

Textpos: absolute positioning of text on the page

Norman Gray
(<http://nxg.me.uk>)

Version 1.8, 2016 June 7*

Contents

1	Description	2
1.1	Package options	3
1.2	Changing options on the fly	4
1.3	Configuration commands, and variants	4
1.3.1	Setting up a positioning grid	4
1.3.2	Box margin	5
1.3.3	Choosing the textblock reference point	5
1.3.4	Specifying textblocks with absolute sizes	6
1.4	Package parameters	7
1.5	Figure and table environments	8
2	Notes	9
2.1	Suggestions: Producing large-format posters	9
2.2	Absolute mode and <code>\newpage</code>	9
2.3	Interactions	10
2.4	Troubleshooting	11
3	History	11
4	Credits	13
5	Example	14

*Mercurial ident: 8aa202e2b283, 2016-06-07 23:52 +0100. This software is copyright, 1999, 2001-03, 2005-7, 2009-12, 2014-16. Norman Gray. It is released under the terms of the LaTeX Project Public License, either version 1.3 of this licence or (at your option) any later version. The latest version of this license is at <http://www.latex-project.org/lppl.txt>.

This package facilitates placing boxes at absolute positions on the L^AT_EX page. There are several reasons why this might be useful, but the reason which originally motivated this package is to help produce a large-format conference poster. However the facility is also useful for filling in forms or other special purpose layout.

This package provides a single environment, which contains the text (or graphics, or table, or whatever) which is to be placed on the page, and which specifies where it is to be placed.

The package tries not to get in the way. That is, you should be able to use most of the apparatus of L^AT_EX in your poster, such as section headings, citations, graphics inclusion, and so on. Please let me know if you experience problems in this respect.

This package requires the services of Martin Schröder's package `everyshi`. If this is not already part of your T_EX installation, you will need to download this package from CTAN. See <http://www.tex.ac.uk/tex-archive/macros/latex/contrib/supported/ms/> or one of the other CTAN hosts.

Textpos has a home page at <http://purl.org/nxg/dist/textpos>. The source is held at bitbucket: <https://bitbucket.org/nxg/textpos>, and there is an issues list there, for bug reports. Code contributions or fixes are welcome, but note that I feel that Textpos is pretty mature now, and I'm reluctant to extend its functionality beyond its natural boundaries, so it would be wise to chat to me about any new features before spending a lot of time drafting them in code.

An article describing Textpos appeared in TUGboat in 2002: Norman Gray, 'Absolute Positioning with Textpos', TUGboat **23** (3/4), pp341–4, 2002, available at <http://www.tug.org/TUGboat/tb23-3-4/tb75gray.pdf>.

1 Description

Load the package as usual, with

```
\usepackage{package-options}{textpos}
```

The *<package-options>* are as described in section 1.1.

textblock The environment is used as follows

```
\begin{textblock}{hsize}{(hpos),vpos)}
```

```
text...
```

```
\end{textblock}
```

The *<hsize>* and *<hpos>* arguments are given in units of a module `\TPHorizModule`, and *<vpos>* is given in units of a module `\TPVertModule`. You set these using the command `\setlength{\TPHorizModule}{<dimen>}`, and similarly for `\TPVertModule`. The arguments may be any dimension, and you may use the modules as units elsewhere in your document if you wish to, for example in `\makebox[2\TPHorizModule]{gnus}`. The text in the environment will be set in a box *<hsize>* modules wide, and placed on the page with its upper left corner

at the position $(hpos, vpos)$. As is natural in \TeX , the $\langle vpos \rangle$ parameter indicates distance *down* from the ‘anchor point’ (see below).

The `{textblock}` parameters $\langle hsize \rangle$, $\langle hpos \rangle$ and $\langle vpos \rangle$ are *multiples or fractions of the horizontal and vertical modules*, as appropriate. If you want or need to give explicit sizes here, see the `{textblock*}` environment below.

Notice that the positioning arguments for the `{textblock}` command – the coordinates $\dots (\langle hpos \rangle, \langle vpos \rangle)$ – are in *round* brackets, not curly ones. This is in imitation of the `picture` environment, and whether or not this is sensible, it’s not going to change now.

