Section 9

HP PCL Graphics Commands

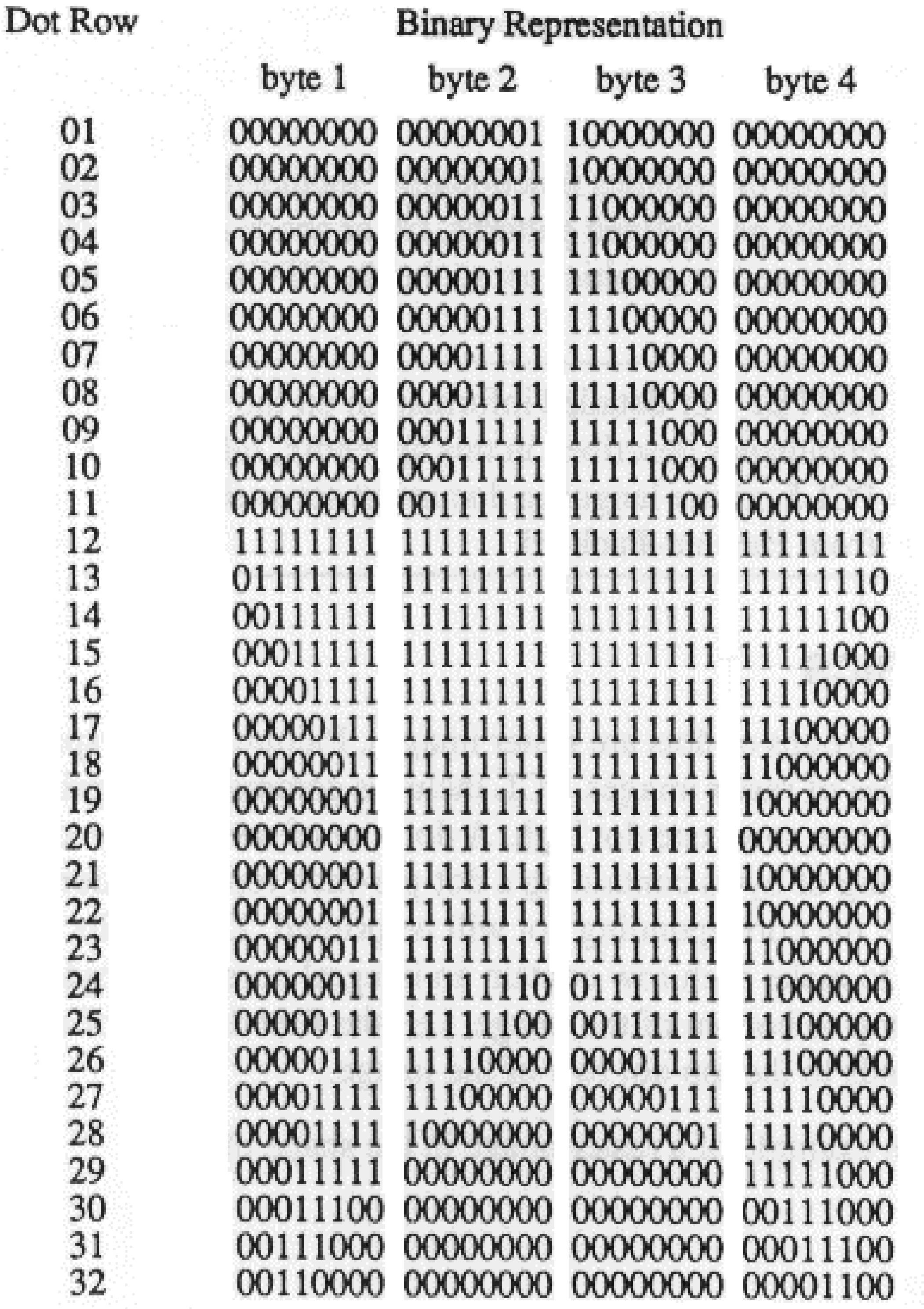
PCL provides several forms of graphics functionality. Included is the ability to build dot-per-bit raster images, create pre-defined patterns, fill or shade rectangular areas with pre-defined patterns, and print vector graphics using the HPGL/2 graphics language.*

NOTE: For detailed information on the commands described in this section, refer to the PCL5 printer language document set available from Hewlett-Packard.

