The vwcol package

Will Robertson

2015/02/10 v0.2

1 *Introduction*

This package provides an environment that allows paragraph text to be typeset into multiple columns of uneven width, with text that flows from one column to the next. The columns can not span over multiple pages.

Due to difficulties with the processing of such a thing, little else *besides* text is allowed within (feel free to experiment, but you're on your own). Here's an example:[1]

```
\begin{vwcol}[widths={0.3,0.2,0.5}]
  \lipsum[1]
\end{vwcol}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 *Options*

As shown above, at heart this package is quite simple. This section discusses the options that can be passed to the `vwcol` environment. The options are:

`widths`	The number and size of the columns.
`sep`	The width of the space between the columns.
`sidesep`	Whether to add space on the outside of the columns (equiv. to the following two options together).
`presep`	Whether to add space before the columns.
`postsep`	Whether to add space after the columns.
`rule`	The width of the rule.

---

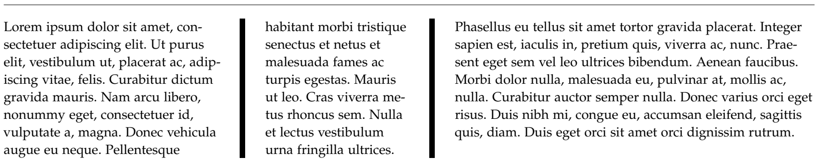
[1] Requires the lipsum package to print the sample text.

`siderule`	Whether to draw a rule on the outside of the columns (equiv. to the following two options together).
`prerule`	Whether to draw a rule before the columns.
`postrule`	Whether to draw a rule after the columns.
`rulecolor`	The colour of the rule.[2]
`justify`	Paragraph justification within the columns.
`indent`	Indentation size within the columns (if relevant).

Paragraph options `justify` and `indent` are covered in section §3 on page 5, and some advanced options are discussed in section §4 on page 6.

`\vwcolsetup` This macro may be used to set the default values for the options (described subsequently) of the `vwcol` environment.

```
\vwcolsetup{widths={0.3,0.2,0.5},rule=2pt}
\begin{vwcol}
  \lipsum[1]
\end{vwcol}
```



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`widths` This option must always be present (either as a default value previously set in `\vwcolsetup` or specified in the environment directly) and consists of any number of comma-separated lengths or ratios. Lengths set the column width to an explicit size, whereas a ratio (as above) sets the column width to a fraction of the available linewidth (leaving some space for some separation between the columns).

As shown in the example in section §1, when the width ratios sum to 100% then the multi-columns will span the entire line width regardless of the chosen separation between the columns. A set of widths may be any combination of ratios and lengths, but the total width should not exceed the linewidth available (a warning will be given if so).

`sep` The separation between the columns can be chosen as either a length, a ratio of the linewidth, or the keyword `fill`. The default is `sep=0.05` (*i.e.*, 5% of the linewidth).

---

[2] Added in v0.2.

```
\begin{vwcol}[widths={0.35,0.25,0.4},sep=5pt]
  \lipsum[1]
\end{vwcol}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

The keyword `fill` adds stretchable space between the columns so the multicolumns fill the entire linewidth (without altering the widths of the columns themselves):

```
\begin{vwcol}[widths={2cm,2cm,0.4},sep=fill]
  \lipsum[1]
\end{vwcol}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

If ratio column widths are used with a variable separation gap, then the separation gap is considered zero for the total width calculation. In this example, because the ratios for the column widths sum to 100% there is no room left over for a separation gap:

```
\begin{vwcol}[widths={0.3,0.2,0.5},sep=fill]
  \lipsum[1]
\end{vwcol}
```

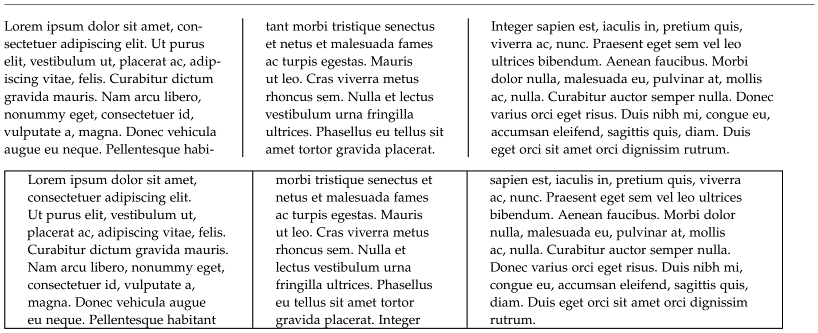
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`presep`
`postsep`
`sidesep`

These options control whether an extra separation is added before and/or after the multicolumns. `presep` (or `presep=true`) adds space before the columns (and `presep=false` suppresses it); `postsep` adds space after the columns; `sidesep` is a shorthand for activating both at once.

