

## UGH Commands and Utilities

VP translates high level commands and data (.exg files) into two-dimensional graphics primitives known as UGH (Universal Graphics Handler) commands. These primitives, in turn, are processed for display and hard copy. Such a separation eases the task of providing graphics on different hardware devices and provides relatively compressed graphics files (.ugh(bin)) for transmission between systems. A similar approach is adopted by many document preparation systems (such as  $\text{\TeX}$  or  $\text{\LaTeX}$ ) where the user input (.tex files) produces hardware independent (.dvi) files that may then be displayed or further converted for printout.

The UGH commands were designed specifically for two-dimensional scientific and technical graphics. Their structure is based on early PC implementations with 2-byte integers that allows a coordinate system with  $2^{16}$  steps in x and y (more than enough resolution even for 1600 dpi printers).

The following sections provide a detailed description of the UGH commands and the common utilities for printout and display. The utilities convert your VP graphics to other common formats for inclusion in word processing documents or for printing.

### 7.1 UGH Commands

In the following summary all numeric values are assumed to be 2-byte integer words (low byte first) unless followed by (b) to indicate a single byte (0–255). Names of numeric values are given in typewriter font. x,y coordinates are given in UGH user units as defined by the current MAP and WINDOW settings with 0,0 as the lower left corner. The description of each command begins with a header of the form:

# data	COMMAND_NAME
--------	--------------

This header indicates the command number (#), the allowed data, and a convenient **COMMAND\_NAME**. Each UGH command may return up to six 2-byte integers named after the PC general registers: AX BX CX DX DI SI.

The data in .ugh(bin) binary files consist of a sequence of UGH commands. Each command consist of 2 or 3 fields (total length NBYTE+3 bytes) in the form:

#### command number (one-byte)

**NBYTE (two-bytes)** Integer bytes of command data to follow.

**data (NBYTE-bytes)** The actual binary data (when NBYTE > 0).

0 data	COMMENT
--------	---------

NOP. The data portion can be used to imbed comment text in the .ugh(bin) file.

1 value	HOT_KEY
---------	---------

**Single-screen DOS only.** If value is specified, defines new screen swap "hot key" (scan code = high byte, char code = low byte) and activates "hot key" for single screen text/graphics swaps. If value is omitted, returns and deactivates AX = current "hot-key." The DOS single-screen default "hot key" is 0x2207 for Ctrl-G.

2 page mode	INITG
-------------	-------

Initializes graphic screen and sets full window and one-to-one mapping. If no page is specified, 0 will be used. Does not erase screen. The optional mode is a BIOS-specific screen mode.

3 mode	INITTEXT
--------	----------

**Single-screen DOS only.** Places screen in text mode. INITG or any UGH command  $\geq 10$  will return the screen to graphics mode. The optional mode is a BIOS-specific screen mode.

4 flag nscreen	DEBUG
----------------	-------

On entry, sets UGH internal debug flag (2 = print data on PC screen before command execution) and new nscreen. Returns the following data:

AX = device id:

1=YODA, 2=CGA, 3=EGA/ECD, 4=PGA, 5=PCGA, 6=NINE-32, 7=NINE-1K; 8=EGA/mono; 9=VGA(640x480); 10=Lap-Top; 11=8514/A; 12=IMAGRAPHS; 13=PS2A VGA in XT/AT; 13=VGA as 320 by 200 8-bit color, 14=XGA single screen 15=SVGA VESA, 18=OS2/2 32-bit PM, 20=X-windows driver

BX = mobile mode:  
1 if mobile mode allowed

CX = version number

DX = modification

DI = mouse available

SI = number of screens

5	FLUSH
---	-------

Flush any internal buffers to make sure that all current image data has been displayed.

6 text	ASCII
--------	-------

Writes data as ASCII text to screen and waits for a single-character reply that will returned in AX. If the last character of the text (as indicated by the NBYTE field) is 0, returns immediately without waiting for a key.

7 x0 y0 nx ny xppi yppi	SCREEN
-------------------------	--------

Defines a usable area on the graphics screen of size nx by ny, lower left corner at x0,y0, for the current UGH window number. If NBYTE > 11, also sets xppi and yppi (the effective pixel sizes in X and Y). If NBYTE < 8, the current settings are returned as AX=nx, BX=nt, CX=x0, DX=y0, DI=xppi, SI=yppi.

8 x0 y0 nx ny rot(b)	MAP
----------------------	-----

Defines a mapping of user coordinates onto the current SCREEN window: x0,y0 maps to the lower left, while x0+nx-1,y0+ny-1 maps to upper right. If rot=1, then x,y values will rotated by 90 degrees. If NBYTE < 8, the current settings are returned as AX=nx, BX=nt, CX=x0, DX=y0, DI=rot.

9 wnum	WINDOW
--------	--------

Places all following graphics in specified window. 10 windows are allowed. Their default definitions are

0 full screen	4 upper left 1/4
1 left half	5 upper right 1/4
2 right half	6 lower left 1/4
3 center 1/4	7 lower right 1/4
8 upper half	9 lower half

Each window number has its own SCREEN and MAP coordinates that may be changed with command 7 and 8. If wnum is not given, current window returned in AX.

10 color	ERASE
----------	-------

Fills current WINDOW with specified color and sets image start to lower left corner (see UGH command 25).

12 filespec	FILESAVE
-------------	----------

Starts a binary save of all following UGH commands on the specified file. If filespec is omitted, then the current FILESAVE is stopped. Not available in all implementations.

14 color red(b) green(b) blue(b) ...	SETVLT
--------------------------------------	--------

Defines the color mapping (Video Lookup Table) beginning with color to the following RGB triplets (each 0 - 255). (NBYTE-2)/3 colors will be specified. For full-color implementations the default colors are:

0 BLACK	8 brown
1 bright BLUE	9 dark blue
2 bright GREEN	10 dark green
3 bright CYAN	11 light green
4 bright RED	12 orange
5 bright MAGENTA	13 dark purple
6 bright YELLOW	14 dark yellow
7 bright WHITE	15 gray
16 - 255 bright white	

Most UGH drivers attempt to mimic this in default mode as much as possible.