RASTER GRAPHICS

Images composed of groups of dots are raster images. Pictures in newspapers or on television are examples of raster images. PCL includes commands for printing raster images. The image is divided into rows one dot high. A dot row of raster image data is transferred to the printer as a string of bytes containing a dot-per-inch representation of the row. If a bit in a row is set to one, the corresponding dot will be printed. Zeroed bits must be added to the end of each row to make each row contain an integral number of bytes. The dot rows are organized from top to bottom of the image, i.e., the first dot row of data transferred to the printer corresponds to the top dot row of the image.

Figure 9-1 shows an example of a raster image in the shape of a star.



Dot Row	Binary Representation			
	byte 1	byte 2	byte 3	byte 4
01	00000000	00000001	10000000	00000000
02	00000000	00000001	10000000	00000000
03	00000000	00000011	11000000	00000000
04	00000000	00000011	11000000	00000000
05	00000000	00000111	11100000	00000000
06	00000000	00000111	11100000	00000000
07	00000000	00001111	11110000	00000000
08	00000000	00001111	11110000	00000000
09	00000000	00011111	11111000	00000000
10	00000000	00011111	11111000	00000000
11	00000000	00111111	11111100	00000000
12	11111111	11111111	11111111	11111111
13	01111111	11111111	11111111	11111110
14	00111111	11111111	11111111	11111100
15	00011111	11111111	11111111	11111000
16	00001111	11111111	11111111	11110000
17	00000111	11111111	11111111	11100000
18	00000011	11111111	11111111	11000000
19	00000001	11111111	11111111	10000000
20	00000000	11111111	11111111	00000000
21	00000001	11111111	11111111	10000000
22	00000001	11111111	11111111	10000000
23	00000011	11111111	11111111	11000000
24	00000011	11111110	01111111	11000000
25	00000111	11111100	00111111	11100000
26	00000111	11110000	00001111	11100000
27	00001111	11100000	00000111	11110000
28	00001111	10000000	00000001	11110000
29	00011111	00000000	00000000	11111000
30	00011100	00000000	00000000	00111000
31	00111000	00000000	00000000	00011100
32	00110000	00000000	00000000	00001100

Figure 9-1 Star-Shaped Raster Image

Raster Graphics Resolution

Raster graphics can be printed at 300, 150, 100 or 75 dots-per-inch. This command designates the resolution of subsequent raster data transfers.

```
ESC*t#R
Decimal:  027 042 116 # 082
Hex:      1B 2A 74 # 52
```

```
#:  75 dots-per-inch
    100 - 100 dots-per-inch
    150 - 150 dots-per-inch
    300 - 300 dots-per-inch
```

This command must be sent prior to the start graphics command. The factory default resolution is 75 dots-per-inch.

The IntelliBar has 300 dots-per-inch print resolution. The IntelliBar automatically expands raster graphics transferred at resolutions less than 300 dots-per-inch to 300 dots-per-inch during printing.

Figure 9-2 illustrates a single bit translated into the corresponding printed dots in each of the four resolutions:

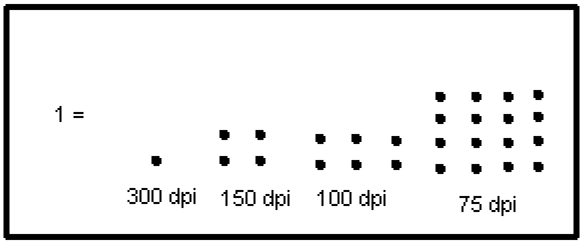


Figure 9-2 Dot Translation

Lower resolution graphics occupy less user memory. For example, the number of bits required to represent a two-inch by three-inch image at 75 dots-per-inch is 34,200. The same image at 300 dots-per-inch requires 540,000 bits.