```
\begin{vwcol}[widths={0.3,0.25,0.4}]
  \lipsum[1]
\end{vwcol}

\setlength\fboxsep{0pt}
\fbox{\begin{vwcol}[widths={0.3,0.25,0.4},sidesep]
  \lipsum[1]
\end{vwcol}}
```

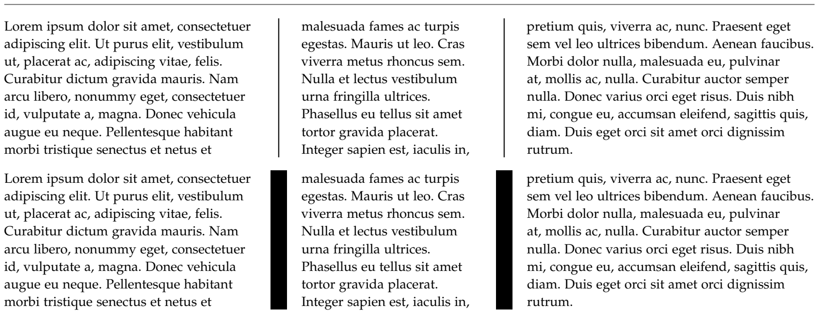


Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`rule` The width of the rule is configurable (again, either a length or a ratio of the line width) and does not affect the separation gap. Use `rule=none` or `rule=0pt` to suppress drawing the rule. The default is `rule=0.4pt`.

```
\begin{vwcol}[widths={0.35,0.25,0.4}]
  \lipsum[1]
\end{vwcol}
\begin{vwcol}[widths={0.35,0.25,0.4},rule=0.02]
  \lipsum[1]
\end{vwcol}
```

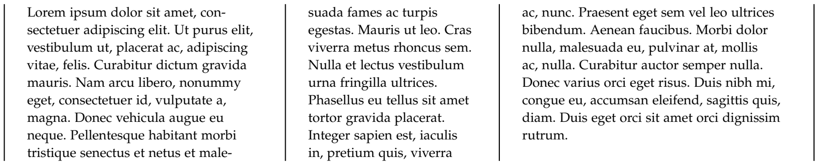


Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`prerule`
`postrule`
`siderule` These options control whether extra vertical rules are added before and/or after the columns. `prerule` places a rule before the columns; `postrule` after them. (Again, `prerule=false` (*etc.*) turns this feature off.) And `siderule` is a shorthand to activate both. Using these options implicitly activates the relevant `presep` and/or `postsep` options, because you can't have the rule without the gap.

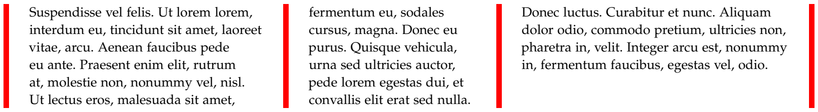
```
\begin{vwcol}[widths={0.35,0.25,0.4},siderule]
  \lipsum[1]
\end{vwcol}
```



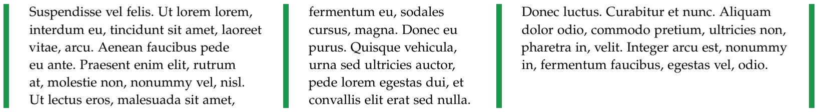
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`rulecolor` The colour of each rule. Either used a pre-defined name or define your own using color or xcolor. The color package is loaded by this package.

```
\vwcolsetup{widths={0.35,0.25,0.4},siderule,rule=2pt}
\begin{vwcol}[rulecolor=red]
  \lipsum[6]
\end{vwcol}
\definecolor{myrulecol}{rgb}{0.1,0.6,0.3}
\begin{vwcol}[rulecolor=myrulecol]
  \lipsum[6]
\end{vwcol}
```



Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.



Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

3 *Paragraph settings*

`justify` The justification to use; one of `ragged` (*default*), `flush`, `raggedleft`, or `center`. These settings are made using the ragged2e package, with the result that hyphenation is enabled even in the ragged settings (this is a good thing!); due to a limitation of TEX's `\parshape`, LATEX's ordinary `\raggedright` setting cannot be used.

```
\begin{vwcol}[widths={0.35,0.25,0.4}]
  \lipsum[66]
\end{vwcol}
\begin{vwcol}[widths={0.35,0.25,0.4},justify=flush]
  \lipsum[66]
\end{vwcol}
\begin{vwcol}[widths={0.35,0.25,0.4},justify=raggedleft]
  \lipsum[66]
\end{vwcol}
\begin{vwcol}[widths={0.35,0.25,0.4},justify=center]
  \lipsum[66]
\end{vwcol}
```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at	purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum.	Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.
Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus	et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mau-	ris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.
Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasel-	lus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante	placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.
Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus	at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo	semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

`indent` This option is used to set the paragraph indent for ragged right and justified paragraph shapes (by default `[indent=1.5em]`).