15 color	GETVLT
----------	--------

Returns the current red green blue values for color in AX BX CX.

16 x y	GETPIXEL
--------	----------

Places pixel value corresponding to MAP coordinate x,y in SI.

17 x y step	CURSOR
-------------	--------

Displays graphic cursor starting at position x,y in MAP coordinates (or at last position). If step is given after or instead of x y, set step size for keyboard-controlled cursor movement. If step < 0, use low byte of step as cursor XOR in this and all following graphical input commands. When a key or mouse button is hit, returns:

AX BX	position in MAP coordinates
CX DX	position in screen coordinates
DI	character entered
SI	pixel value

18 x1 y1 x2 y2 step	RUBLIN
---------------------	--------

With no supplied data, display a graphic cursor. The first character or mouse button positions one end point of the line. The other end is positioned with the second character or mouse button. "Q" or CENTER mouse button aborts input. If just x1,x2 are given, they specify the starting point of the cursor. If x1,y1 x2,y2 are given, x1,y1 is the initial fixed point while x2,y2 are the initial mobile end. In most implementations the RIGHT mouse button swaps mobile and fixed ends and center motion. If step is given, set step size for keyboard-controlled cursor movement. If step < 0 use low byte of step as cursor XOR in this and all following graphical input commands. If step = 0, begin with center motion. Returns

AX BX	1st endpoint in MAP coordinates
CX DX	2nd endpoint in MAP coordinates
DI	1st character
SI	2nd character

19 x1 y1 x2 y2 step	RUBBOX
---------------------	--------

With no supplied data, displays a graphic cursor. The first character or mouse button positions one corner of the box. The other end is positioned with the second character or mouse button. "Q" or CEN-

TER mouse button aborts input. If just  $x1, x2$  are given, they specify the starting point of the cursor. If  $x1, y1$   $x2, y2$  are given,  $x1, y1$  is the initial fixed point while  $x2, y2$  are the initial mobile end. In most implementations the RIGHT mouse button swaps mobile and fixed ends and center motion. If  $step$  is given, set step size for keyboard controlled cursor movement. If  $step < 0$ , use low byte of  $step$  as cursor XOR in this and all following graphical input commands. If  $step=0$  begin with center motion. Returns

AX BX 1st corner in MAP coordinates.  
 CX DX 2nd corner in MAP coordinates.  
 DI 1st character  
 SI 2nd character.

For the following geometric elements,  $width \leq 0$  means the element will be filled or solid. Coordinates are given in MAP units.

20 color width  $x1$   $y1$   $x2$   $y2$   $x3$   $y3$  TRI

A triangle with specified corners.

21 color width  $x1$   $y1$   $x2$   $y2$   $x3$   $y3$   $x4$   $y4$  TRAP

A 4-sided trapezoid with specified corners.

22 color width  $nx$   $ny$   $x0$   $y0$  BOX

A box of size  $nx$   $ny$  relative to a corner at  $x0$   $y0$ .  $nx$   $ny$  may be negative.

23 color width  $x1$   $y1$   $x2$   $y2$  ... POLYGON

A polygon with  $(NBYTE-4)/2$  vertices. Most implementations are limited to a maximum of 256 vertices.

24  $nx$   $ny$  bytes-per-pixel IMAGSIZE

Sets image parameters for following data but generates no graphics. Used to imbed parameters in .ugh(bin) file.

25  $x11$   $y11$  IMAGXY

Sets image's lower left corner position in MAP units.

26  $xps$   $yps$  IMAGPS

Sets image pixel size on display screens in terms of addressable pixels.

27 bank(b)  $x0$   $y0$  color1(b) color2(b) ... PXDATA

Passes one scan line of (NBYTE-5) pixels to the specified color bank starting at  $x11+x0, y11+y0$ . The value of bank is ignored on unsupported devices.

28 bank(b)  $x0$   $y0$  run1(b) color1(b) ... RLDATA

Passes one scan line of run-length encoded data to the specified color bank starting at  $x11+x0, y11+y0$ . The value of bank is ignored on unsupported devices. Run-length data in the form run(0-255), color(0-255).

29 bank(b) SETBANK

Sets image bank for following one byte/pixel images on supported display devices: 0 = all, 1 = red, 2 = green, 3 = blue. *zero.*

30 color width  $rx$   $ry$   $x0$   $y0$   $\theta_1$   $\theta_2$  0 ELLIPSE

An ellipse of  $x, y$  radii  $rx$   $ry$  centered at  $x0$   $y0$ .

31 width MAXWIDTH

Sets the maximum line width for display devices that support variable-width lines. Setting to 0 gives fastest possible line drawing.

32 color width  $x1$   $y1$   $x2$   $y2$  LINE

Draws a single line from  $x1, y1$  to  $x2, y2$ . When possible line type 32 is drawn with rounded ends.

33 color radius  $x1$   $y1$   $x2$   $y2$  ... POINTS

Filled circles of radius centered at  $x0$   $y0$   $x1$   $y1$  .... Some implementations set only single pixels, and most have a maximum of 256 vertices.



34 color width x1 y1 x2 y2

LINE

Draws a single line from x1,y1 to x2,y2. When possible, line type 34 is drawn with square ends extended width/2 beyond the end points.

35 color width x1 y1 x2 y2 ...

VECTOR

A vector connecting (NBYTE-4)/2 vertices. Most implementations are limited to a maximum of 256 vertices.

36 x0 y0 count offset segment

GETSCAN

DOS only. Returns a horizontal scanline of count pixels beginning at SCREEN (not MAP) address x0,y0 to the PC address given by segment:offset. If count < 0, then return -count bytes in the -Y (down) direction starting x0,y0.

37 x1 y1 x2 y2 orient filespec

IMAGINC

Processes the UGH image data in filespec to fill the box defined by the corners x1,y1 and x2,y2 in MAP units. The orient parameter is presently unused. Supported by VP, ughdev, and hardcopy utilities but not by individual DOS screen drivers.