This package works in two modes, relative and absolute. In the first one, the default, the block-positioning parameters in the `{textblock}` environment are taken to be relative to an ‘anchor point’ which is the current position on the page, that is, the place where (the bottom left of) a character would appear if it were typed at this point. This will be appropriate if you are laying out text within a figure environment or the like. In this mode, you will typically give several `{textblock}` environments one after the other, so that they are all relative to the same point.

If, however, your entire document is to be laid out piece by piece (which is the case in the canonical use of the package, to lay out a poster), then you might want to be more sure of where the origin is. In this case, you make the package work in its absolute mode, by invoking it with the `[absolute]` option: `\usepackage[absolute]{textpos}`. In this mode, all the block-positioning parameters are given relative to a single origin on the page. By default, this ‘anchor point’ is the top-left corner of the page, but you may change it with the command `\textblockorigin{\langle hpos \rangle}{\langle vpos \rangle}`. Here $\langle hpos \rangle$ and $\langle vpos \rangle$ are dimensions such as ‘10mm’, relative to the top-left corner of the paper. You may use this command only if the package was invoked with the `[absolute]` option. See also section 1.1 for how to alternate between modes, and section 2.2 for notes on the interaction with the `\newpage` command.

The `textblocks` are placed on the page in the order in which they appear in the file. This means that later `textblocks` will be placed on top of earlier ones, which may matter if one or other contains, for example, a block of opaque colour (I believe this to be true in practice for all output mechanisms, though I doubt it’s guaranteed in principle). This order was unspecified before `textpos 1.7e`; it was changed, and specified, in that version.

The `{textblock}` environment will most often be used in vertical mode. If it is called in horizontal (ie, paragraph) mode, however, it will silently create a paragraph break by inserting a `\par` command before the environment; it remains in vertical mode after the environment is finished. It should have no further effects on spacing, and if you find that it does, that’s a bug. If you try to use the environment when in maths mode, the package objects (as it should!).

1.1 Package options

There are several package options:

Round brackets!

relative & absolute mode

[showboxes] When you are laying things out, it can be useful to have the boxes drawn in for you. This option draws a box fitting closely round the set text.

[noshowtext] This suppresses the display of the text in each block (so it's not really usable without the **[showboxes]** option). The resulting box will be the correct size, but empty, unless the `\textblocklabel` command has been given. This can be useful when you are previewing a document.

[absolute] If this is present, then the positions on the `{textblock}` environment are taken to be absolute positions on the page. See above for more detail, and see section 2.2 for the interaction with the `\newpage` command. There is also a **[relative]** option, but since relative mode is, and will remain, the default, the **[relative]** option is redundant, and is here only for symmetry.

[overlay] When using the absolute-position mode, the textblocks are placed under any other text on the page. This is normally what you want, but if you have page contents, and they have something which *obscures* the textblocks (for example, a block of opaque colour), then the positioned textboxes disappear. In this case, specify the option **[overlay]**, to request that the positioned blocks of text overlay any other page contents, rather than being overlaid.

[verbose], [quiet] The package writes a few messages to the output, describing its calculations. These are potentially irritating, so you can turn them off with the **[quiet]** option or on with the **[verbose]** option. The default is currently **[verbose]**, but this might change in future.

1.2 Changing options on the fly

`\TPoptions` Each of the options mentioned in the previous section can be changed within the body of the text, using the command `\TPoptions` with a comma-separated list of keywords `\langle keyword \rangle=true` or `\langle keyword \rangle=false`. The recognised keywords are 'absolute', 'overlay', 'verbose', 'showboxes' and 'showtext'. Thus the command

```
\TPoptions{absolute=false , showboxes = true }
```

will switch off **[absolute]** mode, and switch on **[showboxes]** mode. As this example illustrates, you can include whitespace in the specification; the arguments *must* be either `true` or `false`, or else bad things will happen.