```
\begin{vwcol}[widths={0.35,0.25,0.4},indent=5em]
  \lipsum[66]\lipsum[66]
\end{vwcol}
```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis.	Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum. Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget,	interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Note that the first column always begins with a `\noindent`. Let me know if you don't like this idea.

4 *Advanced (read: not very useful) options*

`quiet` The vwcol package passes certain information about what it's doing via errors in compilation, warnings in the console output, and info in the `.log` file. Loading vwcol with the `[quiet]` option 'demotes' the priority of these diagnostics: errors become warnings, warnings become info in the `.log` file, and info is suppressed entirely.

`lines` With the default `[lines=auto]`, the `vwcol` environment tries to estimate how much space is required but it will sometimes get it wrong. Pass an integer to the `lines` option to specify exactly how many lines to use (which will also save processing time), but if the value chosen is too small then text will be lost (and an error given):

```
\begin{vwcol}[widths={0.35,0.25,0.4},lines=4]
  \lipsum[1]
\end{vwcol}
\begin{vwcol}[widths={0.35,0.25,0.4},lines=11]
  \lipsum[1]
\end{vwcol}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

The rationale behind producing an error is that you really want to be alerted if text in your input is not making it into the output document (*cf.* with trying to insert a character that doesn't exist in the current font).

`maxrecursion` When the estimate number of lines is calculated, the value is sometimes too small. `vwcol` will increment the number of lines one-by-one at most `maxrecursion` times until the text completely fits into the columns. If it hits `maxrecursion`, then an error is reported explaining what's going on.

The default is `5`, but I'd be surprised if you ever need to adjust this parameter.

5 *Usage notes*

If you want the `widths` ratios to use a different width to denote 100% (instead of `\linewidth`), put the whole thing in a `minipage` or `\parbox`:

```
\begin{minipage}{0.8\linewidth}
  \begin{vwcol}[widths={0.3,0.7},indent=1.8em]
    \lipsum[66]\lipsum[66]
  \end{vwcol}
\end{minipage}
```

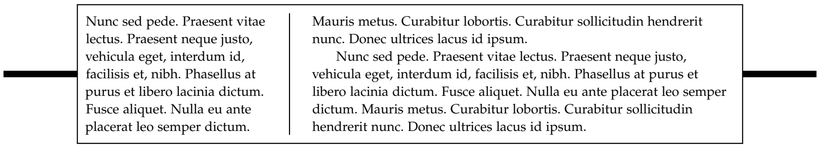
Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

(I might add an option to `vwcol` to allow this directly; *E.g.*, `[totalwidth=0.8\linewidth]`. Let me know if you like the idea.)

The `vwcol` environment ends the previous paragraph at `\begin{vwcol}` and terminates the paragraph it is contained within at `\end{vwcol}`. This means you can't place two `vwcol` environments next to each other, for example (or next to anything else, for that matter). If you want to be able to do this, again, put them in `minipage`s or `\parbox`es:

```
\rule{0.1\linewidth-\fboxsep}{1ex}%
%
\fbox{\parbox{0.8\linewidth}{%
  \begin{vwcol}[widths={0.3,0.7},indent=1.8em]
    \lipsum[66]\lipsum[66]
  \end{vwcol}}}%
%
\rule{0.1\linewidth-\fboxsep}{1ex}%
```



Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

(I might add an option to `vwcol` to allow this directly; *E.g.*, `[par=false]` or `[block=par]` vs. `[block=inline]`. Let me know if you like the idea.)

Note in both of these cases that the `\parindent` length had to be redefined after `{minipage}` or `\parbox` defined it to zero inside themselves.

6 *Acknowledgements*

Many thanks to Flavio Costa for testing an early version of this package and especially for proof-reading this documentation. In large part due to him this manual makes much more sense :)

File I

vwcol implementation

This is the package.

```
\ProvidesPackage{vwcol}
  [2015/02/10 v0.2 Variable-width multicolumn text]
```

7 Preamble

7.1 Packages

```
\RequirePackage{calc}
\RequirePackage{color}
\RequirePackage{environ}[2008/06/18]
\RequirePackage{keyval}
\RequirePackage{ragged2e}
```

7.2 Things we need

```
\newlength\vwcol@sep
\newlength\vwcol@rule
\newlength\vwcol@totalwidth
\newlength\vwcol@averagewidth
\newlength\vwcol@parindent

\newcount\vwcol@last
\newcount\vwcol@Ncols
\newcount\vwcol@Nlines
\newcount\vwcol@maxrecursion

\newbox\vwcol@box
\newbox\vwcol@plainbox
\newbox\vwcol@outputbox

\newif\if@vwcol@boxready
\newif\if@vwcol@prerule
\newif\if@vwcol@postrule
\newif\if@vwcol@presep
\newif\if@vwcol@postsep
```