38 fontnum

HFONTD

Sets the default Hershey font number. If fontnum is not given, returns current font number in AX and extra spacing in BX. If fontnum ≤ 0, -fontnum is taken as extra space between all characters (16 units = normal width). A new spacing stays in effect for all fonts. The default spacing of 0 is reset by INITG. Often 1 will give more readable text.

39 col wid len h font x0 y0 lxp(b) lyp(b) text

HWRITE

Writes text on the graphics screen, where

```
col    text color
wid    width of strokes
len    if >0 gives total length, if <0 character width
h      if >0 gives total height, if <0 character height
font   Stroke font number, <0 for rotation by 90 degrees
x0 y0  starting position
```

```
lxp    X justification 0=left 1=center 2=right
lyp    Y justification 0=bottom 1=center 2=top
text   the NBYTE-18 characters to draw
```

Other values for lxp lyp define a rotation of the text by an angle theta in the counterclockwise direction about the starting point x0,y0, where  $\text{Cos}(\theta) = 100 * \text{lxp}$  and  $\text{Sin}(\theta) = 100 * \text{lyp}$ .

HWRITE supports subscripts, superscripts, and Greek characters in several font styles:

- Subscript characters are enclosed inside single quotes '.
- Superscript characters are enclosed inside double quotes ".
- Arbscript characters are enclosed inside back quotes '. Arbscripts are like sub- and superscripts but with complete control over size and baseline offset. They can be used to provide subscripts within superscripts or just different size characters within a text string. The default arbscripts are the same size as sub- and superscripts (75 percent of normal) but with no baseline offset. To change the characteristics of all following arbscripts, begin an arbscript with the sequence ~siz+off+dx~ or ~siz-off-dx~, where size is the arbscript size in 32nds of font size and +off or -off is the baseline shift in 32nds of arbscript size. The third numeric value, +dx or -dx, gives an X shift in 32nds of the font X size for this arbscript only. The +-dx may be omitted. For example, to set all following arbscripts to act like superscripts, include '~24+18~' and to have arbscripts imitate subscripts, include '~24-10~'. To use arbscripts as subscripts within superscripts try '~16+10~'. If a ~ does not immediately follow the ' it indicates a color+font change. The ~siz+off~ specification can also be used immediately after a ' or " to change the characteristics of all following sub- and/or superscripts. **Note:** The first occurrence of ' " or ' always activates sub-, super-, or arbscripts even if you were already in one of the other scripts.
- Greek characters are enclosed within exclamation points !. A Greek period (!.) gives a half-space.
- A color+font+size+offset change is specified inside ~. Thus, ~23~ will use color 23 for the remainder of the text, while ~23+24~ will use color 23 and change to font 24 for the remainder. ~23+24+64-10~ will change to color 23, font 24, double size (32=normal) lowered 10/32 of total size below baseline. A color change to -13 (~-13~) will stop all following UGH special character translation for this command 39. Note that this use of ~ is different from a ~ immediately following a ' arbscript indicator.

- The @ character acts like a carriage return-linefeed combination.
- The # character gives a backspace and allows characters to easily be written on top of each other.

To include any of these special delimiters in a text string, repeat the character twice. Thus, "" will produce a single " in the output. Many mathematics symbols are included in the Greek character set. Both underlining and boxing are included as part of the UGH handler.

- `_~col+wid+bot~text_` will underline text with a line of color `col`, width `wid`, a distance  $(\text{font-size} \times \text{bot}) / 32$  from the baseline:  
e.g., `_~4+10~10~this text will be underlined in color 4_`
- `|~col+wid+bot+top~text|` will draw a box around text with a line of color `col`, width `wid`, a distance  $(\text{font-size} \times \text{bot}) / 32$  and  $(\text{font-size} \times \text{top}) / 32$  from the baseline. If `wid < 0` the box will be filled:  
e.g., `|~4+10~10+34~this text will be boxed in color 4|`

If `_` or `|` is not immediately followed by the underscore/box `~...~` parameter list, the normal `_` and `|` characters will be used.

40

EXIT

Stop the UGH driver. With DOS resident extensions, attempt to return storage to DOS.

41 data

GESC

All following data are passed to the current device without translation. Actual implementation will depend on the specific UGH device driver. For DOS UGH8514, a data byte of 1 sets a form of mobile mode. The following graphics entries (until cursor input) are drawn off screen and then attached to the cursor. The actual off-screen area is 512 by 256 pixels. For DOS UGH8514, GESC 255(byte) MIX(byte) Xsize Ysize Xsour Ysour Xdest Ydest, specifies a BITBLT move in 8514A coordinates.

Special GESC one-byte codes for X-windows:

- 0 stop mobile mode (also ended by cursor input)
- 1 start mobile mode
- 3 start immediate displays (no pixmap saves)
- 4 start pixmap displays
- 5 raise UGH graphics window
- 6 change UGH window name (text follows)

- 7 change UGH icon name (text follows)
- 8 iconify UGH window
- 9 de-iconify UGH window

Special GESC one-byte codes for OS2 Presentation Manager:

- 0 stop mobile mode (also ended by cursor input)
- 1 start mobile mode
- 6 change UGH window name (text follows)
- 8 minimize UGH window
- 9 un-minimize UGH window
- 10 maximize UGH window

## 7.2 ughb2d

<code>ughb2d fin ( option1 option2 ...</code>	or
<code>ughb2d fin -option1 -option2 ...</code>	or
<code>ughb2d -option1 -option2 ... fin</code>	

**ughb2d** is a filter program to convert a binary `.ugh(bin)` file to `.dec` notation in `stdout`. The output may be redirected with `>fout.dec` on the command line. This command is used primarily for debugging and examining the detailed contents of `.ugh(bin)` files.

The input file **fin** is required. If no extension is given, `.ugh(bin)` will be appended. If no **fin** or options are given, a usage summary will be displayed.

The allowed options are (case insensitive, unique abbreviations allowed)

- min n** skip first `n` UGH commands in input file.
- max n** stop after a total of `n` UGH input commands.
- comlim minc maxc** only output UGH commands  $\text{minc} \leq \text{com} \leq \text{maxc}$ .
- width n** integer output with minimum width `n` (default 4).
- msg fname** place message output in `fname` instead of console.
- labels** only dump labels (ugh command 0).
- hex** all output in hexadecimal.

