NAME

libmeta.a - simplified interface to metafile(5)

SYNOPSIS

extern FILE *pout;

mline(x, y, type, thick, color)

mpoly(x, y, border, pat, color)

mdraw(x, y)

mtext(x, y, s, cpi, color)
char *s;

mvstr(xmin, ymin, xmax, ymax, s, d, thick, color)
char *s;

mrectangle(xmin, ymin, xmax, ymax, pat, color)

mtriangle(xmin, ymin, xmax, ymax, d, pat, color)

msegment(xmin, ymin, xmax, ymax, sname, d, thick, color)
char *sname;

msetpat(pat, pattern)
char *pattern;

mopenseg(sname)
char *sname;

mcloseseg()

minclude(fname)
char *fname;

mendpage()

mdone()

DESCRIPTION

The routines in *libmeta* provide a simple interface to the metafile(5) 2D graphics stream. Output from these routines is sent to *pout. Pout* defaults to the standard output, and should be piped to the appropriate device driver.

All coordinates range from 0 to 16383 and map to a square area on the output device. *D* values are one of 'r', 'u', 'l' and 'd' corresponding to right, up, left, and down respectively. *Color* values range from 0 to 3 and normally correspond to black, cyan, green and blue. *Pattern* values range from 0 to 3 and default to solid, thick lines, thin lines, and candystripe. Pattern value mapping may be changed via *setpat.* All strings are null-terminated, and do not contain newlines.

Mline starts a line at the given coordinates. The line *type* is a number from 0 to 3 corresponding to solid, dashed, dotted, and dot-dashed. The line thickness, *thick,* is a number from 0 to 3. Connected lines are drawn with successive calls to *mdraw.*

Mpoly starts a polygon at the given coordinates. The boolean *border* specifies whether or not a border is desired around the polygon. *Mdraw* is used to add vertices to the polygon. The polygon will be closed automatically after the last call.

Mtext prints a string of hardware characters starting at the given coordinates. The characters per inch are *cpi.* Text is always oriented to the right.

Mvstr places a vector character string within the given boundaries. The string is oriented according to *d.* The character line thickness is given by *thick.*

Mrectangle fills the given box with *pat. Mtriangle* fills the half-box with orientation *d* in the given boundaries. Right corresponds to a triangle in the lower right half of the box. Up corresponds to a triangle in the upper right, left is upper left, and down is lower left.

Msegment places an instance of the segment *sname* within the given boundaries. The segment is oriented according to *d,* where 'r' is null rotation. If either *thick* or *color* is nonzero, its value will replace corresponding values in the segment primitives. (For area filling, thick changes the fill pattern.)

Msetpat maps *pat* to *pattern. Pattern* is a string of the form "Pn" where n is a number between 0 and 11.

Mopenseg opens the segment named *sname.* All graphics calls up to a matching call to *mcloseseg* are stored under *sname.* An instance of the segment is obtained with a call to *msegment.* Segments can be nested to any level, and redefining segments is allowed. Beware of calls to *mtext* within a segment, since text will not rotate or scale.

Minclude includes the graphics metafile *fname* in the output stream. *Mendpage* advances to the next screen or page. On a terminal, the bell rings and a line is read to prevent premature erasure. *Mdone* completes metafile output, and is the only required call.

DIAGNOSTICS

None.

SEE ALSO

t4014(1), mx80(1), impress(1), primout(3), metafile(5)