You can switch between absolute and relative mode within a page. If a document is to use absolute mode *anywhere* within it, however, it must be *started* in absolute mode, with the **[absolute]** option to the `\usepackage` command.

1.3 Configuration commands, and variants

1.3.1 Setting up a positioning grid

`\TPGrid` You will often wish to set up a grid on your page. Rather than calculate and specify the two modules explicitly, you can set up the grid with a command `\TPGrid{\langle nhoriz \rangle}{\langle nvert \rangle}`, which sets `\TPHorizModule` to be *paper*

$\langle width \rangle / \langle nhoriz \rangle$, and `\TPVertModule` to be $\langle paper height \rangle / \langle nvert \rangle$. This takes an optional pair of dimension arguments, which specify a coordinate, as follows.

```
\TPGrid[ $\langle x \rangle$ , $\langle y \rangle$ ]{ $\langle nhoriz \rangle$ }{ $\langle nvert \rangle$ }
```

If these are present, then the modules are set up to leave a border of the given size around the grid. That is, `\TPHorizModule` is set to be $(\langle paper width \rangle - 2\langle x \rangle) / \langle nhoriz \rangle$, and similarly for `\TPVertModule`. Further, if the package was given the `[absolute]` option, then the text origin is set to be $(\langle x \rangle, \langle y \rangle)$ through a call to `\textblockorigin` (see below). For example, the declaration

```
\TPGrid[40mm,20mm]{10}{5}
```

would choose `\TPHorizModule` and `\TPVertModule` so as to give a grid of 10 intervals across and 5 intervals down, after leaving 40mm of a border on the right and left sides, and a 20mm border top and bottom.

1.3.2 Box margin

`\TPMargin` By default, the box that is positioned by the `{textblock}` environment is a tight fit to the block of text (or other material) inside it. This looks rather odd when you also use the `\textblockcolour` macro described below, or specify a larger than default `\TPboxrulesize`, in order to get a noticeable border round a piece of text. In those cases, you will want to request a non-zero margin around your text. If you give the command

```
\TPMargin{ $\langle size \rangle$ }
```

then the block of text inside the `textblock` will be decreased in width, enough to give the specified margin on each side. That is, the $\langle hsize \rangle$ of the `textblock`, and thus the coloured block (or the edge of the displayed border), remains the same, but the text width inside it decreases. The parameter $\langle size \rangle$ may be any non-negative dimension, and may as usual be in units of `\TPHorizModule` or `\TPVertModule`. The default behaviour is recovered by giving a $\langle size \rangle$ of `0pt`.

`\TPMargin*` There is a starred variant of this command, `\TPMargin*{ $\langle size \rangle$ }`, where the argument must again be non-negative. In this case, the text block inside the box is set with the $\langle hsize \rangle$ specified in the `{textblock}` environment, but the coloured block is increased in size, such that there is again a margin of the specified size around the text block.

1.3.3 Choosing the textblock reference point

You may give an optional argument to the `{textblock}` environment, specifying which point in the box is to be placed at the specified point:

```
\begin{textblock}{ $\langle hsize \rangle$ }[ $\langle ho \rangle$ , $\langle vo \rangle$ ]( $\langle hpos \rangle$ , $\langle vpos \rangle$ )
text...
\end{textblock}
```

The coordinates $\langle ho \rangle$ and $\langle vo \rangle$ are fractions of the width and height of the text box, respectively, and state that the box is to be placed so that the reference point $(\langle ho \rangle, \langle vo \rangle)$ within the box is to be placed at the point $(\langle hpos \rangle, \langle vpos \rangle)$ on the page. The default specification is $[0,0]$, indicating the top left of the box; the argument $[0,1]$ (for example) would specify the bottom left, and $[0.5,0.5]$ the middle.

If the margin is non zero, then the position identified by $[\langle ho \rangle, \langle vo \rangle]$ is slightly subtle:

- if the margin was specified with `\TPMargin`, then these coordinates are relative to the box *including* the margin;
- if the margin was specified with `\TPMargin*`, then the coordinates are relative to the contents of the `{textblock}`, *excluding* the margin.