7.3 Conveniences

Start error and warning text on a new line coz I think it looks better that way:

```
\newcommand\vwcol@PackageError[2]{%
  \PackageError{vwcol}{^^J\space\space#1}{#2}}
```

```
\newcommand\vwcol@PackageWarning[1]{%
  \PackageWarning{vwcol}{%
    ^^J\space\space#1^^JThis warning occurred}}
```

```
\newcommand\vwcol@PackageInfo[1]{%
  \PackageWarning{vwcol}{%
    ^^J\space\space#1^^JThis warning occurred}}
```

7.4 Package option

```
\DeclareOption{quiet}{%
  \renewcommand\vwcol@PackageError[2]{%
    \vwcol@PackageWarning{#1.}}%
  \let\vwcol@PackageInfo\@gobble}
\ProcessOptions
```

8 Auxiliary macros

`\vwcol@test@length`

`{#1}`: Rational number or length (*i.e.*, with unit)
`{#2}`: Multiplier for the rational (*e.g.*, `\linewidth`)

This macro returns `\@tempswa` true if the input is a rational number (*e.g.*, `0.1`, `1`, *etc.*) or false if it is a length (*e.g.*, `2pt`, `3cm`).[3] `\@tempdima` contains the length corresponding to the rational number multiplier of #2 or the length input, respectively.

```
\vwcol@test@length{1}{\linewidth}
\if@tempswa Rational\else Length\fi\\
\vwcol@test@length{1cm}{\linewidth}
\if@tempswa Rational\else Length\fi
```

Rational
Length

```
\def\vwcol@test@length#1#2{%
  \afterassignment\vwcol@test@@
  \@tempdima=#1#2\@nil}
```

The afterassignment macro:

```
\def\vwcol@test@@#1\@nil{%
  \ifx\@nil#1\@nil
    \@tempswatrue
  \else
    \@tempswafalse
  \fi}
```

---

[3] Based on a similar macro by David Kastrup: `http://groups.google.com/group/comp.text.tex/msg/9bd5349ea2416c95`

Actually, I don't use `\if@tempswa` in this package (I use `\@tempdima` directly), but I've left the conditional in there in case someone else finds it useful.

9 *Environment options*

`\vwcolsetup` To set the defaults:

```
\def\vwcolsetup{\setkeys{vwcol}}
```

widths The number and size of each column.

```
\define@key{vwcol}{widths}{\def\vwcol@widths{#1}}
```

No defaults.

maxrecursion Number of iterations used to estimate the number of lines. I doubt if it will ever need to be changed from the default.

```
\define@key{vwcol}{maxrecursion}{\vwcol@maxrecursion=#1}
```

Default:

```
\vwcolsetup{maxrecursion=5}
```

rule The width of the intercolumn rule as a length or as a ratio of the total line width or as the keyword `none`.

```
\define@key{vwcol}{rule}{%
  \def\@tempa{#1}%
  \def\@tempb{none}%
  \ifx\@tempa\@tempb
    \vwcol@rule=0pt
  \else
    \vwcol@test@length{#1}{\linewidth}%
    \vwcol@rule=\@tempdima
  \fi}
```

Default:

```
\vwcolsetup{rule=0.4pt}
```

lines The number of lines of text in each column or the keyword `auto`.

```
\define@key{vwcol}{lines}{%
  \def\@tempa{#1}%
  \def\@tempb{auto}%
  \ifx\@tempa\@tempb
    \vwcol@Nlines=0
  \else
    \vwcol@Nlines=#1
  \fi}
```

Default:

```
\vwcolsetup{lines=auto}
```

sep The distance between each column (including space taken up by the rule, if any) as a length or as a ratio or as the keyword `fill`.

```
\define@key{vwcol}{sep}{%
  \def\@tempa{#1}%
  \def\@tempb{fill}%
  \ifx\@tempa\@tempb
    \vwcol@sep=1sp
  \else
    \vwcol@test@length{#1}{\linewidth}%
    \vwcol@sep=\@tempdima
  \fi}
```

Default:

```
\vwcolsetup{sep=0.05}
```

presep Whether to include a gap before the first column.

```
\define@key{vwcol}{presep}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@preseptrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@presepfalse
    \else
      \vwcol@PackageWarning{%
        ’#1’ not a valid option for option ’presep’;
        ’true’ or ’false’ only.}%
    \fi
  \fi}
```

Default:

```
\vwcolsetup{presep=false}
```

postsep Whether to include a gap after the last column.

```
\define@key{vwcol}{postsep}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@postseptrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@postsepfalse
    \else
      \vwcol@PackageWarning{%
        '#1' not a valid option for option 'postsep';
        'true' or 'false' only.}%
    \fi
  \fi}
```

