UGH Commands and Utilities

VP translates high level commands and data (.exx 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 \TeX or \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 SI DI.

The data in .ugh(bin) binary files consist of a sequence of UGH command. 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 swap. If value is omitted, returns and deactivates AX = current "hot-key." The DOS single-screen default "hot key" is 0x2207 for Ctrl-C.

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 ≥ 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 now nscreen. Returns the following data:

5

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 return 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=nx, CX=0, 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 be rotated by 90 degrees. If NBYTE < 8, the current settings are returned as AX=nx, BX=nx, CX=0, DX=y0, DI=rot.
Places all following graphics in specified window. 10 windows are allowed. Their default definitions are:

- full screen
- left half
- right half
- center 1/4
- upper half
- upper left 1/4
- upper right 1/4
- lower right 1/4
- lower left 1/4
- 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.

Fills current WINDCW with specified color and sets image start to lower left corner (see UGH command 28).

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

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

Most UGH drivers attempt to mimic this in default mode as much as possible.
TER mouse button aborts input. If just $x_1, x_2$ are given, they specify the starting point of the cursor. If $x_1, y_1, x_2, y_2$ are given, $x_1, y_1$ is the initial fixed point while $x_2, y_2$ 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 ≤ 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.

```
30 color width rx ry x0 y0 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

Draws a single line from \( x_1, y_1 \) to \( x_2, y_2 \). When possible, line type 34 is drawn with square ends extended width/2 beyond the end points.

35 color width x1 y1 x2 y2 ...

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

36 x0 y0 count offset segment

DOS only. Returns a horizontal scanline of count pixels beginning at SCREEN (not MAP); address \( x_0, y_0 \) to the PC address given by segment:offset. If count < 0, then return -count bytes in the -Y (down) direction starting \( x_0, y_0 \).

37 x1 y1 x2 y2 orient filespec

Processes the UGH image data in filespec to fill the box defined by the corners \( x_1, y_1 \) and \( x_2, y_2 \) 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

Sets the default HoriBy 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

Writes text on the graphics screen, where

<table>
<thead>
<tr>
<th>col</th>
<th>text color</th>
</tr>
</thead>
<tbody>
<tr>
<td>wid</td>
<td>width of strokes</td>
</tr>
<tr>
<td>len</td>
<td>if &gt;0 gives total length, if &lt;0 character width</td>
</tr>
<tr>
<td>h</td>
<td>if &gt;0 gives total height, if &lt;0 character height</td>
</tr>
<tr>
<td>font</td>
<td>Stroke font number, &lt;0 for rotation by 90 degrees</td>
</tr>
<tr>
<td>x0</td>
<td>starting position</td>
</tr>
</tbody>
</table>

HWRITE

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

Other values for 1lp 1yp define a rotation of the text by an angle theta in the counterclockwise direction about the starting point \( x_0, y_0 \), where \( \cos(\theta)=100+1lp \) and \( \sin(\theta)=100+1yp \).

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 "size=off+dx" or "size-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 "size-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\width\bot\text \_ will underline text with a line of color col, width wid, a distance (font-size\bot)/32 from the baseline:
  \text{e.g. \_4+10-10 this text will be underlined in color 4.}
- \_col\width\top\text \_ will draw a box around text with a line of color col, width wid, a distance (font-size\top)/32 and (font-size\bot)/32 from the baseline. If wid<0 the box will be filled:
  \text{e.g. \_4+10-10+8 this text will be boxed in color 4.}

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

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

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 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 A sour Ysource Xdest Ydest, specifies a BITBLT move in 8514A coordinates.

Special GESC one-byte codes for X-windows:

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

7.2 ughb2d

UGHb2d is a filter program to convert a binary .ugh(bin) file to .dec notation in stdout. The output may be redirected with > output 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 min max: only output UGH commands min ≤ com ≤ max.
- width n: integer output with minimum width n (default 4).
- msg frame: place message output in frame instead of console.