For example, in the default case where the positioning argument is $[0,0]$, `\TPMargin` will cause the top-left of the surrounding box to be at position $(\langle hpos \rangle, \langle vpos \rangle)$ (and the text to be narrower), but `\TPMargin*` will cause the top left of the enclosed text to be at $(\langle hpos \rangle, \langle vpos \rangle)$ (and the enclosing box to be wider).

Note: This behaviour was somewhat underspecified in versions of `textpos` before v1.8, and in consequence inconsistently implemented. The rationalisation here *may* change documents which relied on the previous behaviour.

1.3.4 Specifying textblocks with absolute sizes

`textblock*` There is an alternative, starred, form of the `{textblock}` environment. In the argument to the `{textblock*}` environment, the block width, and the block position (but *not* the specification of the block reference point) are given as absolute dimensions, rather than as numbers in units of the horizontal and vertical modules. Thus

```
\begin{textblock*}{\langle hsize \rangle}[\langle ho \rangle, \langle vo \rangle](\langle hpos \rangle, \langle vpos \rangle)
text...
\end{textblock*}
```

produces a `textblock` of the given size, where this time $\langle hsize \rangle$, $\langle hpos \rangle$ and $\langle vpos \rangle$ are absolute dimensions, but $\langle ho \rangle$ and $\langle vo \rangle$ are still pure-number offsets (that is, fractions of the width and height of the `textblock`), as above.

Each `{textblock}` environment takes up zero space on the page (which means, by the way, that it cannot detect that it's overprinting or being overprinted), so you can (and typically will) use several of the environments in a row to scatter text all over the page.

The package is compatible with the `calc` package, so that you may use `calc`-style expressions when specifying lengths. Thus

```
\usepackage{calc}
\textblockorigin{56.9055pt-10mm}{0pt+1cm}
\begin{textblock*}{10mm+14cm}(0.3cm*5, 10\TPVertModule+5mm)
```

```
text...
\end{textblock*}
```

Note that you can only use calc-style expressions where you would specify a length with units, such as the width and location arguments of `{textblock*}` or the arguments to `\textblockorigin` – you can't use them when specifying a length in units of the horizontal and vertical modules, such as in the width and location arguments to the (unstarred) `{textblock}` environment.

1.4 Package parameters

<code>\TPHorizModule</code>	\TPHorizModule The length unit which is used for the horizontal positioning and size parameters of the <code>{textblock}</code> environment. Set it using the command <code>\setlength{\TPHorizModule}{<dimen>}</code> (or indeed <code>\addtolength</code>). The default is one sixteenth of the paper width.
<code>\TPVertModule</code>	\TPVertModule The length unit which is used for the vertical positioning and size parameters of the <code>{textblock}</code> environment. Set it using the command <code>\setlength{\TPVertModule}{<dimen>}</code> (or <code>\addtolength</code>). The default is one sixteenth of the paper height.
	\TPshowboxestrue and \TPshowboxesfalse You can control whether text blocks have the rule around them by using the <code>\TPshowboxestrue</code> and <code>\TPshowboxesfalse</code> commands. The <code>[showboxes]</code> option simply sets the initial value of this switch.
<code>\TPboxrulesize</code>	\TPboxrulesize When you use the <code>[showboxes]</code> option, the lines drawn are of this width. If this too small to show up when you are previewing your document, or if you simply like bold frames and wish to make them a feature of your poster's design, you may adjust the size using <code>\setlength</code> or <code>\addtolength</code> . The default is 0.4pt. See also the <code>\textblockrulecolour</code> command.
<code>\textblocklabel</code>	\textblocklabel This may be used within any <code>{textblock}</code> environment. It is ignored, unless the <code>[noshowtext]</code> option has been specified, when it will be used to label the textblock it is inside. Use: <code>\textblocklabel{Identifying text}</code> .
<code>\showtextsize</code>	\showtextsize When <code>\textblocklabel</code> is being shown, the text appears in size <code>\showtextsize</code> , which is defined by default to be <code>\normalsize</code> . If this is too small, you may adjust it using <code>\newcommand{\showtextsize}{\large}</code> , or whatever size you prefer.
<code>\textblockorigin</code>	\textblockorigin Sets the position of the top-left of the printable area. See above.
<code>\textblockcolour</code>	The text blocks can be coloured in. If you load the <code>color</code> package, then the commands of that package, <code>\textcolor</code> , <code>\pagecolor</code> and the like, should work as usual. The <code>textpos</code> package adds a new command, <code>\textblockcolour</code> . If you give the command

