Textpos: absolute positioning of text on the page*

Norman Gray
(norman@astro.gla.ac.uk)

Version 1.4, 2003/09/07

This package facilitates placing boxes at absolute positions on the LaTeX page. There are several reasons why this might be useful, but the main one (or at least my motivating one) is to help produce a large-format conference poster.

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 LaTeX 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`. You will need to download this package from CTAN first. See `<http://www.tex.ac.uk/tex-archive/macros/latex/contrib/supported/ms/>` or one of the other CTAN hosts.

1 Description

`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 ⟨*vpos*⟩ parameter indicates distance *down* from the reference point.

[*] This file has version number 1.4, last revised 2003/09/07

The `textblock` parameters may be whole numbers or fractional. 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 `...(⟨hpos⟩,⟨vpos⟩)` – 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.

`\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{⟨nhoriz⟩}{⟨nvert⟩}`, which sets `\TPHorizModule` to be ⟨*paper width*⟩/⟨*nhoriz*⟩, and `\TPVertModule` to be ⟨*paper height*⟩/⟨*nvert*⟩. This takes an optional pair of dimension arguments, which specify a coordinate, as follows.

```
\TPGrid[⟨x⟩,⟨y⟩]{⟨nhoriz⟩}{⟨nvert⟩}
```

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 (⟨*paper width*⟩ − 2⟨*x*⟩)/⟨*nhoriz*⟩, and similarly for `\TPVertModule`. Further, if the package was given the 'absolute' option, then the text origin is set to be (⟨*x*⟩,⟨*y*⟩) 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.

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

```
\begin{textblock}{⟨hsize⟩}[⟨ho⟩,⟨vo⟩](⟨hpos⟩,⟨vpos⟩)
text...
\end{textblock}
```

The coordinates ⟨*ho*⟩ and ⟨*vo*⟩ are fractions of the width and height of the text box, respectively, and state that the box is to be placed so that the point (⟨*ho*⟩,⟨*vo*⟩) within the box is to be placed at the point (⟨*hpos*⟩,⟨*vpos*⟩) on the page. The default specification is [0,0], the top left of the box: [0,1] would be the bottom left, and [0.5,0.5] the middle.

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!).

`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*}{⟨hsize⟩}[⟨ho⟩,⟨vo⟩](⟨hpos⟩,⟨vpos⟩)
text...
\end{textblock*}
```

produces a textblock of the given size, where this time ⟨*hsize*⟩, ⟨*hpos*⟩ and ⟨*vpos*⟩ are absolute dimensions, but ⟨*ho*⟩ and ⟨*vo*⟩ are 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.

This package works in two modes, relative and absolute. In the first one, the default, the block-positioning parameters in the `textblock` envirionment are taken to be relative to the current position on the page. 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 is the top-left corner of the page, but you may change it with the command `\textblockorigin{`⟨*hpos*⟩`}{`⟨*vpos*⟩`}`. Here ⟨*hpos*⟩ and ⟨*vpos*⟩ 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.

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.

`\textblockcolour`

The text blocks can be coloured in. If you load the color package, then the commands of that package, `\textcolor`, `\pagecolor` and the like, should work perfectly well. The textpos package adds a new command, however, `\textblockcolour`. If you give the command

```
\textblockcolour{⟨colour⟩}
```

all text blocks following will have their background filled with the specified colour, which has to be declared, as usual, 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{}`.

`\textblockcolor`

For the benefit of those who observe Webster's spelling reforms, `\textblockcolor` is defined as a synonym for `\textblockcolour`, but those who would condemn such anaemic half measures can use `\tekstblokkulur` instead. `\tekstblokkulur`

1.1 Package options

There are several package options:

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.

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 overlayed.

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 Package parameters

`\TPHorizModule`

`\TPHorizModule` The length unit which is used for the horizontal positioning and size parameters of the textblock environment. Set it using the command `\setlength{\TPHorizModule}{⟨dimen⟩}` (or indeed `\addtolength`). The default is one sixteenth of the paper width.

`\TPVertModule`

`\TPVertModule` The length unit which is used for the vertical positioning and size parameters of the textblock environment. Set it using the command `\setlength{\TPVertModule}{⟨dimen⟩}` (or `\addtolength`). The default is one sixteenth of the paper height.

`\TPboxrulesize`

`\TPboxrulesize` When you use the showboxes option, the lines drawn are of this width. If this too small to show up when you are previewing your document, you may adjust the size using `\setlength` or `\addtolength`. The default is 0.4pt.

`\textblocklabel`

`\textblocklabel` This may be used within any textblock environment. It is ignored, unless the noshowtext option has been specified, when it will be used to label the textblock it is inside. Use: `\textblocklabel{Identifying text}`.

\showtextsize `\showtextsize` When `\textblocklabel` is being shown, the text appears in size `\showtextsize`, which is defined by default to be `\normalsize`. If this is too small, you may adjust it using `\newcommand{\showtextsize}{\large}`, or whatever size you prefer.

\textblockorigin `\textblockorigin` Sets the position of the top-left of the printable area. See above.

1.3 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://www.astro.gla.ac.uk/users/norman/docs/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 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.

3 Example

Here is a short example file.

```
⟨*example⟩
\documentclass{article}

\usepackage[absolute]{textpos}

\setlength{\TPHorizModule}{30mm}
\setlength{\TPVertModule}{\TPHorizModule}
\textblockorigin{10mm}{10mm} % start everything near the top-left corner

\begin{document}

\begin{textblock}{3}(0,0)
```

```
This block is 3 modules wide, and is placed with its top left corner
at the ‘origin’ on the page.  Note that the length of the block is not
specified in the arguments -- the box will be as long as necessary to
accomodate the text inside it.  You need to examine the output of the
text to adjust the positioning of the blocks on the page.
\end{textblock}

\begin{textblock}{2}(2,1)
\textblocklabel{block two}
Here is another, slightly narrower, block, at position (2,1) on the page.
\end{textblock}

\begin{textblock}{3}[0.5,0.5](1,3)
This block is at position (1,3), but because the optional argument
[0.5,0.5] has been given, it is the centre of the block which is
located at that point, rather than the top-left corner.
\end{textblock}

\end{document}
⟨/example⟩
```