**Examples**

```
ughb2d vossplot           ; display contents of vossplot.ugh
ughb2d vossplot -max 4    ; display first 4 commands
ughb2d vossplot -com 39 39 ; only display HWRITE commands
```

**7.3  
ughd2b**

```
ughd2b fin [fout] ( option          or
  ughd2b fin [fout] -option         or
  ughd2b -option fin [fout]
```

**ughd2b** is a filter program (the inverse of **ughb2d**) to convert a decimal .dec file to .ugh(bin) binary according to the following conditions:

- Lines beginning with >> are continuations of command data.
- Lines beginning with .. are treated as ASCII text. Trailing blanks are normally stripped from the text. To include trailing blanks, end the line with a pair of matching periods (..).
- Lines not beginning with a number are treated as comments.
- Portions of a numeric line following a ; are ignored as comments.

The input file **fin** is required. If the output file **fout** is not given, fin.ugh(bin) will be used for output. If no extension is given, .dec will be appended to **fin** and .ugh(bin) to **fout**. If no files or options are given, a usage summary will be displayed.

The only allowed option is (case insensitive, unique abbreviations allowed)

**replace**          allow an existing .ugh(bin) file to be overwritten.

**Examples**

```
ughb2d vossplot -rep      ; convert vossplot.dev to vossplot.ugh
ughd2b default newcol    ; convert default.dec to newcol.ugh
```

**7.4  
ughplot**

```
ughplot fn1 fn2 ... ( option1 option2 ...
```

**ughplot.com** is a DOS-only program to convert .ugh(bin) files to GL (plotter graphic control language) and directly drive a pen plotter connected to a PC via a serial or GPIB interface.

**ughplot.com** should work with almost any pen plotter. It does not require information back from the plotter, and it uses only a minimal subset of GL. The user, however, will be required to specify information about each particular plotter to the program. This information is typically placed in the special file ughplot.pro that determines the details of conversion.

**fn1 fn2 ...** specify the input data and profile files to the program. The files are assumed to contain UGH commands and, if necessary, the extension .ugh is appended. If a file is specified as name.pro, then it is read as an alternate profile, allowing a temporary change of plotter parameters or the inclusion of other GL commands.

Since **ughplot.com** is available only with PC DOS, a detailed discussion of the many **ughplot** options and the format of the ughplot.pro file are included only with the DOS software package in the file ughplot.ref and in the default ughplot.pro.

**7.5  
u-ps**

```
u-ps fn1 fn2 ... ( option1 option2 ...          or
  u-ps fn1 -option1 -fn2 -option2 ...
```

**u-ps** converts binary .ugh(bin) to PostScript (.ps) or embedded PostScript (.eps) graphics for high-quality output or use in text documents. The default output is placed in fn1.ps. The **stdout** option allows the output to be redirected or piped. The options in the files u-ps.pro and u-ps-usr.pro are read before the command line is processed.

The options (with defaults) are listed here and are discussed in more detail in the following sections.

exact (default)	fraction f (1.0)	output name.ext
center	rotate (optional deg)	stdout
best f	antirotate	replace
fill f	ppi dot-in-X dot-in-Y (800 800)	append
bw (default)	gray color-num fraction	power pow (1.0)
wb	rgb color-num R G B	reverse
shade	vltimage name.ext	ehandler
colors	min ugh-com-number (0)	hershey



for 'name'	max ugh-com-number (100)	bare
label col pntsiz (7 8)	com first last (1 100000)	noprologue
page Xin Yin (8.5 11)	margins LRin TBin col (0.25 0.5 1)	msg name.ext
gsize GXin GYin (8 10)	font num PSname how smul xmul ymul	
substitute 'old' 'new'	greek char trans-char font-num	multipage
screen cells-per-inch	cmap name.ext	botoptions
ugh num capjoin wmult	eps	4bits
copies n	background col	slidecolors

### 7.5.1 General considerations

**u-ps** converts binary .ugh(bin) files to a standard PostScript text file. PostScript fonts may be substituted for all or some of the UGH Hershey stroke fonts. Colors, shaded halftones, and images are supported. The output .ps or .eps are ordinary text files that may be edited with any text editor. They begin with a prologue that includes the actual **u-ps** command that produced the file and a copy of a set of PostScript procedures (from the file u-ps-pro.ps) that implement many UGH commands in PostScript.

**u-ps** should produce the same output independent of the operating system. It supports a wide variety of customization for PostScript printing of .ugh(bin) binary files through \*.pro profile files and command line options.

Upon startup, **u-ps** first reads the options in u-ps.pro. We suggest that users make no modifications to this file. **u-ps** then reads any options in **u-ps-usr.pro** if it exists. Place any "permanent" options in this file.

Finally, **u-ps** checks the command line for filenames and additional options. Filenames should be given in the form name.ext. When no ext is given, .ugh(bin) is assumed. Additional profiles may be specified on the command line as name.pro. Command line .pro option files are evaluated *after* any command line options and interspersed with the .ugh(bin) files. They can be used to change some PostScript conversion options (like **font** or Greek translation) midway through a conversion.

Command line option fields may begin individually with "-" or collectively with "(" (on UNIX systems it may be necessary to use "\\"). They are evaluated *after* the u-ps.pro and u-ps-usr.pro profiles but before any .ugh(bin) processing.

Command line options and specifications in .pro files may be abbreviated by their initial letters. Only enough letters to distinguish an option from all other options need be specified. Thus, -ce, -cent, and -center all indicate the **CENTER** option.

**u-ps** produces pages of PostScript graphics in the default size of 8.5 inches by 11 inches. This size may be altered with the **PAGE** option. Margins limit the graphics to the center of the page. The default margins are 1/4 inch from the left and right edges and 1/2 inch from the top and bottom, which give an 8-inch by 10-inch graphics area. The margin sizes

### 7.5.2 Output page format

may be changed with the **MARGIN** option (which resets the graphics size to  $GX = PageX - 2 * MarginX$ ,  $GY = PageY - 2 * MarginY$ ). The graphics size may be altered independently of the margins with the **Gsize** option.