`\textblockcolour{<colour>}`

all text blocks following will have their background filled with the specified colour, which must be one of the standard colours or have been declared in a `\definecolor` declaration in the document preamble. This colour may be overridden for individual text blocks by giving this command within the `{textblock}` environment. If you wish a block not to have any background colour, you can suppress it, again for one block at a time, with the command `\textblockcolour{}` inside the `{textblock}` environment.

`\textblockrulecolour` You can similarly change the colour of the borders around the text block. If you give the command

`\textblockrulecolour{<colour>}`

then following text blocks will have their border in the given colour, which must again be either one of the standard ones or declared in the document preamble.

`\textblockcolor` For the benefit of those who observe Mr. Noah Webster's spelling reforms, `\textblockcolor` is defined as a synonym for `\textblockcolour`, but those who

`\tekstblokkulur` would condemn such anaemic half measures can use `\tekstblokkulur` instead.
`\textblockrulecolor` There are also the corresponding spelling-reform variants of `\textblockrulecolour`.
`\tekstblokkrookulur`

1.5 Figure and table environments

Textpos changes the behaviour of any `{figure}` and `{table}` environments *within* instances of the `{textblock}` environment, in such a way that the figure or table contents do not float away from the `{textblock}` environment. For the same reason, `\marginpar` is forbidden within a `textblock`. It makes no change, however, outside the environment, where figures float as normal, and you are still able to use *textblock* within figures, as described above. Within a `{textblock}`, these environments do nothing beyond accepting the usual `\caption` command, which behaves correctly with respect to caption numbering and `\label` commands (it also respects `\makecaption`, so you can tinker with that if you like that sort of thing). There's no real need to use either the `{figure}` or `{table}` environments within a `textblock` – you don't require them to allow `\includegraphics` or `{tabular}` to work, for example – but many people automatically use them to surround graphics or tables, and also expect to use these environments to number figures and tables within `{textblock}` environments; they are therefore here on a principle of least surprise.

The support here is admittedly simple, and it is known to fail in the case where there are `{figure}` (or `{table}`) environments both inside and outside `{textblock}`s on the same page (the LoF is ordered incorrectly in this case, due to the different times that the various environments write to the `.lof` file). I don't have immediate plans to fix this: the situation is surely sufficiently rare as not to justify the (potentially fragile) complication of the fix – if you disagree, let me know.

Since both L^AT_EX's floats mechanism (that is, `{figure}` and `{table}`) and the `[absolute]` mode are designed to move content around, we can't expect them to play

together nicely. With `[absolute]` mode on, the contents of a `{textblock}` inside a floating `{figure}` probably isn't going to end up where you expect it to. The only case I can think of where this would inconvenience you, is if you wanted some absolutely-positioned material to appear on the 'next' page. You might at first try to use a `\begin{figure}[p]` – that won't work, but the following will, if you first load the *afterpage* package:

```
\afterpage{%
  \newpage%
  \begin{textblock*}{297mm}(0mm,0mm)%
    \includegraphics{picture.png}% a full-page picture?
  \end{textblock*}%
  \null%
  \newpage}
```

This inserts a complete page, with some graphic on it, immediately after the end of the current page (thanks to Matthias Gloede for this technique).

2 Notes

2.1 Suggestions: Producing large-format posters

If you are producing a large-format poster, such as A0 size, you might want to use Gerlinde Kettl and Matthias Weiser's `a0poster` class, which painlessly deals with the miscellaneous hassles of printing to a large-format postscript printer.

I have a collection of suggestions for producing such posters at <http://purl.org/nxg/note/posters>.

