

NAME

epsutil – a tool for manipulating Encapsulated PostScript files

SYNOPSIS

```
epsutil [ BG=color-name ] [ BOUNDINGBOX=llx,lly,urx,ury ] [ CHECK=check ] [ CLIP=clip ]
[ CMYK=cyan,magenta,yellow,black ] [ FRAME=frame-thickness ] [ GRAY=gray ]
[ GRID=grid ] [ HEIGHT=height ] [ HSB=hue,saturation,brightness ]
[ ID=idx,idy[,fontsize[,fontname]] ] [ LINEWIDTHSCALE=line-width-scale ]
[ MAXLINEWIDTH=maximum-line-width ] [ MINLINEWIDTH=minimum-line-width ]
[ OXY=origin ] [ OX=llx-origin ] [ OY=lly-origin ]
[ PSCODE='POSTSCRIPT code' ] [ PSFILE=psfile,psfile,... ]
[ RGB=red,green,blue ] [ RGBFILE=rgbfile ] [ ROTATE=degrees ]
[ SXY=scale-factor ] [ SX=x-scale-factor ] [ SY=y-scale-factor ] [ WIDTH=width ]
[ < > old-PS-file > new-PS-file
```

DESCRIPTION

epsutil filters an Encapsulated POSTSCRIPT file, or files, named on the command line (or *stdin* if no file-names are given), to *stdout*, inserting wrapper POSTSCRIPT to optionally provide

- background color,
- BoundingBox adjustment and measurement,
- checking for non-EPS operators,
- clipping,
- framing,
- linewidth scaling,
- POSTSCRIPT alteration,
- rotation,
- scaling, and
- translation.

The output file fully conforms to the *POSTSCRIPT Document Structuring Conventions* defined in Appendix G of the *POSTSCRIPT Language Reference Manual*, Second Edition, Addison-Wesley (1985), ISBN 0-201-10174-2 (the Adobe red book).

The output of this program is also suitable input to it; after four filtering steps with 90-degree rotation selected, the original figure will be recovered, although the output POSTSCRIPT file will be about 4300 bytes larger for each rotation because of the wrapper overhead.

The original command line that was used to create the output file is preserved in a *%%Title* comment in the second line of the output file, as a record of the requested transformations.

A verbatim copy of the original file, minus any *%%EOF* comments, is embedded in the output between the *%%BeginDocument* and *%%EndDocument* lines, where it can be recovered simply by stripping the wrapping POSTSCRIPT.

OPTIONS

epsutil options take the form of **NAME=value** assignments, rather than the more usual hyphen-prefixed options common in other UNIX utilities. If an option is given more than once, only the last one is used.

If no dimension unit is specified in the variable values described below, physical dimensions are assumed to be measured in standard POSTSCRIPT units of big points:

$$72\text{bp} = 1\text{in} = 25.4\text{mm}$$

Otherwise, any of the \TeX dimension units of *bp* (big point), *cc* (cicero), *cm* (centimeter), *dd* (didot point), *in* (inch), *mm* (millimeter), *pc* (pica), *pt* (point), or *sp* (scaled point), may suffix a number, and the value will be converted to big points internally.

Background color

A background color can be specified as a named color, or as a triple (or, for **CMYK**, quadruple) of numbers, each in the range 0..1. The numbers must be separated by any character other than a digit or period. Comma or slash is a readable choice for a separator character, and avoids the need to quote the values to protect them from possible shell interpretation.

Five color models are available. The models are tried in alphabetical order: the first one specified is used, and any settings for the others are silently ignored.

For **BG**, the value is the name of a color found in either the file specified by the **RGBFILE**=*rgbfile* option, or else in the X Window System *rgb.txt* color mapping file. These files associate red, green, and blue intensities with named colors; their lines contain three numbers in the range 0..255, followed by a color name. The *rgb.txt* file is found by searching a built-in directory path that covers all major UNIX systems; that path can be overridden by defining a value for the environment variable **RGBPATH**. For example, on an IBM RS/6000 AIX system, you could set that variable to */usr/lpp/X11/lib/X11/It_IT* to get Italian color names.