Default:

```
\vwcolsetup{postsep=false}
```

sidesep Shorthand for setting both `presep` and `postsep` at once.

```
\define@key{vwcol}{sidesep}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@preseptrue
    \@vwcol@postseptrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@presepfalse
      \@vwcol@postsepfalse
    \else
      \vwcol@PackageWarning{%
        '#1' not a valid option for option 'sidesep';
        'true' or 'false' only.}%
    \fi
  \fi}
```

prerule Whether to place a rule before the first column (implies `presep`).

```
\define@key{vwcol}{prerule}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@preseptrue
    \@vwcol@preruletrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@prerulefalse
    \else
```

```
      \vwcol@PackageWarning{%
        ’#1’ not a valid option for option ’prerule’;
        ’true’ or ’false’ only.}%
    \fi
  \fi}
```

Default:

```
\vwcolsetup{prerule=false}
```

postrule Whether to place a rule after the last column (implies `postsep`).

```
\define@key{vwcol}{postrule}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@postseptrue
    \@vwcol@postruletrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@postrulefalse
    \else
      \vwcol@PackageWarning{%
        ’#1’ not a valid option for option ’postrule’;
        ’true’ or ’false’ only.}%
    \fi
  \fi}
```

Default:

```
\vwcolsetup{postrule=false}
```

siderule Shorthand for setting `prerule` and `postrule` simultaneously.

```
\define@key{vwcol}{siderule}[true]{%
  \def\@tempa{#1}%
  \def\@tempb{true}%
  \ifx\@tempa\@tempb
    \@vwcol@preseptrue
    \@vwcol@postseptrue
    \@vwcol@preruletrue
    \@vwcol@postruletrue
  \else
    \def\@tempb{false}%
    \ifx\@tempa\@tempb
      \@vwcol@prerulefalse
      \@vwcol@postrulefalse
    \else
```

```
      \vwcol@PackageWarning{%
        '#1' not a valid option for option 'siderule';
        'true' or 'false' only.}%
    \fi
  \fi}
```

justify The justification to use; one of `flush`/`ragged`/`raggedleft`/`center`.

```
\define@key{vwcol}{justify}{%
  \def\@tempa{#1}%
  \def\@tempb{ragged}%
  \ifx\@tempa\@tempb
    \let\vwcol@justify\RaggedRight
  \else
    \def\@tempb{flush}%
    \ifx\@tempa\@tempb
      \let\vwcol@justify\justifying
    \else
      \def\@tempb{raggedleft}%
      \ifx\@tempa\@tempb
        \let\vwcol@justify\RaggedLeft
      \else
        \def\@tempb{center}%
        \ifx\@tempa\@tempb
          \let\vwcol@justify\Centering
        \else
          \vwcol@PackageWarning{%
            '#1' not a valid option for option 'justify';
            one of 'flush'/'ragged'/'raggedleft'/'center' only.}%
        \fi
      \fi
    \fi
  \fi}
```

Default:

```
\vwcolsetup{justify=ragged}
```

indent The paragraph indent to use with `flush` or `ragged` justification.

```
\define@key{vwcol}{indent}{\setlength\vwcol@parindent{#1}}
```

Default:

```
\vwcolsetup{indent=1.5em}
```

rulecolor The colour of each rule.

```
\define@key{vwcol}{rulecolor}{\def\vwcol@rulecol{#1}}
\vwcolsetup{rulecolor=black}
```

No defaults.

10 vwcol environment definition

vwcol Always start a new par.

```
\NewEnviron{vwcol}[1][]{%
  \par\noindent
```

Initialisation:

```
  \@vwcol@boxreadyfalse
  \vwcolsetup{#1}%
```

Ensure the space at the top of each column is uniform:

```
  \splittopskip=\ht\strutbox
```

Setup widths (this counts the columns and calculates the average and total widths of the columns):

```
  \expandafter\vwcol@process@widths\expandafter{\vwcol@widths}%
```

Set up the paragraph parameters:

```
  \vwcol@para@setup
```

From the width of the columns, the total width of the environment can be calculated. First, if `sep=fill` then the whole linewidth will be used:

```
  \ifdim\vwcol@sep=1sp
    \vwcol@totalwidth=\linewidth
```

Otherwise calculate the total from the number of separation gaps:
(`\vwcol@totalwidth` is currently the total of the columns widths, which was calculated above in `\vwcol@process@widths`)

```
  \else
    \vwcol@totalwidth=\numexpr
      \vwcol@totalwidth+(\vwcol@Ncols-1)*\vwcol@sep
      \relax sp
```