The text on a large poster will typically use a very large font. It can be a hassle to create (or have dvips create) these fonts, and they take up a good deal of space on your disk. You might want to investigate the BlueSky/AMS fonts (available at CTAN), which are postscript versions of the Computer Modern fonts, and which can therefore be scaled arbitrarily.

2.2 Absolute mode and `\newpage`

You can sometimes get rather puzzling behaviour when you use `\newpage` in absolute mode.

When using the `[absolute]` option, you will often have all of the text on your page inside `{textblock}` environments. In this case, `TEX` does not believe that you have anything on the page at all, and so if you give a `\newpage` command to start a second sheet (perhaps you have a particularly generous poster space allocation at your conference, or you are filling out a form), `TEX` thinks it is redundant, and ignores it, so that all your `{textblock}` environments end up on just one page. To work round this, use `\null\newpage` instead: the `\null` (which produces a zero-width box) is enough to persuade `TEX` to respect the page-break.

This happens for the same reason that, also quite surprisingly, two `\newline` commands in a row do not produce a blank page.

2.3 Interactions

textpos & prosper

Textpos does not appear to get on terribly well with Prosper (a problem reported by Erik Van Eynde). This is because Textpos in absolute mode places its text onto the page at the last moment before the page is shipped out, which will be *after* Prosper has rotated the page from portrait to landscape format, so that the `{textblock}` contents end up at the correct position on the portrait page, but the wrong position on the landscape page. This unfortunately means that Textpos in absolute mode is incompatible with Prosper. But all is not lost: since Prosper slides start at a consistent point, you can fairly happily use Textpos in relative mode as long as the `{textblock}` environment is the first thing on the slide. For a similar reason, you should use the `[overlay,absolute]` options with Beamer.¹

... & beamer

In general, however, *anything* doing things at `\shipout` time (which includes Textpos in absolute mode) is going to be in a slightly precarious position with respect to anything else which plays games here.

textpos & color

There's also an unfortunate interaction with the `color` package. Textpos in absolute mode, and the `pdftex` graphics driver, both squabble over who gets to re-define the `\shipout` command: using the `\pagecolor` command with `pdflatex` causes textpos absolute mode to behave strangely². A fix may appear here in time, but for the present, there are three workarounds. The first is to place the `\pagecolor{...}` command before `\begin{document}`, which causes the redefinitions of `\shipout` to happen in a working order. If this is impossible for some reason, then (as with the Prosper workaround above) you can use textpos not in absolute mode: if the textblocks are the first printable material on the page, then they're anchored at a fixed position, and you should be able to use textblocks and `\pagecolor` with `pdflatex` much as normal. The final possibility is to use `latex`, `dvips` and `ps2pdf` to produce PDF, since the `dvips` driver happens not to have this problem (thanks to Joris Vankerschaver for the initial report, and to Heiko Oberdiek for one of the workarounds).

page grid

Gabriel Zachmann suggested having Textpos put a grid on the page, so that it is easier to work out `{textblock}` coordinates. I may yet do this, but it may not be necessary, since Rolf Niepraschk's `eso-pic` package can help you create this grid yourself. There is a vivid example of using Textpos along with Rolf's `eso-pic` package and the `calc` package on the Textpos web pages, at <http://purl.org/nxg/dist/textpos>, and an example of how to create a grid with TikZ at <http://tex.stackexchange.com/a/85088/96>.

textpos & texdraw

Finally, Robert Wenner reported a problem when using Textpos along with the `texdraw` package, with `\move(0,0)` apparently making a difference when it should be a no-op. I haven't worked out what's going on here, and further reports of this, ideally with a minimal example, would be most welcome.

¹This is also because the Beamer background frame overlays the textpos material, unless the `[overlay]` option is used to tell textpos to delay it. Thanks to Marius Arenz for useful comments here.

²See the `comp.text.tex` thread 'Colour in a0 poster' starting 2007 April 3

2.4 Troubleshooting

Switching to absolute mode with `\TPoptions` doesn't work In order for the [absolute] mode to work anywhere, the document has to be *started* in absolute mode (see Sect. 1.2). This is a hard-to-avoid limitation of the way that this mode is implemented.