Raster Graphics Presentation

```
ESC*r0F  (prints in orientation of logical page)
Decimal:  027 042 114 048 070
Hex:      1B 2A 72 30 46
```

```
ESC*r3F  (raster image prints along the width of the physical page)
Decimal:027 042 114 051 070
Hex:      1B 2A 72 33 46
```

This command specifies the orientation of the raster image on the logical page.

A value of 0 means that a raster row prints in the positive X-direction of the PCL coordinate system. (The print direction translates the PCL coordinate system.)

A value of 3 means that the raster graphics prints along the width of the physical page, regardless of the logical page orientation.

The IntelliBar prints raster graphics along the width of the physical page, regardless of addressable print area orientation. In portrait orientation, a raster row will be printed in the positive X-direction of the PCL coordinate system and a subsequent raster row will be printed beginning at the next dot row position in the positive Y-direction. In landscape orientation, a raster row will be printed in the positive Y-direction of the PCL coordinate system and a subsequent raster row will be printed beginning at the next dot row position in the negative X-direction. Figure 9-3 illustrates a master graphics representation.

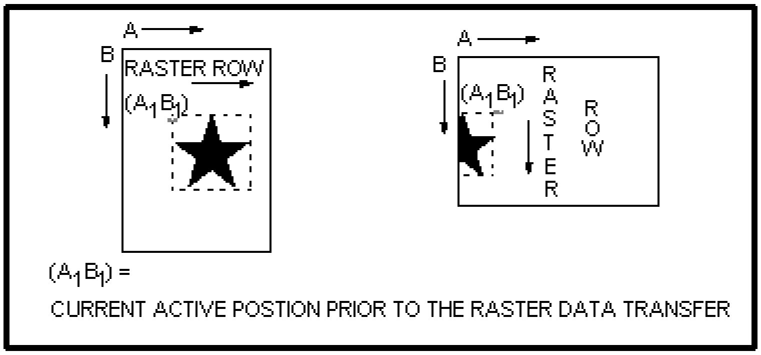


Figure 9-3 Raster Graphics Presentation

Raster Height Command

ESC*r#T

Decimal: 027 042 114 #..# 084
Hex: 1B 2A 72 #..# 54

#: height in raster rows
Range = 0 to (logical page length – current Y-position of the 0, cursor)

This command specifies the height in raster rows of the raster area. Height is the direction perpendicular to the direction that raster rows are laid down; height is therefor subject to the current raster presentation mode and print direction (see “Transfer Raster Data” and “Start Raster Data” for further information. Unspecified data maps to either white or transparent, depending on the source transparency mode (see “Select Source Transparency Mode”).

This command fills the raster area to the full raster height with zeroed rows. Only raster data appearing within the intersection of the logical page, the printable area, and if set, the raster width and height, is printed. Data outside the intersection is clipped.

Raster Width Command

ESC*r#S

Decimal: 027 042 114 #..# 083
Hex: 1B 2A 72 #..# 53

#: width in pixels of the specified resolution
Default = depends on the raster presentation mode setting
0: width = width of logical page – left graphics margin
3: width = dimension of logical page along paper length – left graphics margin
Range = 0 to (logical page length – left graphics margin)

This command the width in pixels of the raster area. Width is the direction that the raster rows are laid down; height is therefore subject to the current raster presentation mode and print direction (see “Transfer Raster Data” and “Start Raster Data” for further information.

This command allows you to tell the printer to pad raster rows that are not specified for the full raster width with zeroes. Unspecified data maps to either white or transparent, depending on the source transparency mode (see “Select Source Transparency Mode”).

Only raster data appearing within the intersection of the logical page, the printable area, and if set, the raster width and height, is printed. Data outside the intersection is clipped.

Start Raster Graphics

The start raster graphics sequence specifies the left raster graphics margin.

