multirow.sty — Span multiple rows of a table[*]

Jerry Leichter Piet van Oostrum

Abstract

The multirow macro package can make an entry that will span multiple rows of a table.

Usage

```
\multirow{nrows}[bigstruts]{width}[fixup]{text}
```

`nrows`	the number of rows to span. It's up to you to leave the other rows empty, or the stuff created by `\multirow` will over-write it. With a positive value of nrows the spanned colomns are this row and (nrows-1) rows below it. With a negative value of nrows they are this row and (1-nrows) above it.
`bigstruts`	mainly used if you've used `bigstrut.sty`. In that case it is the total number of uses of `\bigstrut` within the rows being spanned. Count 2 uses for each `\bigstrut`, 1 for each `\bigstrut[x]` where x is either t or b. The default is 0.
`width`	the width to which the text is to be set, or * to indicate that the text argument's natural width is to be used.
`text`	the actual text. If the width was set explicitly, the text will be set in a parbox of that width; you can use `\\` to force linebreaks where you like. If the width was given as * the text will be set in LR mode. If you want a multiline entry in this case you should use a tabular or array in the text parameter. The text is centered vertically within the range spanned by nrows.
`fixup`	a length used for fine tuning: The text will be raised (or lowered, if fixup is negative) by that length above (below) wherever it would otherwise have gone.

[*]V1.5 version (16-July-1999)

For example (using both multirow and bigstrut)

```
\newcommand{\minitab}[2][l]{\begin{tabular}{#1}#2\end{tabular}}
\begin{tabular}{|c|c|}
\hline
\multirow{4}{1in}{Common g text} & Column g2a\\
      & Column g2b \\
      & Column g2c \\
      & Column g2d \\
\hline
\multirow{3}[6]*{Common g text} & Column g2a\bigstrut\\\cline{2-2}
      & Column g2b \bigstrut\\\cline{2-2}
      & Column g2c \bigstrut\\
\hline
\multirow{4}[8]{1in}{Common g text} & Column g2a\bigstrut\\\cline{2-2}
      & Column g2b \bigstrut\\\cline{2-2}
      & Column g2c \bigstrut\\\cline{2-2}
      & Column g2d \bigstrut\\
\hline
\multirow{4}*{\minitab[c]{Common \\ g text}} & Column g2a\\
      & Column g2b \\
      & Column g2c \\
      & Column g2d \\
\hline
\end{tabular}
```

will give the follow table:

Common g text	Column g2a
	Column g2b
	Column g2c
	Column g2d
Common g text	Column g2a
	Column g2b
	Column g2c
Common g text	Column g2a
	Column g2b
	Column g2c
	Column g2d
Common g text	Column g2a
	Column g2b
	Column g2c
	Column g2d

If any of the spanned rows are unusually large, or if you're using bigstrut.sty and `\bigstrut`'s are used assymetrically about the centerline of the spanned rows, the vertical centering may not come out right. Use the fixup argument in this case.

Just before "text" is expanded, the `\multirowsetup` macro is expanded to set up any special environment. Initially, `\multirowsetup` contains just `\raggedright`. It can be redefined with `\renewcommand`.

Bugs: It's just about impossible to deal correctly with descenders. The text will be set up centered, but it may then have a baseline that doesn't match the baseline of the stuff beside it, in particular if the stuff beside it has descenders and "text" does not. This may result in a small missalignment. About all that can be done is to do a final touchup on "text", using the fixup optional argument. (Hint: If you use a measure like .1ex, there's a reasonable chance that the fixup will still be correct if you change the point size.)

`\multirow` is mainly designed for use with table, as opposed to array, environments. It will not work well in an array environment since the lines have an extra `\jot` of space between them which it won't account for. Fixing this is difficult in general, and doesn't seem worth it. The bigstruts argument can be used to provide a semi-automatic fix: First set `\bigstrutjot to .5\jot`. Then simply repeat nrows as the bigstruts argument. This will be close, but probably not exact; you can use the fixup argument to refine the result. (If you do this repeatedly, you'll probably want to wrap these steps up in a simple macro. Note that the modified `\bigstrutjot` value will not give reasonable results if you have bigstruts and use this argument for its intended purpose elsewhere. In that case, you might want to set it locally.)

If you use `\multirow` with the colortbl package you have to take precautions if you want to color the column that has the `\multirow` in it. colortbl works by coloring each cell separately. So if you use `\multirow` with a positive nrows value, colortbl will first color the top cell, then `\multirow` will typeset nrows cells starting with this cell, and later colortbl will color the other cells, effectively hiding the text in that area. This can be solved by putting the `\multirow` is the last row with a negative nrows value.
Example:

```
\begin{tabular}{l>{\columncolor{yellow}}l}
  aaaa & \\
  cccc & \\
  dddd & \multirow{-3}*{bbbb}\\
\end{tabular}
```

will produce:

aaaa	
cccc	bbbb
dddd	