Documentation for multido.tex:
A loop macro for Generic \TeX

Timothy Van Zandt*

Version 1.41
May 18, 2004

multido.tex/multido.sty contains the \multido macro, which is a loop facility for Generic TeX. This macro happens to be useful for drawing pictures, and was originally developed for the PSTricks package,¹ but you can use it for other purposes as well.

A special feature is support of fixed-point addition. For example, PSTricks uses the \multido to put numbers on axes, much like in the following \LaTeX example:

\begin{verbatim}
\setlength{\unitlength}{1cm}
\small
\begin{picture}(8,1)(0,-.5)
  \put(0,0){\vector(1,0){8}}
  \multido{i=0+1,n=0+0.25}{8}{%  
    \put(i,-.1){\line(0,1){.2}}
    \put(i,-.2){\makebox(0,0)[t]{n}}}
\end{picture}
\end{verbatim}

The general syntax for \multido is:

\multido\{variables\}\{repetitions\}\{stuff\}

\textit{stuff} is whatever you want repeated; it can be any balanced \TeX input. \textit{repetitions} is the number times \textit{stuff} is repeated.

The first argument is the interesting one. \textit{variables} is a comma-separated list of variable declarations.² Each variable declaration is of the form:

\begin{itemize}
  \item[\texttt{\{\textbf{variables}\}}]\texttt{\{\textbf{repetitions}\}}\texttt{\{\textbf{stuff}\}}
\end{itemize}

¹PSTricks is an extensive collection of PostScript-based macros for Generic \TeX. It is available from CTAN://graphics/pstricks/.

²Don’t use commas to mark the decimal point within the \textit{variables} argument, as they will be confused for delimiters.

*Author’s address: Department of Economics, Princeton University, Princeton, NJ 08544-1021, USA. Internet: tvz@Princeton.EDU. This document was edited by Rolf Niepraschk Rolf.Niepraschk@ptb.de and Herbert Voss voss@perce.de to get it run with \LaTeX2e (2006-01-01)
variable = initial value + increment

variable is a command sequence that can be used in stuff. It is initially set to initial value, and is then incremented by increment with each repetition. The first letter of the variable name determines the variable type. There are four variable types:

**Dimension (d or D)** The initial value and the increment should be dimensions (lengths, in \TeX\ parlance). The substitution text is a dimension, with \textbf{sp} units. E.g., $\dx=4\text{cm}+5\text{pt}.$

**Number (n or N)** The initial value and increment should be integers or numbers with the same number of digits to the right of the decimal. The one exception is that it is always OK for the initial value to be an integer. There can be at most 8 digits on each side of the decimal. The substitution text is a number, with fixed-point addition. E.g., $\n=3+7.05, \Nx=5.30-1.25.$

**Integer (i or I)** The initial value and increment should be integers. This gives the same result as using a number variable, but it is faster. E.g., $\I=2+-1.$

**Real (r or R)** The initial value and increment should be integers or numbers with at most 4 digits on each side of the decimal. The substitution text is a number, but with floating point addition and occasional small errors. This gives a less satisfactory result than using a number variable, but it is faster. E.g., $\ry=4.2+1.05.$

Here are some examples that illustrate how the substitution text is determined:

```
\multido{}{10}{\TeX\ }
\multido{\d=2pt+3pt}{5}{\d, }
\multido{\n=2+3}{10}{\n, }
\multido{\i=2+-3}{10}{\i, }
\multido{\r=2+3.05}{6}{\r, }
```

For PSTricks users, the unit is optional.
\multido{\n=2.00+-3.05}{8}{\n, }
2.00, -1.05, -4.10, -7.15, -10.20, -13.25, -16.30, -19.35,

Here are some details about the choice of names:

• Your computer won’t explode if you use names that conflict with \TeX\ internal commands, but you might want to check name conflicts if you get inexplicable errors. The command \MultidoCheckNames can be useful in this case. It causes \multido to report an error whenever you use a variable name that is already defined. But see the next item.

• The whole \multido loop is grouped. This means, e.g., that although \i is a Plain \TeX\ command sequence (giving a dotless “i”), you can use the variable \i if you do not use any dotless i’s in stuff (and if you do not use \MultidoCheckNames).

Here are a few more details:

• \Multido commands can be nested.

• Spaces after a \multido command are ignored. This makes \multido more hospitable for pictures.

• Spaces between the various parts of the variables argument are ignored.

And finally here a few special features, some of which are of interest mainly macro writers and other \TeX\nicians:

• The material that is repeated is not grouped, so that you can insert your own recursive routines.

• There is a variant, \mmultido, which works just like \multido except that the variables are all incremented once before starting.

• If you use \Multido or \MMultido instead of \multido or \mmultido, resp., then the whole loop is not grouped. This can be useful, e.g., for making entries in an alignment environment. However, these cannot be nested within any \multido macro.

• If the number of repetitions is a negative number, the variables are incremented backwards.

• The count register \multidocount keeps track of the number of current iteration.

• The command \multidostop causes the \multido loop to quit at the end of the current iteration.
• Fixed point addition is performed by \texttt{\fpAdd} and \texttt{\fpSub}:

\begin{verbatim}
\fpAdd{num1}{num2}{command}
\fpSub{num1}{num2}{command}
\end{verbatim}

\textit{num2} is added to or subtracted from \textit{num1}, and the answers is stored in the command sequence given as the third argument. The rules about decimals and so on that apply to number variables apply here as well. E.g., after

\begin{verbatim}
\fpSub{1.75}{-0.15}{\answer}
\end{verbatim}

the definition of \texttt{\answer} is 1.90.

Changes:

V1.1 Fixed bug in \texttt{\FPadd} that gave wrong answer for, e.g., 3.4 + -0.2.

V1.2 Made unit optional for dimension variables when using PStricks.

V1.3 Now 0 repetitions really gets 0 repetitions. \texttt{\def\multido@} changed to \texttt{\long\def\multido@}.

V1.4 Small change to make it compatible with PStricks v0.93 and later.

V1.41 Fix bug when using the fp-package (same macronames) and small changes to this documentation (RN/HV)