Default processing adds 1/4-inch corner tics to indicate the graphics area on the output page and includes a text label above the top graphics margin. This label consists of

system u-ps fns options	name	date time
-------------------------	------	-----------

If **BOTOPTIONS** is specified, the u-ps fns options is moved below the lower left corner tic mark.

The **FOR** option specifies the center name for identifying printer output.

Both the corner tics and the top text label can be omitted with the **BARE** option.

The UGH graphics elements are placed inside the graphics margins. Default output places the UGH origin 0,0 at the lower left corner tic and preserves **EXACT** scaling of 800 points/inch. The following options can be used to change the scaling and placement:

<b>center</b>	Center all UGH elements in page.
<b>fill f</b>	Change scaling to center and fill up to a fraction <i>f</i> of the graphics size.
<b>best f</b>	Same as <b>FILL</b> with optional rotation.
<b>fraction f</b>	Change <b>FILL</b> fraction from default of 1.0.
<b>rotate</b>	Force 90 degree rotation.
<b>rotate 0</b>	Force no rotation.
<b>rotate 45</b>	45 degree rotation.
<b>antirotate</b>	Force -90 degree rotation.
<b>ppi 1600 1600</b>	Reduce size to 50% of standard (by increasing the effective UGH units per inch).
<b>gsize 6 8</b>	Use a reduced graphics size inside margins (here 6 by 8 inches).

Default processing produces a single page of output as the union of all command line .ugh(bin) files. In many cases, however, it is desirable to collect multiple graphics pages into the same PostScript file. This can be accomplished in two ways:

1. The **append** option will append PostScript output to the existing output file without including another set of procedures. Each page may use different options as, such as



```
u-ps demovp (fill shade replace
u-ps greek (best append
```

2. The **multipage** option creates a separate output page for each .ugh(bin) file named on the command line. Each file is processed with exactly the same options. This option also sets **botoptions** to place the command line at the bottom and the current filename at the top of each page.

```
u-ps demovp greek (fill multipage replace
```

### 7.5.3 Colors and shading

Default output is mapped to black and white with UGH color 0 (the background) as white and all other colors as black. The **wb** option reverses this option to produce a negative image. Even with default **bw** processing, included images are processed as shaded halftones.

The **shade** option specifies that different UGH colors correspond to different halftone densities. The default u-ps.pro sets up 16 sample shades with the **gray** option that are used when **shade** is specified. The actual halftone screen size may be changed with the **screen** option, but the results depend on the output device.

The **color** option specifies that different UGH colors correspond to different RGB intensities. The default u-ps.pro sets up 16 sample RGB values with the **rgb** option for the default UGH colors. These colors should be preserved on PostScript printers that process color. Other printers will probably approximate the colors with gray shades. The colors may also be changed by reading a standard UGH colormap file (command 14) with the **cmap** option or by including the filename at the start of the list of input files. The **color** option sets color 0 to white and color 7 to black.

Color 0 is the traditional background color (on UGH monitors the default is black; on hard copy the default is white). To remain consistent while allowing users to change colormaps, a **color** specification with color 0 as RGB 0 0 0 is translated to 255 255 255 for output. To force a black background, use color 0 as RGB 0 0 1.

The **slidecolors** option sets the same default colors as **color** but does not reverse black and white. Color 0, the background, is black and color 7, the normal text color, is white.

### 7.5.4 Encapsulated PostScript files

When the **eps** option is specified, the output file attempts to conform to the Adobe Encapsulated PostScript format for including graphics in other documents. The output file will use a default extension of .eps instead of .ps. No bitmapped screen preview is included and no showpage operator is included at the end. The **eps** option is incompatible with the **multipage**, **noprologue**, and **ehandler** options. It is recommended that the **exact bare** option be used along with **eps** to prevent the margin tics and header from appearing in the output and to prevent

any additional rescaling by **u-ps**. Using **fill** or **best** with **eps bare** may cause the BoundingBox specifications to be incorrect.

The bounding box is an estimate of the extent of the graphics based on Hershey font character sizes (not on their eventual PostScript translation) and does not include line widths. To allow "extra space" around the bounding box, you can also specify the **fraction** option as with **fill** or **best**. Thus, **eps exact bare fraction 0.9** will give an .eps file with a bounding box 10% larger (5% in each direction) than the actual graphics.

### Examples

```
u-ps vossplot -best -replace
```

Options are read from u-ps.pro and then from u-ps-usr.pro. The UGH elements in vossplot.ugh are scaled (and possibly rotated) to fill the available page, and the output is placed in vossplot.ps.

```
u-ps f1 spec.pro f2 (bare stdout > lpt1
```

Options are read from u-ps.pro and then from u-ps-usr.pro. The UGH elements in f1.ugh(bin) are translated with exact sizing. spec.pro changes some of the font specifications, and f2.ugh(bin) is processed with exact scaling. The **BARE** option eliminates any header information and margin markings. The output is sent directly to an attached PC PostScript printer (via redirected stdout) without creating a file.

```
u-ps f1 f2 f3 -fill -fract 0.8 -std | lpr -Pps
```

Options are read from u-ps.pro and then from u-ps-usr.pro. The UGH elements in f1.ugh(bin), f2.ugh(bin), and f3.ugh(bin) fill the center 80% of the available page and are piped directly to the UNIX line printer queue for PostScript.

### 7.5.5 Font customization

With the **u-ps** customization features, output can be tailored for different printers without changing the input .ugh(bin) file. Examples of customization are given in the default u-ps.pro file.

The line end types and joins (0 = but, 1 = round, 2 = square) can be altered for individual UGH element types. In addition, the UGH **com** option can specify a line width multiplier for individual element types.