Add on extra space due to the optional pre- and post-separation gaps and rules. Note that while rules between columns do not contribute to the total width of the columns (they subtract from the empty space in the gaps between the columns, which explains why the correction is needed in the `presep`/`postsep` length processing), pre- or post-rules *do*.

```
    \if@vwcol@presep
      \advance\vwcol@totalwidth\dimexpr(\vwcol@sep-\vwcol@rule)/2\relax
    \fi
    \if@vwcol@postsep
      \advance\vwcol@totalwidth\dimexpr(\vwcol@sep-\vwcol@rule)/2\relax
    \fi
    \if@vwcol@prerule \advance\vwcol@totalwidth \vwcol@rule\fi
    \if@vwcol@postrule\advance\vwcol@totalwidth \vwcol@rule\fi
  \fi
```

Finally, warn the author if their columns are going to be too large:

```
  \ifdim\vwcol@totalwidth > \linewidth
    \vwcol@PackageWarning{%
      Total width of columns plus their separations
      is greater than the linewidth^^J\space\space
      (by \the\vwcol@totalwidth\space - \the\linewidth\space =
       \the\dimexpr \vwcol@totalwidth-\linewidth\relax)}%
  \fi
  \ifnum\vwcol@Nlines=0%
```

If the `lines` are not explicitly selected then they must be estimated. Typeset the text into a single box of the average column width (while ignoring overfull/underfull boxes):

```
    \@tempcnta=\hbadness
    \hbadness=\maxdimen
    \setbox\vwcol@plainbox\hbox{%
      \parbox{\vwcol@averagewidth}{\vwcol@justify\BODY}}%
    \hbadness=\@tempcnta
```

Now the estimate of the number of lines per column, L, can be calculated. Start by assuming that the 'area' of the material in the single block will be about the same when split into columns of un-equal width, w_i. (By 'area' we *actually* mean the number of lines in a block multiplied by the number of lines N.) If T is the total number of lines of the single block typeset above (which is calculated by dividing the height of the block by the baselineskip), this gives

$$T \times w_a \approx L \times w_1 + L \times w_2 + \cdots = L \times \sum_{i=1}^{N} w_i.$$

The width of the single block is defined above to be the average of the column widths:

$$w_a = \text{ave}(w_i) = \sum_{i=1}^{N} w_i / N$$

These two expressions are easily combined to give

$$L = \frac{T \times \text{ave}(w_i)}{\sum_{i=1}^{N} w_i} = \frac{T}{N}.$$

In words, the number of lines per column is simply to simply the number of lines in the single block divided by the number of columns.

```
    \vwcol@Nlines=\numexpr
        (\ht\vwcol@plainbox+\dp\vwcol@plainbox)/
        (\baselineskip*\vwcol@Ncols)
      \relax
```

However, differences may arise due to rounding (due to TEX's integer arithmetic, the floor of the resultant value is always calculated[4]) and hyphenation/justification variations between the two cases.

Due to these differences, we start with the calculated number of lines and increment in a loop if necessary to ensure all of the material does actually fit. It's unlikely that the number of lines estimated will be *greater* than the number of lines required due to the effect of the 'flooring' of the calculations.

```
    \@tempcnta=1%
    \loop\unless\if@vwcol@boxready
      \savebox\vwcol@outputbox{%
        \hbox to \vwcol@totalwidth{\vwcol@{\BODY}}}%
      \unless\if@vwcol@boxready
        \advance\@tempcnta 1%
        \advance\vwcol@Nlines 1%
```

Here we could keep looping for as long as necessary, but in case of weird input we put a hard limit on the number of iterations. Stop after the line number has been incremented five times (by default) because surely the calculation couldn't have been that far wrong.

```
        \ifnum\@tempcnta>\vwcol@maxrecursion
          \@vwcol@boxreadytrue
          \vwcol@PackageError{%
            The estimated number of lines is greater than
            \the\vwcol@maxrecursion\space lines too small,%
              ^^J\space\space
            so I gave up (last tried maximum value of
            [lines=\the\vwcol@Nlines])%
          }{%
            Text will be truncated in the multicolumns;
            please select the%
            ^^J\space\space
            number of lines explicitly or increase
            [maxrecursion=\the\vwcol@maxrecursion].%
          }%
        \fi
      \fi
    \repeat
    \usebox\vwcol@outputbox
```

If the `lines` was chosen explicitly then just run with it, giving an error if the lines were too small. I can imagine an `approxlines` option that varies the number of

---

[4] I think.

lines over a range of say, 5 lines up and down then chooses the best one, but I can't be bothered implementing that right now.

```
  \else
    \hbox to \vwcol@totalwidth{\vwcol@{\BODY}}%
    \unless\if@vwcol@boxready
      \vwcol@PackageError{%
        Not enough lines to fit the entire text;
        some text has been truncated.^^J\space\space
        Increase [lines=\the\vwcol@Nlines] to fit more%
      }{%
        Or remove [lines=\the\vwcol@Nlines] altogether
        to have 'vwcol' estimate the value.}%
    \fi
  \fi\par}
```

That's it!