- labels: only dump labels (ugh command 0).
- hex: all output in hexadecimal.
7.3 ughd2b

ughd2b is a filter program (the inverse of ughd3d) 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

ughd2d voosp -rep ; convert voosp.dev to voosp.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 ...)

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-user.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 append
fill f ppi dot-in-X dot-in-Y (800 8000)
bu (default) gray color-num fraction
vb rgb color-num R G B
shade vttimage name.ext
colors min ugh-color-number (0)
replace pow pow (1.0)
reverse handler
hershey
7.5.1 General considerations

_u-ps_ converts binary .ug(b) 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.pro_ 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 .ug(b) 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-user.pro_ if it exists. Place any "permanent" options in this file.

Finally, _u-ps_ checks the command line for filenames and additional options. File names should be given in the form name.pro. When no ext is given, .ug(b) 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 .ug(b) 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-user.pro_ profiles but before any .ug(b) 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.

7.5.2 Output page format

_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 PAGESIZE 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 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 GSIZEx option.

Default processing adds 1/4-inch corner ticks 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 ticks 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 930 points/inch. The following options can be used to change the scaling and placement:

- **center**: Center all UGH elements in page.
- **fill f**: Change scaling to center and fill up to a fraction f of the graphics size.
- **best f**: Same as FILL with optional rotation.
- **fraction f**: Change FILL fraction from default of 1.0.
- **rotate**: Force 90 degree rotation.
- **rotate 0**: Force no rotation.
- **rotate 45**: 45 degree rotation.
- **antitolate**: Force -90 degree rotation.
- **ppi 1600 1600**: Reduce size to 50% of standard (by increasing the effective UGH units per inch).
- **gsizex 8**: 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 .ug(b) 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
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 \texttt{bw} option reverses this option to produce a negative image. Even with default \texttt{bw} processing, included images are processed as shaded halftones. The \texttt{shade} option specifies that different UGH colors correspond to different halftone densities. The default \texttt{u-ps}.pro sets up 16 sample shades with the \texttt{gray} option that are used when \texttt{shade} is specified. The actual halftone screen size may be changed with the \texttt{screen} option, but the results depend on the output device.

The \texttt{color} option specifies that different UGH colors correspond to different RGB intensities. The default \texttt{u-ps}.pro sets up 16 sample RGB values with the \texttt{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 \texttt{14}) with the \texttt{map} option or by including the filename at the start of the list of input files. The \texttt{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 hardcopy the default is white). To remain consistent while allowing users to change colormaps, a \texttt{color} specification with color 0 as \texttt{RGB} 0 0 0 is translated to 255 255 255 for output. To force a black background, use color 0 as \texttt{RGB} 0 0 1.

The \texttt{sidecolors} option sets the same default colors as \texttt{color} but does not reverse black and white. Color 0, the background, is black and color 7, the normal text color, is white.

When the \texttt{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 \texttt{.eps} instead of \texttt{.ps}. No bitmapped screen preview is included and no showpage operator is included at the end. The \texttt{eps} option is incompatible with the \texttt{multipage}, \texttt{noprologue}, and \texttt{ehandler} options. It is recommended that the \texttt{exact page} option be used along with \texttt{eps} to prevent the margin tics and header from appearing in the output and to prevent any additional rescaling by \texttt{u-ps}. Using \texttt{fill} or \texttt{best} with \texttt{eps bare} may cause the BoundingBox specification to be incorrect.

The bounding box is an estimate of the extent of the graphics based on the 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 \texttt{fraction} option as with \texttt{fill} or \texttt{best}. Thus, \texttt{eps exact bare fraction 0.9} will give an \texttt{ eps} file with a bounding box 10% larger (5% in each direction) than the actual graphics.