Error: 'Missing number, treated as zero' If you give absolute values (that is, values with units) to the `{textblock}` environment, then you will be confronted with the rather opaque error 'Missing number, treated as zero'. Remember that the `{textblock}` environment requires relative sizes, and the `{textblock*}` environment requires sizes with units; see Sect. 1.3.4. You get a similar error message if you give relative sizes to the `{textblock*}` environment.

3 History

1.8, 2016 June 5 • Added the `\TPoptions` command, to switch modes on and off within the document. Various documentation tweaks.

- The behaviour of `\TPmargin` and `\TPmargin*` were somewhat under-specified in versions of `Textpos` before v1.8, and in consequence inconsistently implemented. This has now been rationalised, but the change *may* change documents which relied on the previous behaviour. Thanks to Richard Schreiber for the detailed bug report.

1.7j, 2014 January 3 Re-released under the LPPL.

1.7i, 2012 November 10 Bugfix: further change to the way the color package is loaded (fixes issue 2); now finally fixed?

1.7h, 2012 June 1 Bugfix: further change to the way the color package is loaded. Some documentation tweaks. Pointers to bitbucket repository.

1.7g, 2010 September 30 Bugfix: change the way we handle the color package not being loaded replacement `\color` command is now robust. Thanks to Joseph Wright for the bugreport. Also adjusted documentation of reference points.

1.7f, 2009 May 28 The change in behaviour introduced in v1.7e is now documented (it was unspecified before, and 1.7e didn't commit itself one way or the other).

1.7e, 2009 March 29 Daniel Richard G noted that the order in which textblock contents was laid down on the page was counter-intuitive, since one would expect that later environments go 'on top of' earlier ones. This order was unspecified before this version, but I've changed this, satisfying a principle of least surprise (later ones now go 'on top').

- 1.7d, 2007 March 30** Axel Sommerfeldt suggested a further alternative approach, even more lightweight, and I incorporated a version of that.
- 1.7c, 2007 March 29** Giovanni Radilla reported a problem with captions, which meant that the captions weren't appearing properly in the list of figures. Dan Luecking and Axel Sommerfeldt analysed the problem precisely, and the latter provided code which I've incorporated in this fix.
- 1.7b, 2007 March 21** Robert Whittaker reported a problem with `\TPmargin`, which meant that lists and quotations (and other things which manipulated `\leftskip` and `\rightskip`) were not decreasing in size when you set `\TPmargin non-zero`. Fixed.
- 1.7a, 2006 September 2** Version 1.7 created an inadvertant dependency on the `{color}` package. Now, if you do not load that package, `\textblockrulecolour` will have no effect, rather than failing. `Textpos` will give you a warning in this case, reminding you to load the `{color}` package.
- 1.7, 2006 August 24** Added the `\textblockrulecolour` and `\TPshowboxes{true,false}` commands, to further control the display of the rules around the text blocks.
- 1.6b, 2006 August 10** Minor documentation fixes
- 1.6a, 2005 October 13** The overriding of the figure and table environments now also works when there is no previous environment to override.
- 1.6, 2005 August 30** Made `{calc}`-style dimensions to the `{textblock*}` argument work again (so *that's* what regression tests are for...). Override the figure and table environments within `textblock` environments, to avoid their surprising and undesirable interaction with `textblock`.
- 1.5b, 2005 June 13** The 1.5 release broke the `textblock` environment's optional argument, controlling the position of the reference point within the block. Fixed.
- 1.5a, 2005 March 26** Documentation fixes: added a section on the interaction between absolute mode and LaTeX's `\newpage` command.
- 1.5, 2005 March 23** Implement `\TPMargin` command, which causes a margin to appear round the blocks of text within `textblock` environments. This makes it easy to use blocks of colour which are larger than the block of text by a decent margin, or to put a border round `textblocks` by setting a suitably-sized margin and using the `showboxes` package option.
- 1.4, 2003 September 7** Changes in the handling of vertical spacing; inconsistent in some circumstances before. Slight (consequent) change to the algorithm which ensures that material is output in absolute mode even when the page is otherwise empty. See README for details. Version 1.3a will remain available for some time in case these fixes break things.