ESC*r#A
Decimal: 027 042 114 48 065 (left raster graphics margin)
Hex: 1B 2A 72 30 41 (left raster graphics margin)

Decimal: 027 042 114 49 065 (current cursor)
Hex: 1B 2A 72 31 41 (current cursor)

#: 0 = left graphics margin location is x-position 0.
1 = left graphics margin at the current x-position (current cursor position

A value of 0 specifies that the left graphics margin is at X position 0. A value of 1 specifies that the left graphics margin is at the current X position (the current cursor position).

Once a start raster graphics command is received by the printer, raster graphics resolution, raster graphics presentation mode, raster height, raster width, and left raster graphics margins are fixed until an end raster graphics command is received.

Raster Y Offset Command

```
ESC*b#Y
Decimal:  027 042 098 #..# 089
Hex:      1B 2A 62 #..# 59
```

# = number of raster lines of vertical movement
Range = 0 - 32767

This command moves the cursor position vertically the specified number of raster lines from the current raster position in the raster area.

This command is recognized only while in raster graphics mode and only within the raster area.

Set Compression Method Command

```
ESC*b0M  (uncoded - no compression)
Decimal:  027 042 098 048
Hex:      1B 2A 62 30 41

ESC*b1M (run-length encoded)
Decimal:  027 042 098 049 077
Hex:      1B 2A 62 31 41

ESC*b2M (TIFF - not supported by IntelliBar)
Decimal:  027 042 098 050 077
Hex:      1B 2A 62 32 41

ESC*b3M (delta row)
Decimal:  027 042 098 051 077
Hex:      1B 2A 62 33 41

ESC*b4M  (reserved)
Decimal:  —
Hex:      —

ESC*b5M  (adaptive compression)
Decimal:  027 042 098 053 077
Hex:      1B 2A 62 35 41
```

This command allows you to code raster data in one of four compressed formats:

- run-length encoding
- tagged imaged file format (TIFF)

- delta row compression
- adaptive compression

The choice of compression methods affects both the amount of code needed to generate a raster graphics image and the efficiency with which the image is printed. Compressed data formats allow for efficient transfer of data from the host system to the printer. However, compressed data formats do not reduce the amount of printer memory required to produce an image.

Transfer Raster Data

The transfer raster data command is used to transfer a row of raster data to the printer.

ESC*b#W [raster data]	
Decimal:	027 042 098 # 087
Hex:	1B 2A 62 # 57

The value field (#) identifies the number of bytes in the raster row. These bytes are interpreted as one row of raster graphics data that will be printed at the current Y position at the left raster graphics margin. Upon completion of this command, the current active position is at the beginning of the next raster row at the left raster graphics margin.

Within the raster data, each bit describes a single dot. The most significant bit (bit 7 is the most significant, bit 0 is the least significant) of the first byte of data corresponds to the first dot within that row. If a bit is set to 1, the corresponding dot will be printed. Each dot of the raster data is expanded according to the specified raster resolution.

Raster graphics is independent of the text area and perforation skip mode, i.e., these boundaries are ignored.

Raster graphic images are limited to the printable area; images that extend beyond the printable area are clipped.

End Raster Graphics

The end raster graphics sequence signifies the end of the transfer of a raster graphic image.

ESC*rB	
Decimal:	027 042 114 066
Hex:	1B 2A 72 42

RASTER GRAPHICS EXAMPLE

To transfer a raster graphic image (see Figure 9-4) in the shape of an star, perform the following steps:

1. Position the cursor:

   ```
   ESC*p300x400Y
   ```

   This moves the cursor to dot position (300, 400) within the PCL coordinate system.

2. Specify the raster graphics resolution:

   ```
   ESC*t75R
   ```

   This sets the raster graphics resolution to 75 dots-per-inch.