In **CMYK**, values for cyan, magenta, and yellow are provided (0 means no color, 1 means full color), and then an amount of black (0 means none, 1 means completely black) is added to darken the color.

In **GRAY**, 0 is black and 1 is white. **GRAY**=*g* is equivalent to **RGB**=*g,g,g*.

In **HSB**, hue runs through the rainbow (approximately, red=0, orange=0.1, limegreen=0.2, green=0.3, cyan=0.5, blue=0.6, magenta=0.8, deeppink=0.9, red=1). Saturation goes from 0 (no color == black) to 1 (full color). Brightness goes from 0 (dark) to 1 (bright).

In **RGB**, major colors are:

```
Black 0,0,0
Red   1,0,0
Green 0,1,0
Yellow 1,1,0
Blue  0,0,1
Magenta 1,0,1
Cyan  0,1,1
White 1,1,1
```

Sample options:

```
BG=bisque
CMYK=1,0,0,0.2
GRAY=0.9
HSB=0.5,1,,0.8
RGB=1,1,0
RGBFILE=mycolors.rgb
```

BoundingBox

Encapsulated POSTSCRIPT files are required to contain a *%%BoundingBox* comment that defines the lower-left and upper-right corner coordinates in big points. Sadly, most POSTSCRIPT-producing programs produce (sometimes wildly) incorrect coordinate values in this comment, so manual adjustment is frequently necessary. Also, sometimes only a portion of the original figure is required, necessitating definition of a new BoundingBox.

To avoid the need for modifying the original POSTSCRIPT file, the two corner coordinate pairs can be specified as four optionally-dimensioned numbers for the value of the **BOUNDINGBOX** option. The numbers can be separated by any characters other than plus, minus, digits, period, or lowercase letters; comma or slash are readable choices. Values specified this way override those in the *%%BoundingBox* comment.

When a **BOUNDINGBOX** value is given, especially if it covers only a portion of the original figure, clipping should be selected.

Although fractional dimensions are accepted, after converting to big points, the values actually used will be rounded down to the nearest integer at the lower-left corner, and up to the nearest integer at the upper-right corner. This is a requirement of the *POSTSCRIPT Document Structuring Conventions*.

Sample options:

BOUNDINGBOX=10,20,500,750

BOUNDINGBOX=10 20 500 750

BOUNDINGBOX=0.1in,0.3in,18cm,25cm

BOUNDINGBOX=10mm/20mm/30pc/40pc

BoundingBox frame

A rectangular frame of a specified line thickness will be drawn around the *outside* of the original BoundingBox if the **FRAME**=*frame-thickness* option is given.

If linewidth scaling is requested, the frame will also be affected by the scaling.

The output BoundingBox will automatically be enlarged by the twice the frame thickness, and any **MAX-LINEWIDTH** value will be adjusted upward if necessary to ensure that the frame can be drawn properly.

Sample options:

FRAME=3

FRAME=3mm

Checking for non-EPST operators

If a non-zero value of the **CHECK**=*check* option is given, check for the presence of POSTSCRIPT operators that are not permitted in EPS files.

Because this option requires additional pattern matching on every input line, and is only occasionally needed, the default is **CHECK**=0 to suppress the check.

Sample options:

CHECK=0

CHECK=1

Clipping

If a zero value of the **CLIP**=*clip* option is given, suppress clipping the figure to the BoundingBox.

Although EPS-including programs are supposed to provide this clipping automatically, not all do so. Thus, the default in this program is to turn on clipping.

The clipping path is set after geometric transformations have been completed, but before any drawing operations are performed. Thus, the BoundingBox frame, and the background color, are subject to clipping.

Sample options:

CLIP=0

CLIP=1

Grid

A non-zero value for the **GRID**=*grid* option requests that a grid be drawn over the picture to allow convenient determination of BoundingBox coordinates for a portion of the view.

Clipping is automatically suppressed when a grid is selected, in order for the complete grid, with its edge labels, to be visible.