`\vwcol@para@setup` Set up the paragraph options.

```
\def\vwcol@para@setup{%
```

Justification:

```
    \vwcol@justify
```

`\parindent` override if `justify` is `ragged` or `flush`:

```
    \@tempswafalse
    \ifx\vwcol@justify\RaggedRight
      \@tempswatrue
    \else\ifx\vwcol@justify\justifying
        \@tempswatrue
    \fi\fi
    \if@tempswa
      \parindent=\vwcol@parindent
    \else
      \vwcol@PackageInfo{%
        'indent' ignored for [justify=raggedleft]
        or [justify=center]}
    \fi
```

The algorithm, unfortunately, doesn't work with non-zero `\parskip`:

```
    \parskip=0pt}
```

`\vwcol@process@widths` This macros takes the `widths` input and calculates the number of columns and the total and average widths of the columns.

```
\def\vwcol@process@widths#1{%
```

Count the number of columns: (this must be done in a loop before the main one so that `\vwcol@Ncols` is known first)

```
  \@for\@ii:=#1\do{\advance\vwcol@Ncols 1}%
```

Based on the colsep and rule width, calculate allowable space. For stretchable column gaps, the separation gap counts as zero but the rules still take up some space:

```
  \ifdim\vwcol@sep=1sp
    \@tempdimb=\numexpr
      \linewidth-(\vwcol@Ncols-1)*\vwcol@rule
      \relax sp
```

And for fixed-width column gaps: (chuck in the warning here about `sep`≥`rule` coz it's convenient)

```
  \else
    \ifdim\vwcol@rule > \vwcol@sep
      \vwcol@sep=\vwcol@rule
      \vwcol@PackageWarning{%
        'sep' must be greater than or equal to 'rule'}%
    \fi
    \@tempdimb=\numexpr
      \linewidth-(\vwcol@Ncols-1)*\vwcol@sep
      \relax sp
```

Remember that the rules do not take up any space of their own between the columns, so they subtract from the white space of the separation gap; this must be mirrored when additional space is included before or after the columns:

```
    \if@vwcol@presep
      \advance\@tempdimb\dimexpr(-\vwcol@sep+\vwcol@rule)/2\relax
    \fi
    \if@vwcol@postsep
      \advance\@tempdimb\dimexpr(-\vwcol@sep+\vwcol@rule)/2\relax
    \fi
  \fi
```

The prerule and postrule both contribute to the total width, unlike the rules between the columns:

```
  \if@vwcol@prerule\advance\@tempdimb-\vwcol@rule\fi
  \if@vwcol@postrule\advance\@tempdimb-\vwcol@rule\fi
```

`\@tempdimb` now contains the maximum width that the columns can span before the environment is wider the `\linewidth`, after the rules and gaps are added in too. Use this as the reference length to calculate the lengths of the columns that have widths specified as ratios.

Now iterate to do stuff:

```
  \@for\@ii:=#1\do{%
```

If the column width is a plain rational number (like `0.4`) then set the columnwidth to be that fraction of the allowable width.

```
    \vwcol@test@length\@ii\@tempdimb
```

Keep a running total of the total width being used:

```
    \advance\vwcol@totalwidth\@tempdima
```

Save the column widths for later in the `\parshape` processing:

```
    \expandafter\expandafter\expandafter\def
    \expandafter\expandafter\expandafter\vwcol@setup@parlines
    \expandafter\expandafter\expandafter{%
                \expandafter\vwcol@setup@parlines
                \expandafter\vwcol@addlines
                \expandafter{\the\@tempdima}}}%
```

End the loop. Finally, calculate the average width of the columns:

```
  \vwcol@averagewidth=\dimexpr \vwcol@totalwidth/\vwcol@Ncols \relax}
```

`\vwcol@setup@parlines` This is the macro used to locally store the setup for the `\parshape` line specification: (see a few lines back for the `\expandafter` fun of getting stuff into it)

```
\def\vwcol@setup@parlines{\let\vwcol@parlines\@empty}
```

`\vwcol@addlines` Adds paragraph specifications to `\vwcol@parlines`for a single column in the `\parshape`Ḟor *N* columns there will be *N* calls to this macro inside `\vwcol@setup@parlines`, which gets expanded at the beginning of every paragraph to create the required `\parshape` specification.

`\@tempcntb` starts at `0` at the beginning of each paragraph and counts the number of lines over all the columns. `\vwcol@last` is the total number of lines that have so far been put into the columns. `\vwcol@parlines` is initialised at the beginning of each paragraph.