3. Specify the left raster graphics margin:

   ```
   ESC*r1A
   ```

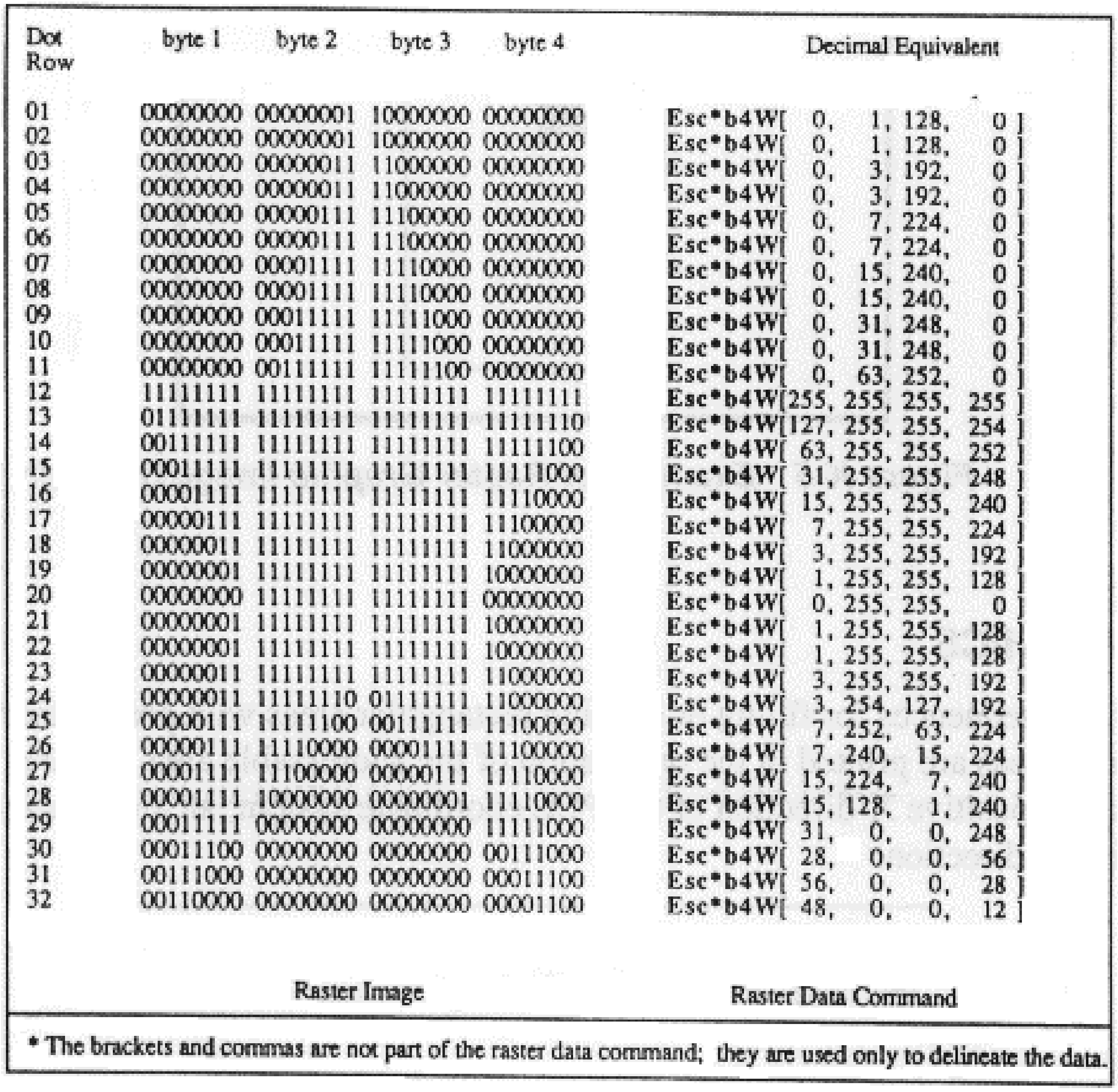
   This sets the left graphics margin to the current X position (300).

4. Transfer the raster data to the printer:

   Divide the image into dot rows and transfer each dot row to the printer as a string of bytes as illustrated in Figure .