Each Hershey font number may correspond to a different PostScript font or may use the original Hershey stroke fonts. Special effects are possible with the PostScript fonts by specifying the **how** field in the **font** option:

```
font num PSname how smul xmul ymul ; where how =
0 for normal (solid) PostScript characters.
>0 for outline characters with width = how MOD 100
and color = how/100.
    if color>0 characters filled with normal font color
    and outlined with width (how MOD 100)
    in color (how/100).
    if color=0 characters not filled, but
    outlined with width (how MOD 100)
    in normal font color.
examples:
font 24 Times-Bold 3 1.29 1.15 1.00
; hollow outline width 3 color=0 gives normal color.
font 24 Times-Bold 701 1.29 1.15 1.00
; filled font default color, outline width 1 in color 7
```

Since the programs that generate the .ugh(bin) binary files (such as **VP**) only have access to the Hershey stroke fonts, output that depends on exact character placement (such as **VP** symbols) may require adjustments (increases or decreases) of the PostScript character sizes using the **font** option. With the **font** option, the PostScript character size may be increased or decreased.

Option **font 33** provides control over how **u-ps** interprets Greek characters. The default specified in **u-ps.pro** uses the PostScript symbols font for most characters but keeps a few as Hershey stroke characters. In some cases, the Greek translation is changed to include PostScript symbols (such as copyright) not available in the Hershey fonts. Any of these translation options may be changed for special effects.

### 7.5.6 Special PostScript characters (German and French accents)

It is possible to modify the PostScript fonts to include special characters. The default **u-ps-pro.ps** now includes most standard French and German accents. The PostScript vector Extracodes (located near the end of the file) shows how the characters are encoded (see the PostScript Red Book, p. 255, for other possible characters). Once encoded, these characters may be accessed by including the PostScript octal escape sequences (\nnn) directly in the UGH labels or through the use of standard substitutions. Sample substitutions are include in **u-ps.pro** for most of the French and German accented characters. These substitutions are designed to pass transparently through **VP**, give a readable (but incorrect) graphic with the UGH Hershey characters, and provide the correct character with PostScript output. Sample substitutions from **u-ps.pro** are

```
A#o for angstrom sign (# = UGH backspace)
e#> for e acute
i#< for i grave
```

```
o#- for o circumflex
u#> for u umlaut
```

Thus, to include a German ü, from **VP** you could use one of the following:

```
comment X Y ; for default {\bf u-ps} substitution
u#>bung hilft

comment X Y ; for direct PostScript escape codes
\\366bung hilft ; double \ because VP special char

comment X Y asis ; for direct PostScript escape codes
\\366bung hilft ; ASIS eliminates need for double \\
```

The **u#>** substitution method is recommend because the Hershey fonts will give a better approximation of the exact size.

If an explicit extension (.ext) is not specified with the input file names, **u-ps** searches for defaults as follows:

```
PC DOS: fname.ugh (case insensitive)
current directory, then all subdirectories in PATH.
unix: fname.ugh or fname.ughbin (lower case)
current directory, then all subdirectories in
UGHBINPATH, then all subdirectories in PATH.
```

**u-ps** uses a similar strategy to find the Hershey font stroke data:

```
PC DOS: Hxx.FNT (case insensitive)
current directory, then all subdirectories in PATH.
unix: Hxx.FNT or Hxx.FNTBIN (upper case)
current directory, then all subdirectories in
FNTBINPATH, then all subdirectories in PATH.
```

### 7.5.7 Searching for files

### 7.5.8 Including images

UGH command 37 (image include) is supported using the standard PostScript image procedures to produce halftone approximations on black-and-white printers. The actual image data (from **fname.uim(bin)**) are included in HEX notation in the output file. This, of course, can result in *large* output files. If **4bits** is given as an option, pixel values are truncated to the range 0–15; otherwise the full 8 bits is used for values in the range 0–255. With **4bits** the files are half as large.

When **color** has been specified, the image values are taken as indices into the RGB colormap. If a **vlname** is given, it supplies the RGB triplets; otherwise the colors are taken from the current colormap as determined by the **cmap** or **rgb** option.

Similarly, a **.uim(bin)** image file may be printed by itself by including **name.uim(bin)** in the list of input files. Thus,

```
u-ps sample.uimbin (fill fraction 0.6
```



### 7.5.9 PC considerations

will print a default halftone view of the UGH image in `sample.ugh(bin)` (color 0 = black, color 255 = white) in the center 60% of the page.

On PCs, `u-ps` can be used as a filter to directly process UGH files for an attached PostScript printer without the need to create an intermediate file. Thus,

```
u-ps demovp (fill stdout > lpt1
```

will redirect the PostScript stdout output directly to the printer.

### 7.5.10 AIX/UNIX considerations

UNIX shells often object to the “( option” form of indicating options. In this case, either use the standard -option specification or precede the “(” with “\”:

```
u-ps demovp \ (fill
```

As with PS's, output may be redirected or piped directly to the printer queue:

```
u-ps demovp -fill -stdout | lpr -Pps
```

### 7.5.11 Detailed option specifications

<b>out <code>fname.ext</code></b>	Places PostScript output in <code>fname.ext</code> . If no <code>.ext</code> is given, <code>.ps</code> will be used. If the file already exists, it can be overwritten with REPLACE.
<b>replace</b>	Allows default output to <code>fn1.ps</code> to overwrite any existing file. If default output to <code>fn1.ps</code> is used and REPLACE is not given, an existing <code>fn1.ps</code> will not be overwritten.
<b>append</b>	Appends PostScript output to the existing output file and does not include a new set of procedures. If the output file does not exist, a new file will be created, but the initial procedures will not be included.
<b>multipage</b>	Generates a separate page of PostScript output for each <code>.ugh(bin)</code> file named on the command line.
<b>stdout</b>	Places PostScript in stdout. This output may be redirected on the command line or piped to another process such as the printer queue.
<b>copies <code>n</code></b>	Produces <code>n</code> copies of the page, where $n \leq 5$ . For more than 5 copies, edit the output <code>.ps</code> file to include the line <code>/#copies 9 def before showpage.</code>
<b>page <code>X Y</code></b>	Sets page's X and Y size in inches (default 8.5 11).