Each time `\vwcol@addlines` is executed, `\@tempcnta` iterates through each line in that column. Once the total line count reaches the number of lines that have been typeset, `\vwcol@parlines` starts filling up with `\parshape` lines for the next paragraph.

```
\def\vwcol@addlines#1{%
  \@tempcnta=0
  \loop\ifnum\@tempcnta<\vwcol@Nlines
    \advance\@tempcntb 1
    \ifnum\@tempcntb>\vwcol@last
      \xdef\vwcol@parlines{\vwcol@parlines 0cm #1 }%
    \fi
    \advance\@tempcnta 1
  \repeat}
```

`\vwcol@` This is the macro for splitting the text into variable-width columns.

```
\newcommand\vwcol@[1]{%
```

Setting the paragraphs First set the text into a special box that varies width at the appropriate places so when it is split into equal segments they can be arranged into variable-width columns.

```
\setbox\vwcol@box\vbox{%
```

The trick is to keep a running counter of lines that we've gone through by inspecting every paragraph after it is typeset:

```
  \def\par{\endgraf\advance\vwcol@last\the\prevgraf}%
```

(see `\vwcol@addlines` for a more detailed explanation):

```
  \everypar{%
    \@tempcntb=0
    \vwcol@setup@parlines
    \parshape=\numexpr \vwcol@Nlines*\vwcol@Ncols-\vwcol@last \relax
    \vwcol@parlines}%
```

Insert a `\strut` at the top to ensure we chop off the first column at the same height as all the others:

```
  \noindent\strut#1}%
```

Splitting the columns First insert a pre-sep and -rule, if appropriate:

```
\if@vwcol@presep
  \if@vwcol@prerule
    \begingroup
      \color{\vwcol@rulecol}
      \vrule width \vwcol@rule
    \endgroup
  \fi
  \hskip\dimexpr (\vwcol@sep-\vwcol@rule)/2 \relax
\fi
```

Iterate over the total number of columns:

```
\@tempcnta=0
\loop\ifnum\@tempcnta < \vwcol@Ncols
  \advance\@tempcnta 1
```

Skip the separations and rules in the first case:

```
  \unless\ifnum\@tempcnta=1
```

Sep and rule between the columns if `[sep=fill]`:

```
    \ifdim\vwcol@sep=1sp
      \hfill
      \begingroup
        \color{\vwcol@rulecol}
        \vrule width \vwcol@rule
      \endgroup
      \hfill
    \else
```

Sep and rule between the columns if `sep` is a length:

```
      \@tempdima=\dimexpr (\vwcol@sep-\vwcol@rule)/2 \relax
      \hskip\@tempdima
      \begingroup
        \color{\vwcol@rulecol}
        \vrule width \vwcol@rule
      \endgroup
      \hskip\@tempdima
    \fi
  \fi
```

Split off and place the text column, then loop:

```
  \vsplit\vwcol@box to \numexpr
    (\vwcol@Nlines-1)*\baselineskip+\ht\strutbox \relax sp
\repeat
```

Finally place the post-sep and -rule, if appropriate:

```
\if@vwcol@postsep
  \hskip\dimexpr (\vwcol@sep-\vwcol@rule)/2 \relax
  \if@vwcol@postrule
    \begingroup
      \color{\vwcol@rulecol}
      \vrule width \vwcol@rule
    \endgroup
  \fi
\fi
```

If `\vwcol@box` is void then we've used up all the material. This fact is passed on so we can re-run the algorithm with a different number of lines (or give a warning) if the material was truncated.

```
\ifvoid\vwcol@box
  \global\@vwcol@boxreadytrue
\fi}
```

11 *Problem with* `\raggedright` *and* `\parshape`

Check it out; when you make a `\parshape` with more lines specified than necessary, the linebreak of the first line is totally wrong:

```
\raggedright
\newlength\tmp\tmp=241.84842pt

\def\oneline{ 2.5em \tmp}
\def\fivelines{\oneline\oneline\oneline\oneline\oneline}

\textbf{Wrong}:\\
\parshape 10 \fivelines\fivelines
\lipsum[66]

\textbf{Right}:\\
\parshape 7 \fivelines\oneline\oneline
\lipsum[66]
```

Wrong:
Nunc
sed pede. Praesent vitae lectus. Praesent neque justo,
vehicula eget, interdum id, facilisis et, nibh. Phasellus
at purus et libero lacinia dictum. Fusce aliquet.
Nulla eu ante placerat leo semper dictum. Mauris
metus. Curabitur lobortis. Curabitur sollicitudin
hendrerit nunc. Donec ultrices lacus id ipsum.

Right:
Nunc sed pede. Praesent vitae lectus. Praesent
neque justo, vehicula eget, interdum id, facilisis
et, nibh. Phasellus at purus et libero lacinia dictum.
Fusce aliquet. Nulla eu ante placerat leo semper
dictum. Mauris metus. Curabitur lobortis. Curabitur
sollicitudin hendrerit nunc. Donec ultrices lacus id
ipsum.

This is why this package uses ragged2e's `\RaggedRight` instead of LaTeX's `\raggedright`.

In actual fact, when isolated into plain TeX code this reveals a legitimate bug in TeX's line-breaking algorithm. Unfortunately discovered too late in TeX's life to be awarded a cheque.