Examples

\begin{verbatim}
\texttt{u-ps vossplot -best -replace}
\end{verbatim}

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

\begin{verbatim}
\texttt{u-ps f1 spec.pro f2 \{bare stdout \> lpt1}
\end{verbatim}

Options are read from \texttt{u-ps}.pro and then from \texttt{u-ps-user.pro}. The UGH elements in \texttt{f1.ugh(bin)} are translated with exact sizing \texttt{spec.pro} changes some of the font specifications, and \texttt{f2.ugh(bin)} is processed with exact scaling. The \texttt{BASE} 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.

\begin{verbatim}
\texttt{u-ps f1 f2 f3 -fill -frac 0.8 -std \> 1lp \> lps}
\end{verbatim}

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

With the \texttt{u-ps} customization features, output can be tailored for different printers without changing the input \texttt{.ugh(bin)} file. Examples of customization are given in the default \texttt{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 \texttt{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 box field in the \texttt{font} option.
Thus, to include a German ü, from \texttt{VP} you could use one of the following:

- \texttt{comment X Y} \texttt{\textbackslash\texttt{366bung hilft}} \texttt{; for default \{bf u-ps\} substitution}
- \texttt{comment X Y} \texttt{\textbackslash\texttt{366bung hilft}} \texttt{; double \ because VP special char}
- \texttt{comment X Y anis} \texttt{\textbackslash\texttt{366bung hilft}} \texttt{; ASIS eliminates need for double \textbackslash}

The \texttt{u:} substitution method is recommended 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, \texttt{u-ps} searches for defaults as follows:

- on DOS: \texttt{fname.ugh} \texttt{(case insensitive)}
- current directory, then all subdirectories in \texttt{PATH}.
- unix: \texttt{fname.ugh} or \texttt{fname.ubin} \texttt{(lower case)}
- current directory, then all subdirectories in \texttt{UGHSBINPATH}, then all subdirectories in \texttt{PATH}.
- \texttt{u-ps} uses a similar strategy to find the Hershey font stroke data:

- on DOS: \texttt{Hxx.FNT} \texttt{(case insensitive)}
- current directory, then all subdirectories in \texttt{PATH}.
- unix: \texttt{Hxx.FNT} or \texttt{Hxx.FNTBIN} \texttt{(upper case)}
- current directory, then all subdirectories in \texttt{FNTDBINPATH}, then all subdirectories in \texttt{PATH}.

\texttt{UHG} command \texttt{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 \texttt{fname ula(bin)}) are included in HEX notation in the output file. This, of course, can result in large output files. If \texttt{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 \texttt{8bits} the files are half as large.

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

Similarly, a \texttt{.uim(bin)} image file may be printed by itself by including \texttt{name ula(bin)} in the list of input files. Thus:

\texttt{u-ps sample.uimbin \{fill fraction 0.6}
7.5.9 PC considerations

`u-ps demovp (fill stdout > lpt1`

will redirect the PostScript stdout output directly to the printer.

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

`out fname.ext` Places PostScript output in `fname.ext`. If no `.ext` is given, `.ps` will be used. If the file already exists, it can be overwritten with REPLACED.

`replace` Allows default output to `fn1.ps` to overwrite any existing file. If default output to `fn1.ps` is used and REPLACED is not given, an existing `fn1.ps` will not be overwritten.

`append` 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.

`multipage` Generates a separate page of PostScript output for each `.ugh(bin)` file named on the command line.

`stdout` Places PostScript in stdout. This output may be redirected on the command line or piped to another process such as the printer queue.

`copies n` Produces a copies of the page, where `n≥5`. For more than 5 copies, edit the output `.ps` file to include the line

`/n#copies 9 def before showpage.`

`page X Y` Sets page's X and Y size in inches (default 8.5 11).

`margin L,R T,B col` Sets left-right top-bottom margins in inches and margin option `color col` (0 omit, > 0 for 1/4 inch corner ticks, < 0 for full lines).

`gscale GX GY` Redefines the allowed graphics area inside margins.

`fill f` Adjust scaling to fill to margins, optional fill fraction `f`.

`rotate` Force 90 degree rotation on page.

`rotate DEG` Force rotation by DEG degrees.

`best f` Fill with optional rotation to best match graphics aspect ratio with `gscale`, optional fill fraction `f`.

`fraction f` Fill center fraction `f` of graphics area (default 1.0).

`exact` No scaling of UGH coordinates (default). This is useful to generate graphics having exactly the size specified in `VP` for overlaying graphs.

`center` No scaling of UGH coordinates, but center output in page.

`nopro` Do not include any PostScript prologue procedures.