<b>margins <code>LR TB col</code></b>	Sets left-right top-bot margins in inches and margin option color <code>col</code> (0 omit, > 0 for 1/4 inch corner tics, < 0 for full lines).
<b>gsize <code>GX GY</code></b>	Redefines the allowed graphics area inside margins.
<b>fill <code>f</code></b>	Adjust scaling to fill to margins, optional fill fraction <code>f</code> .
<b>rotate</b>	Force 90 degree rotation on page.
<b>rotate <code>DEG</code></b>	Force rotation by <code>DEG</code> degrees.
<b>best <code>f</code></b>	Fill with optional rotation to best match graphics aspect ration with GSIZE, optional fill fraction <code>f</code> .
<b>fraction <code>f</code></b>	Fill center fraction <code>f</code> of graphics area (default 1.0).
<b>exact</b>	No scaling of UGH coordinates (default). This is useful to generate graphics having <i>exactly</i> the size specified in VP for overlaying graphs.
<b>center</b>	No scaling of UGH coordinates, but center output in page.
<b>nopro</b>	Do not include any PostScript prologue procedures.
<b>eps</b>	Output Encapsulated PostScript format.
<b>for "text"</b>	PostScript header For: field and top center label. If the text includes spaces, use quotes as delimiters.
<b>label <code>col pnt</code></b>	Page label color and optional point size.
<b>bare</b>	Omit page label and margin lines or corner tics.
<b>antirotate</b>	Force -90 degree rotation on page.
<b>min <code>n</code></b>	Begin output with UGH command <code>n</code> .
<b>max <code>n</code></b>	End output after UGH command <code>n</code> .
<b>comlim minc maxc</b>	Only output UGH commands $\text{minc} \leq \text{com} \leq \text{maxc}$ .
<b>ppi <code>ppix ppiy</code></b>	Sets points-per-inch resolution (default 800 800). This can be used to rescale the plot by a fixed amount in one or both directions. Thus, PPI 1600 1600 reduces the plot size by a factor of 2. PPI 1600 400 reduces the x-direction but increases the y-direction by 2.
<b>msg <code>fname.ext</code></b>	Output messages to specified file (default console).



**ehandler** Include `ehandler.ps` in prologue. This is useful as a trace when the printer produces no output.

**hershey** Force all fonts to use Hershey stroke characters.

**ugh com capjoin wmult**  
Specify capjoin and width multiplier for ugh element com (0=butt, 1=rounded, 2=extended square). For Hershey Labels capjoin = 10\*cap + join.

The following options (**font greek substitute**) allow customization of how UGH text information is translated into PostScript.

**font h PostScriptName how smult xmult ymult**  
Specify PostScript font info for Hershey font h. Use h=33 for Greek characters and symbols. Normal fonts use how=0. For an outline font use |how|>0 with width = how MOD 100. how/100 gives the outline color number. If color<0, only the outline will be drawn. smult xmult ymult are space charX charY multipliers for Hershey stroked characters. Examples:

```
font 12 Helvetica 0 1 1 1 ; normal Helvetica
font 12 Helvetica 0 1.3 1.2 1 ; twiddled Helvetica
font 25 Times-Italic -701 1.3 1.2 1 ; narrow outline, color 7
font 12 Hershey 1 1 1 1 ; Hershey (default)
```

**greek c gc gfont**

Specify character translation inside UGH Greek ! delimiters.

c character to be translated.

gc is the translation character(s) specified inside double quotes (up to 6 characters are allowed) or the numeric value (decimal or \$hex) of the single translated character.

gfont is the numeric specification of the font to be used. 0 = use base Hershey font. 33 = PostScript Symbol font. Other numbers are the fonts specified in the font option above.

If not explicitly specified, Greek characters are taken from the base Hershey font instead of a PostScript font. Examples:

```
greek "a" "a" 13 ; a = lowercase alpha from Symbols
greek "1" "1" 0 ; 1 = path integral from Hershey fonts
greek "I" 242 13 ; I = integral as char 242 from Symbols
```

**substitute old new**

Specify substitution text inside UGH labels. Typically this is used to replace characteristic UGH sequences with *exact* PostScript characters. Up to 50 old new pairs (with a maximum of eight characters/string) may be specified. Examples:

```
sub C#!@@! !""! ; copyright symbol
sub u#: \366 ; German u umlaut
```

The following options allow customizing how PostScript interprets UGH color numbers. All **gray** and **rgb** colors used must be explicitly specified. Color numbers greater than the largest specified number with be set to the largest specification.

**bw** Use black-white coloring (default). Color 0 prints white, all other colors give black.

**wb** Use white-black coloring. Color 0 prints black, others white.

**shade** Use shaded halftone gray levels. See **gray** below.

**colors** Use **rgb** colors. Set standard VP colormap with background (color 0) as white and normal text (color 7) as black. See **rgb** below.

**slidecolors** Use **rgb** colors. Set standard VP colormap with background (color 0) as black and normal text (color 7) as white. See **rgb** below.

**background col** Sets background to specified color number.

**gray num fract** Specify gray level fraction (fract=0 for white, fract=1 for black) for color num.

**screen cells-per-inch** Changes the PostScript halftone screen resolution. Fewer cells-per-inch give more gray scale resolution but coarse shading. The results are device-dependent. This command only changes the screen size, leaving the angle and actual procedure as the PostScript device defaults. For more detailed halftone customization, modify the included PostScript procedures.

**rgb num r g b** Specify RGB (0-255) intensities for color num.

**cmap name** Use the colormap in `name.ugh(bin)` to determine **rgb** colors and force **colors** as the output type.

**Note:** A standard UGH colormap (command 14) will also set the **rgb** colors for the following files when it is given as an input filename. Thus, either of the commands

```
u-ps demovp (fill fract 0.9 cmap mycolors
u-ps mycolors demovp (out demovp fill fract 0.9 color
```

will use the colors in `mycolors.ugh(bin)` for the graphics elements in `demovp.ugh(bin)` to produce `demovp.ps`.

The following options allow customization of halftone image rendering for the UGH command 37. In normal operation, image byte 255 = white and 0 = black.

