IV LaTeX 2ε standard graphics and colour support

David Carlisle
carlisle@cs.man.ac.uk
Sebastian Rahtz
spqr@ftp.tex.ac.uk

1 Introduction

With the release of LaTeX 2ε, graphics file inclusion, rotation, scaling and colour macros are now a standard part of the system (though implemented as two separate packages). These are, of course, dependent on the abilities of the driver in use, as they are all implemented by using `\special` commands; it is hoped that until (if ever?) a common `\special` format is agreed upon, the standardized interface will make life easier for authors. This article *briefly* describes the facilities of the new packages, but for the full glory readers are advised to book for the July training meeting which covers this area.

Both these packages are now available on CTAN[1] in `macros/latex2e/packages/graphics`, but users are warned that documentation is incomplete, and that the *internal* interface is still open to change. Making the packages available is simple; if we start our document:

```
\documentclass[dvips]{article}
\usepackage{graphics}
\usepackage{color}
```

we load both packages, and the global option 'dvips' is passed to them, indicating which driver we are using (most common drivers are supported).

2 The graphics package

2.1 Rotation

Any TeX box can be rotated with the command

`\rotatebox{`*angle*`}{`*text*`}`

where *angle* degrees is measured anti-clockwise. Normally the rotation is about the left-hand end of the baseline of *text*, but more complex examples are possible. A simple example of rotation is

```
I like \rotatebox{45}{cats} but
I cannot bear \rotatebox{-45}{dogs}
```

which produces: I like cats but I cannot bear dogs ; note that the right amount of space is left for the rotated material.

2.2 Scaling

A box can be resized in two ways, by scale or by specifying the desired size. The first is achieved with

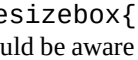
`\scalebox{`*h-scale*`}[`*v-scale*`]{`*text*`}`

If *v-scale* is omitted, the vertical scale factor is the same as the horizontal one. Thus `\scalebox{2}[.5]{Cats}` produces Cats.

Scaling to size is done with

`\resizebox*{`*h-length*`}{`*v-length*`}{`*text*`}`

[1] Subscribers to the Mac or DOS disk packages from UKTUG will receive copies during June with the first full release of LaTeX 2ε.

which resizes *text* so that the width is *h-length*. If `!` (exclamation mark) is used as either length argument, the other argument is used to determine a scale factor that is used in both directions. Normally *v-length* refers to the height of the box, but in the starred form, it refers to the 'height + depth'. As normal in LaTeX 2ε, box length arguments, `\height`, `\width`, `\totalheight`, `\depth` may be used to refer to the original size of the box. The cats can be resized again using `\resizebox{1in}{0.2in}{Cats}` to produce  .

The user should be aware that these scaling operations are done by the *driver*, and it is likely that bitmap fonts will be scaled, instead of a new size being selected from scratch.

2.3 Graphics inclusion

The basic command to include a graphics file is:

`\includegraphics*[`*llx,lly*`][`*urx,ury*`]{`*file*`}`

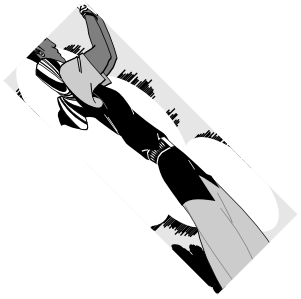
If `*` is present, then the graphic is 'clipped' to the size specified. If `*` is omitted, then any part of the graphic that is outside the specified 'bounding box' will over-print the surrounding text.

If the optional arguments are omitted, then the size of the graphic will be determined by reading the graphic file itself, if possible. If `[`*urx,ury*`]` is present, then it should specify the coordinates of the top right corner of the image, as a pair of TeX dimensions. If the units are ommited they default to `bp`. So `[1in,1in]` and `[72,72]` are equivalent. If only one optional argument appears, the lower left corner of the image is assumed to be at `[0,0]`. Otherwise `[`*llx,lly*`]` may be used to specify the cordinates of this point.