`eps` Output Encapsulated PostScript format.

`for "text"` PostScript header for field and top center label. If the text includes spaces, use quotes as delimiters.

`label col pnt` Page label color and optional point size.

`bare` Omit page label and margin lines or corner ticks.

`antirotation` Force -90 degree rotation on page.

`min n` Begin output with UGH command `n`.

`max n` End output after UGH command `n`.

`comlim min maxc` Only output UGH commands `min≤com≤maxc`.

`ppi ppxpix` 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.

`msg fname.ext` 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 xmult
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.

tfont 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 26 Times-Italic -701 1.3 1.2 1 ; narrow outline, color 7
font 12 Hershey 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 "i" "i" 0 ; i = 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 \#\#1111 ; copyright symbol
sub u# : \366 ; German 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 will 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.
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

\texttt{u-ps demovp (fill frac 0.9 cmap mycolors)}
\texttt{u-ps colors demovp (out demovp fill frac 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.

\textbf{reverse} Reverse byte value mapping so 0 = white, 255 = black.

\textbf{power pow} Set image scaling pow>1 for increased, pow<1 for decreased contrast (default 1.0).

\textbf{vltimage name} Use the colormap in name.ugh(bin) to determine brightness (0.0–1.0) as

\[
brite[k] = (B[k]+G[k]+R[k])/765
\]

of color k for shaded images or as specifying the rgb triplets for \texttt{color}.

\section*{required files}
In addition to the executable program and user-specified .ugh(bin) files, \texttt{u-ps} searches for the following files:

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

\texttt{u-ps-user.pro} optional .pro file where the user should place any desired modifications to \texttt{u-ps.pro}. This file is read after \texttt{u-ps.pro}.

\texttt{u-ps.ref} short reference list of the available options.

\texttt{u-ps-pro.ps} the \texttt{u-ps} prologue of special UGH PostScript procedures.

\texttt{ehandler.ps} standard PostScript error-handling procedures.

\section*{u-cgm}
\texttt{u-cgm fin (option1 option2 ...)}
\texttt{u-cgm fin -option1 -option2 ...}
\texttt{u-cgm -option1 -option2 fin}

\texttt{u-cgm} converts a binary .ugh(bin) file to CGM (Computer Graphics Metafile) format in fin.cgm(bin). \texttt{u-cgm} is a good choice for including graphics in many pc-based word processors such as WordPerfect or Microsoft Word. \texttt{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 \texttt{hershey} 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 \texttt{vb} option.

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

\textbf{hershey} Stroke all characters as UGH Hershey fonts (default).

\textbf{nohershey} Use CGM default text (no subscripts, superscripts or Greek).

\textbf{replace} Replace an existing fin.cgm(bin) file.

\textbf{out name} Place output in name.cgm(bin).

\textbf{fraction f} Set output size so graphics occupy the center fraction f.

\textbf{page Xin Yin pp1} Set output limits corresponding to a Xin by Yin page (inches) with optional pp1 (default 800).

\textbf{center} When used with PAGE, forces centered output.

\textbf{min n} Skip first n UGH commands in input file.

\textbf{max n} Stop after a total of n UGH input commands.

\textbf{comlim minc maxc} Only output UGH commands \texttt{minc} to \texttt{maxc}.

\textbf{cmap name} Include colors from name.ugh(bin). Try cmap \texttt{vb} to force black graphics on white background.

\textbf{msg name} Place message output in name instead of console.
u-uim

converts .ugh (bin) files into various types of pixel image files:

- UGH images in fn1.ugh (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.

Examples

- **u-cgm voessplot -horeh -rep -cmap default**
- **u-cgm voessplot ( horeh rep cmap wb page 6 4 center**

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

**7.7 u-uim**

- **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 0 degree rotation.

- **anticotate**
  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 fn1.ugh (bin) file.

- **out frame**
  Place output in frame. 1.ugh (bin).

- **image fname**
  Initialize with existing image frame. 1.ugh (bin) (non-DOS only). This image forms the background for any current rendering.

- **path p**
  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 mxc**
  Only output UGH commands minc <= com <= mxc.

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 voessplot ( erase 9 fill**
- **; DOS convert and print on a HP-compatible laser printer u-uim voessplot ( hp fill 0.9 rep copy voessplotpcl lpt1 /S**