Sample options:

GRID=0

GRID=1

Identifying label

When working with large numbers of figures, or experimenting with various options, it is sometimes difficult to keep track of them. The **ID** option can be specified to provide a page location where an identifying label recording the command line will be printed. The label location is measured on the output page, relative to the lower-left corner, independent of geometric transformations. It is drawn before any background fill color is supplied, and will be overwritten by that fill if it lies within the BoundingBox of the output figure.

An optional third value defines the font size (default: 10bp), and an optional fourth value defines the font name (default: *Helvetica*).

Values are separated by characters other than plus, minus, digits, period, or letters.

Sample options:

```
ID=50,750
ID=0.4in/0.4in/12pt
ID= '10mm 10mm 8pt Helvetica-Narrow'
```

Linewidth control

Sometimes, linewidths need adjustment, usually to thicken them for better reproduction. The **LINEWIDTHSCALE** value causes all linewidths to be multiplied by that value. This scaling also affects any framing rectangle that is requested.

POSTSCRIPT allows zero linewidths, which means the thinnest line that the output device is capable of producing. Such linewidths would not be affected by scaling. Thus, a value for **MINLINEWIDTH** can be given to force all lines to be at least that thick, and therefore, subject to scaling.

To prevent scaling from producing ridiculously thick lines, a **MAXLINEWIDTH** value can be set to force an upper limit to linewidths. If not specified, it defaults to 24.

If linewidth scaling is applied to a POSTSCRIPT file that already has linewidth scaling, the two scalings are *not* cumulative: only the last scaling applied will have any effect. This unfortunate irregularity is a consequence of the inability of the POSTSCRIPT language to permit redefinition of an operator to more than one level; attempts to do otherwise simply produce infinite loops in the POSTSCRIPT interpreter.

Sample options:

```
LINEWIDTHSCALE=3
MAXLINEWIDTH=6
MAXLINEWIDTH=6bp
MINLINEWIDTH=0.1pt
```

POSTSCRIPT insertions

Two options, **PSCODE** and **PSFILE**, provide for insertion of additional POSTSCRIPT code into the end of the `%%BeginProlog ... %%EndProlog` section where POSTSCRIPT commands are defined.

The value of **PSCODE** is a string containing arbitrary literal POSTSCRIPT code.

The value of **PSFILE** is the name of one or more comma- or space-separated files whose contents are to be inserted.

When both options are specified, **PSFILE** is handled before **PSCODE**, so that any POSTSCRIPT read from files can be overridden by literal POSTSCRIPT from the command line.

Sample options:

```
PSCODE='0 1 0 setrgbcolor'
PSCODE='/setrgbcolor {add add 3 div setgray} bind def'
PSFILE=mymacros.eps
PSFILE=header1.eps,.../tools/monochrome.eps
```

Rotation

The **ROTATE** option value is a rotation angle in degrees, positive counterclockwise, and negative clockwise. Only multiples of +/-90 degrees are supported. Non-conforming values will elicit a warning and be ignored, preventing any rotation.

For example, to turn landscape orientation with the right side at the top of the page into portrait orientation, specify **ROTATE**=-90. It will usually be necessary to supply additional scaling (**HEIGHT** or **WIDTH**, or **SX**, **SY**, or **SXY** options) to make the picture fit on the output page.

Unless otherwise specified by the translation variables (see below), the position of the lower-left corner of the BoundingBox on the output display device is kept invariant, so that after a -90-degree rotation, the lower-right corner of the original BoundingBox will be positioned where the lower-left corner was.

Sample options:

```
ROTATE=0
```

ROTATE=-90
ROTATE=90
ROTATE=180
ROTATE=270
ROTATE=360
ROTATE=450

Translation

The origin option variables move the lower-left corner of the BoundingBox to (**OX,OY**), relative to the lower-left page corner, or if **OX** and **OY** are not specified, then to (**OXY,OXY**). Otherwise, the lower-left corner of the new BoundingBox will be at the same physical page location as one of the corners of the original picture; which one depends on the rotation angle chosen.