**reverse** Reverse byte value mapping so 0 = white, 255 = black.

**power pow** Set image scaling `pow>1` for increased, `pow<1` for decreased contrast (default 1.0).

**vlimage name** Use the colormap in `name.ugh(bin)` to determine brightness (0.0–1.0) as

$$\text{brite}[k] = (R[k] + G[k] + B[k]) / 765$$

of color `k` for shaded images or as specifying the **rgb** triplets for **color**.

#### required files

In addition to the executable program and user-specified `.ugh(bin)` files, **u-ps** searches for the following files:

**u-ps.pro** supplied file for standard colors, font specifications, and special character translations. Do not modify this file.

**u-ps-usr.pro** optional `.pro` file where the user should place any desired modifications to `u-ps.pro`. This file is read after `u-ps.pro`.

**u-ps.ref** short reference list of the available options.

**u-ps-pro.ps** the **u-ps** prologue of special UGH PostScript procedures.

**ehandler.ps** standard PostScript error-handling procedures.

## 7.6 u-cgm

<code>u-cgm fin ( option1 option2 ...</code>	<b>or</b>
<code>u-cgm fin -option1 -option2 ...</code>	<b>or</b>
<code>u-cgm -option1 -option2 fin</code>	

**u-cgm** converts a binary `.ugh(bin)` file to CGM (Computer Graphics Metafile) format in `fin.cgm(bin)`. **u-cgm** is a good choice for including graphics in many pc-based word processors such as WordPerfect or Microsoft Word. **u-cgm** has only limited text capabilities, and these are system-dependent. Therefore, for all but the most simple labels you will probably need the default **hershhey** option to convert the UGH text to stroked lines. By default many word processors convert colors to halftone shades for printing. You may want to force output using a `cmap wb` option.

The allowed options are (case insensitive, unique abbreviations allowed)

**hershhey** Stroke all characters as UGH Hershey fonts (default).  
**nohershhey** Use CGM default text (no subscripts, superscripts or Greek).

**replace** Replace an existing `fin.cgm(bin)` file.

**out fname** Place output in `fname.cgm(bin)`.

**fraction f** Set output size so graphics occupy the center fraction `f`.

**page Xin Yin ppi** Set output limits corresponding to a `Xin` by `Yin` page (inches) with optional `ppi` (default 800).

**center** When used with **PAGE**, forces centered output.

**min n** Skip first `n` UGH commands in input file.

**max n** Stop after a total of `n` UGH input commands.

**comlim minc maxc** Only output UGH commands `minc ≤ com ≤ maxc`.

**cmap fname** Include colors from `fname.ugh(bin)`. Try `cmap wb` to force black graphics on white background.

**msg fname** Place message output in `fname` instead of console.

**Examples**

```
u-cgm vossplot -hersh -rep -cmap default
```

```
u-cgm vossplot ( hersh rep cmap wb page 6 4 center
```

**7.7****u-uim**

u-uim fn1 fn2 ... ( option1 option2 ...	or
u-uim fn1 fn2 ... -option1 -option2 ...	or
u-uim -option1 -option2 ... fn1 fn2 ...	

**u-uim** converts .ugh(bin) files into various types of pixel image files:

- UGH images in **fn1.uim(bin)** (using UGH commands 27 and 28) with 256 colors (8-bit).
- HP LaserJet printer compatible images in **fn1.pcl(bin)** at 300 dpi on a graphics page of 8 by 10 inches with 2 colors.
- Windows-compatible bitmaps in **fn1.bmp(bin)** with 2, 16, or 256 colors (1, 4 or 16 bits-per-pixel).
- CompuServe GIF-compatible compressed images in **fn1.gif** with 256 colors.

A number entered as a **fn** specifies a new UGH window for the following **fn**. The allowed options are (case insensitive, unique abbreviations allowed)

**hp xi yi ppi Mleft Mtop**

Generate HP LaserJet printer compatible output in **fn1.pcl(bin)**. The default graphics page size **xi yi** is 8 by 10 inches with **ppi** of 300 dots-per-inch. **Mleft Mtop** are optional left and top margins in inches. This option also reprints the command line and date and time in the upper left corner of the page. Remember to copy the resulting output file to the printer in binary.

**bmp nx ny bpp**

Generate a Windows-compatible bitmap file in **fn1.bmp(bin)** with default image size **nx ny** of 640 by 480 with **bpp**, bits-per-pixel, of 1. The allowed **bpp** values are 1, 4, or 8.

**gif nx ny**

Generate a CompuServe-compatible compressed GIF image in **fn1.gif** with default image size of 640 by 480.

**size nx ny**

Use image size of **nx** by **ny** pixels.

**inches xi yi** specify image size of **xi** by **yi** inches. If **nx ny** are not set, multiply by **ppi** to convert to pixels. If **nx ny** are set, calculate new **ppi**.

**ppi p** Set pixels/inch effective resolution.

**erase col** Initialize image to color **col**.

**fill f** Adjust MAP to fill fraction **f** of image.

**rotate** Force 90 degree rotation.

**antirotate** Force -90 degree rotation.

**upsidedown** Force 180 degree rotation.

**best f** FILL with optional rotation.

**fraction f** FILL/BEST fraction of total image.

**window n** Use window **n**.

**replace** Replace an existing **fn.uim(bin)** file.

**out fname** Place output in **fname.uim(bin)**.

**image fname** Initialize with existing image **fname.uim(bin)** (non-DOS only). This image forms the background for any current rendering.

**path pname** Prefix **pname** to the **fn.ugh(bin)** files.

**min n** Skip first **n** UGH commands in input file.

**max n** Stop after a total of **n** UGH input commands.

**comlim minc maxc**  
Only output UGH commands **minc** ≤ **com** ≤ **maxc**.

**Examples**

```
; X-windows: add text to a background
u-uim textlab -inch 8 6 -image backgrd -out backtext
```

```
; DOS simple conversion with colored background
u-uim vossplot ( erase 9 fill
```

```
; DOS convert and print on a HP-compatible laser printer
u-uim vossplot ( hp fill 0.9 rep
copy vossplot.pcl lpt1 /B
```