat).
(This example may be
contrived, but it works.)

There are probably many ways actually to define struts, but only two will be shown here. In one approach, strut is defined within the range of each "size" definition (this is Knuth's approach):

```
\def \tenpoint{\baselineskip 12pt ...
    \def\strut{\lower 3.5pt\vbox to 12pt{}}
    ... }
```

The following is equivalent for `\tenpoint`, but more efficient (because rules take less memory space

than boxes):

```
\def \strut{\vrule
    height 8.5pt depth 3.5pt width 0pt}
```

The preceding defined a strut of the "second" kind, which at AMS we call a \strutt. We also use "throwaway" definitions, in which the size of the strut is calculated on the basis of the current text font, so that only two definitions are necessary, rather than one for each size (some applications include as many as 6 "text" sizes, including titles, footnotes, *et al.*). These probably run less efficiently, but permit more generality in the construction of \tenpoint and its friends, a memory-saver when there are frequent size changes.

```
\def \strut{\save7\hbox{()\vrule
    height 1ht7 depth 1dp7 width 0pt
    \save7\hbox{}}
\def \strutt{\save7\hbox
    {(\lower2pt\hbox{()}\vrule
    height 1ht7 depth 1dp7 width 0pt
    \save7\hbox{}}
```

Just because \hbox pars go away in TEX82 doesn't mean that the need for struts will vanish; it won't. But there will be a more efficient way to define them. See item 137 in the current list of differences between TEX80 and TEX82.

* * * * * * * * * * *

DETERMINING HASHTABLE SIZE AND OTHER QUANTITIES

Barbara N. Beeton
American Mathematical Society

The hashtable, in which names of control sequences are recorded, is limited in size. In TEX80, a common size is 1009 (= MIX); it will be larger in TEX82, but the number of primitives and names in PLAIN.TEX is also larger than the basic set. And once a name is loaded, it is never removed. So it's not too hard to run out of space for new names.

At AMS, a file called HOWMANY.MAC (a shortened version is shown below) is used to help determine how much space is left after all the header file \defs for a job have been installed. To use it, load in the header(s), then \input howmany.mac; this will rapidly fill the hashtable and blow up when there's no more room, returning with the error message the number of control sequence names that were able to be loaded before space ran out.

```
\chcode`176=11 % ~ - make it a letter

\def \~~~{0 }

\def \~AA{1 }
\def \~AB{2 }
\def \~AC{3 }
   . . .
\def \~AS{19 }
\def \~AT{20 }
\def \~BA{21 }
\def \~BB{22 }
   . . .
\def \~OS{299 }
\def \~OT{300 }
```

The control sequence names used in such a file must be unique. In HOWMANY.MAC, only 20 letters of the alphabet were used, so that the file is easily extensible with the help of a few commands to an editor. This also made checking the count easier. If the count is wrong, or a name in this file duplicates that of an existing control sequence, results will be unreliable.

The following test ran on a SAIL version of TEX80 on a DEC 2060.

```
@tex header.fil
[input from: notices]
AMS TeX varsize of 11500 created
  Wednesday, August 18, 1982 16:07:51
this run of TeX begun:
  Tuesday, March 1, 1983 23:00:32

*
(PS:<BNB>HEADER.FIL.1)
*\input howmany.mac

(PS:<AMSTEX>HOWMANY.MAC.3 1
! TEX capacity exceeded, sorry [hashsize=1009].
p.1,l.67 \def \~DC
                {63 }
No output file.
```

It is useful to run a new set of header files through this test before putting them into production, and some other useful checks can be performed at the same time. Use a 'slow' version of TEX (one compiled in such a way that statistics are saved for such things as memory size), and insert \ddt (for TEX80) before \input howmany.mac. This will report the location on the page and the current nesting level, all of which should ordinarily be 0 if there aren't any braces missing in the header. It will also tell you how much memory the header requires—if total available memory is 22,000 words, and the header alone occupies 19,000, some pruning is in order.

* * * * * * * * * * *