The package works by examining the suffix of the file name and looking that up in a rule-table, which tells it whether or not the file can be read for a size, how to do so, and what driver-specific macro to call for the type of file. This allows extensible support for whatever bitmap or vector graphic file types the driver can work with. The interested reader should consult the documentation for the interface to this system.

A combination of features in the graphics package allows us to rotate a scaled portion of a figure:

```
\rotatebox{45}{%
  \resizebox{1.5cm}{4cm}{%
  \includegraphics*%
    [100,100][500,600]{%
    golfer.ps}%
  }%
}
```



which produces

Those users familiar with the *(e)psfig* package can use an extended form of the graphics package (provisionally entitled `graphicx`) which offers a full 'key=value' interface to all the commands described above, using the generic `keyval` parser by David Carlisle described elsewhere in this issue of *Baskerville*. A small wrapper package (`epsfig`) provides an exact emulation of `psfig` (and Rokicki's `epsf` macros) for those with existing documents marked up in this way.

3 The colour package

The LaTeX 2ε colour support offers a variety of facilities:

- colouring text;
- colouring box backgrounds;
- setting the page colour;
- defining new colour names

3.1 Using colours

There are two text colouring commands, which work in the same was as the normal font-changing macros. The first one is a *command*:

`\textcolor{<colourname>}{<text>}`

This takes an argument enclosed in brackets and writes it in the selected colour. This can be used for local colour changes, since it restores the original colour state when it is completed, *e.g.*

```
This will be in black
\textcolor{Blue}{This text will be in blue}
and this reverts to black
```

The second colour macro is a *declaration*:

`\color{<colourname>}`

This colour macro takes only one argument and simply sets a new colour at this point, *e.g.*

```
\color{red} All the following text
        will be red.
\color{black} Set the text colour
        to black again.
```

The colour declaration does of course respect normal TEX grouping; if we write

```
We start in black, but now
{\color{red} all text
is in red, {\color{green} but this
should be in green} and this
should be back in red.}
And we finish in black
```

we will see[2]

We start in black, but now all text is in red, but this should be in green and this should be back in red. And we finish in black

The *background* of a normal LR TEX box can also be coloured:

`\colorbox{<colourname>}{<text>}`

This takes the same argument forms as `\textcolor`, but the colour specifies the background colour of the box.

There is an extended form:

`\fcolorbox{<colourname>}{<colourname>}{<text>}`

This has an extra *colourname* argument, and puts a frame of the first colour around a box with a background specified by the second colour.

The line width and the offset of the frame from the text are controlled by the standard `\fboxsep` and `\fboxrule` lengths.

TEX does not have internal support for colour attributes of text, and TEX 'grouping' across pages, floats, footnotes etc will not always yield the expected results. However, LATEX 2ε has extended support to cope with most situations, and it is hoped that more driver support will make this even better.

3.2 Defining new colours

The colour names 'white', 'black', 'red', 'green', 'blue', 'cyan', 'magenta' and 'yellow' are predefined by all driver files. New colour names can be defined with:

`\definecolor{<name>}{<model>}{<spec>}`

where *spec* is usually a list of comma-separated numbers needed by the *model*. Typically, drivers can cope with the models *gray*, *rgb* and *cmyk* (although the system is extensible), allowing, *e.g.* :

```
\definecolor{lightgrey}{gray}{.25}
\definecolor{cornflowerblue}{rgb}{.39,.58,.93}
\definecolor{GreenYellow}{cmyk}{0.15,0,0.69,0}
```

[2] The examples of colour like this will be set using gray scales.

It is also possible to use the `\textcolor` and `\color` macros with an explicit colour model and specifications, to avoid the overhead of defining new colors.

One of the important concepts inherited from James Hafner's `colordvi` macros is the allowance for a layer of colour 'names' above the actual specification given to the printer; Hafner worked out a set of 68 CMYK colours which correspond to a common set of Crayola crayons; these are predefined in the header files used by *dvips*, and the user calls them *by name*, allowing for tuning of the header files for a particular printer without changing the source. This system is provided by the LaTeX 2ε colour package for those drivers which support it, and its use is strongly recommended.