Sample options:

OX=0
OY=0
OX=10mm
OY=36bp
OXY=1in
OXY=3pc

Scaling (Absolute)

The **HEIGHT=height** and **WIDTH=width** options provide a way to request absolute scaling of the BoundingBox to a specified height or width. If only one of them is given, then the other is computed to preserve the aspect ratio for distortion-free scaling. If both are given, then the horizontal and vertical scale factors may be different; the aspect ratio will then not be preserved.

The two sizes refer to the final figure, *after* any rotation.

If both absolute (**HEIGHT** and **WIDTH**) and relative (**SX**, **SY**, and **SXY**) scaling are specified, then absolute scaling is done, and the relative scale factors are ignored.

Sample options:

HEIGHT=400bp
HEIGHT=15cm
WIDTH=340
WIDTH=150mm

Scaling (Relative)

The **SX**, **SY**, and **SXY** options provide for relative scaling. Any scale factors provided are applied *after* the rotation, so that they apply to the final figure. **SX** and **SY** are used if provided; otherwise, **SXY** is used for both *x* and *y* scaling if provided.

Sample options:

SX=0.5
SY=2.0
SXY=0.7727

HAND MODIFICATION

The output file can be easily modified by hand later to adjust any of the options that can be set on the command line. Look for the lines following the first `%%BeginProlog` comment that begin `/NAME ... def`, and then change the numeric values as needed. Comment lines precede each set of related variable definitions to provide suitable documentation.

BUGS

Until a valid `%%BoundingBox` comment has been located, or a **BOUNDINGBOX** command-line option has been supplied, the input POSTSCRIPT is buffered in memory. On small machines, or with very large POSTSCRIPT files that contain the `%%BoundingBox` comment near the end instead of near the beginning, it is possible that your computer memory (and swap space, if any), might be exceeded. The easiest solution for such a problem is to supply a command-line **BOUNDINGBOX** value. This problem is expected to be

exceedingly rare.

Encapsulated POSTSCRIPT is a subset of standard POSTSCRIPT, and excludes approximately twenty operators whose use can prevent successful inclusion of a POSTSCRIPT inside another POSTSCRIPT file, usually because those operators would obviate the use of a local independently-scaled coordinate system, or would modify the global state of the POSTSCRIPT environment. When the output file fails to conform to the specified transformations, it is usually because of the use of these excluded operators. If requested by the **CHECK=1** option, **epsutil** looks for those operators, and will raise warnings when they are seen. Unfortunately, their apparent absence does not mean that they are not used, because POSTSCRIPT files can also contain binary, hexadecimal, and other encodings that can hide such operators from view. Thus, the only truly reliable check is visual examination of the output of **epsutil** with a POSTSCRIPT previewer or printer, and its successful incorporation in other documents, such as with \TeX `\epsffile` or `\special` commands, or in word processors.

ENVIRONMENT VARIABLES

RGBPATH Search path (a colon-separated list of directories) for the X Window System color definition file, *rgb.txt*.

FILES

rgb.txt Color mapping file distributed with the X Window System.

AVAILABILITY

The master source distribution for **epsutil** is maintained at the Internet archive location <ftp://ftp.math.utah.edu/pub/misc/index.html#epsutil-1.00> for both anonymous ftp and World-Wide Web access. The source code is in the public domain, and copies of the program may be freely distributed and used for any purpose.

SEE ALSO

dvialw(1), **dvips(1)**, **enscript(1)**, **genscript(1)**, **ghostscript(1)**, **gs(1)**, **lptops(1)**, **pageview(1)**, **plot79(1)**, **postscript(7)**, **psposter(1)**, **tek2ps(1)**, **tekalw(1)**, **transcript(1)**, **x79ps(1)**, **xpsview(1)**.

AUTHOR

Nelson H. F. Beebe
 Center for Scientific Computing
 Department of Mathematics
 University of Utah
 Salt Lake City, UT 84112
 USA
 Tel: +1 801 581 5254
 FAX: +1 801 581 4148
 Email: <beebe@math.utah.edu>
 URL: <http://www.math.utah.edu/~beebe>