1.3a, 2003 June 24 Added the `\textblockcolour` command, to set the background colour of text blocks

1.3, 2003 June 24 (there was a release 1.3, but it was broken, and immediately replaced by 1.3a)

1.2b, 2002 July 1 Works around a bug present in at least one package, which leaves `box255` holding an `hbox` at the wrong moment

1.2a, 2002 April 28 Version 1.2 had an error, which caused a confusing error if you gave any fractional part in the arguments to the `{textblock}` environment. This was fixed in version 1.2a, which adds a `{textblock*}` environment (fully compatible with `calc`), and does not attempt to support `calc`-style expressions in the parameters to the unstarred `{textblock}` environment.

1.2, 2002 April 21 Rolf Niepraschk `niepraschk@ptb.de` provided code to make `textpos` compatible with the `calc` package

Version 1.1 Released in 1999

4 Credits

Olaf Maibaum, `Olaf.Maibaum@informatik.uni-oldenburg.de`, made an elegant improvement to an earlier version of this package, by producing the code which I've incorporated here as the 'absolute mode' (I'd had something like this before, but it was *very* kludgy).

Bjoern Pedersen, `bjoern@poseidon.org.chemie.tu-muenchen.de`, made the excellent suggestion that the horizontal and vertical modules should be independent, and provided code to implement this.

Rolf Niepraschk, `niepraschk@ptb.de`, provided the code changes which made `textpos` compatible with the `calc` package.

Wybo Dekker, `wybo@servalys.nl`, reported a problem when `box 255` was (erroneously) not a `vbox`, and passed on a fix from Hans Hagen.

Jenny Maresh and Matthias Jerg independently suggested that it would be useful to specify a margin around the block of text in a `{textblock}` environment. That resulted in the `\TPMargin` command (after an unconscionably long gestation period). Rusen Lu suggested that one should be able to specify the colours of the box borders, and that it would be useful to turn the bordering feature on and off within the file.

Axel Sommerfeldt provided elegant code to fix incorrect behaviour of `\caption` within the `{figure}` environment.

Section 2.3, above, lists numerous people who have provided problem reports about the interactions between `Textpos` and other packages, and provided suggestions for workarounds and fixes.

Thanks also for general bugreports and other suggestions to Jozef Bednarcik, Richard G Daniel, Wolfgang Fleischer, Greg Petriccione, Giovanni Radilla,

Richard Schreiber, Brian Stephanik, Robert Whittaker, Joseph Wright, and Joachim Wuttke,

If you've reported a bug or made a suggestion and I haven't credited you here, please do accept my apologies, and please let me know.

5 Example

Here is a short example file.

```
1 \example
2 \documentclass{article}
3
4 \usepackage[absolute]{textpos}
5
6 \setlength{\TPHorizModule}{30mm}
7 \setlength{\TPVertModule}{\TPHorizModule}
8 \textblockorigin{10mm}{10mm} % start everything near the top-left corner
9 \setlength{\parindent}{0pt}
10
11 \begin{document}
12
13 \begin{textblock}{3}(0,0)
14 This block is 3 modules wide, and is placed with its top left corner
15 at the 'origin' on the page. Note that the length of the block is not
16 specified in the arguments -- the box will be as long as necessary to
17 accomodate the text inside it. You need to examine the output of the
18 text to adjust the positioning of the blocks on the page.
19 \end{textblock}
20
21 \begin{textblock}{2}(2,1)
22 \textblocklabel{block two}
23 Here is another, slightly narrower, block, at position (2,1) on the page.
24 \end{textblock}
25
26 \begin{textblock}{3}[0.5,0.5](2,3)
27 This block is at position (2,3), but because the optional argument
28 [0.5,0.5] has been given, it is the centre of the block which is
29 located at that point, rather than the top-left corner.
30 \end{textblock}
31
32 \end{document}
33 \example
```