5. Signify the end of the raster graphic image transfer:

   ```
   ESC*rB
   ```



Dot Row	byte 1	byte 2	byte 3	byte 4	Decimal Equivalent
01	00000000	00000001	10000000	00000000	Esc*b4W[0, 1, 128, 0]
02	00000000	00000001	10000000	00000000	Esc*b4W[0, 1, 128, 0]
03	00000000	00000011	11000000	00000000	Esc*b4W[0, 3, 192, 0]
04	00000000	00000011	11000000	00000000	Esc*b4W[0, 3, 192, 0]
05	00000000	00000111	11100000	00000000	Esc*b4W[0, 7, 224, 0]
06	00000000	00000111	11100000	00000000	Esc*b4W[0, 7, 224, 0]
07	00000000	00001111	11110000	00000000	Esc*b4W[0, 15, 240, 0]
08	00000000	00001111	11110000	00000000	Esc*b4W[0, 15, 240, 0]
09	00000000	00011111	11111000	00000000	Esc*b4W[0, 31, 248, 0]
10	00000000	00011111	11111000	00000000	Esc*b4W[0, 31, 248, 0]
11	00000000	00111111	11111100	00000000	Esc*b4W[0, 63, 252, 0]
12	11111111	11111111	11111111	11111111	Esc*b4W[255, 255, 255, 255]
13	01111111	11111111	11111111	11111110	Esc*b4W[127, 255, 255, 254]
14	00111111	11111111	11111111	11111100	Esc*b4W[63, 255, 255, 252]
15	00011111	11111111	11111111	11111000	Esc*b4W[31, 255, 255, 248]
16	00001111	11111111	11111111	11110000	Esc*b4W[15, 255, 255, 240]
17	00000111	11111111	11111111	11100000	Esc*b4W[7, 255, 255, 224]
18	00000011	11111111	11111111	11000000	Esc*b4W[3, 255, 255, 192]
19	00000001	11111111	11111111	10000000	Esc*b4W[1, 255, 255, 128]
20	00000000	11111111	11111111	00000000	Esc*b4W[0, 255, 255, 0]
21	00000001	11111111	11111111	10000000	Esc*b4W[1, 255, 255, 128]
22	00000001	11111111	11111111	10000000	Esc*b4W[1, 255, 255, 128]
23	00000011	11111111	11111111	11000000	Esc*b4W[3, 255, 255, 192]
24	00000011	11111110	01111111	11000000	Esc*b4W[3, 254, 127, 192]
25	00000111	11111100	00111111	11100000	Esc*b4W[7, 252, 63, 224]
26	00000111	11110000	00001111	11100000	Esc*b4W[7, 240, 15, 224]
27	00001111	11100000	00000111	11110000	Esc*b4W[15, 224, 7, 240]
28	00001111	10000000	00000001	11110000	Esc*b4W[15, 128, 1, 240]
29	00011111	00000000	00000000	11111000	Esc*b4W[31, 0, 0, 248]
30	00011100	00000000	00000000	00111000	Esc*b4W[28, 0, 0, 56]
31	00111000	00000000	00000000	00011100	Esc*b4W[56, 0, 0, 28]
32	00110000	00000000	00000000	00001100	Esc*b4W[48, 0, 0, 12]

Raster Image

Raster Data Command

* The brackets and commas are not part of the raster data command; they are used only to delineate the data.

Figure 9-4 Example of Raster Graphic Image Data

This prints the star as shown in Figure 9-5.

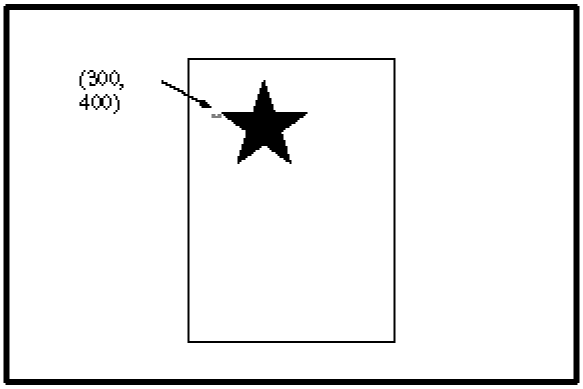


Figure 9-5 Example of Raster Graphic ImageTransfer

FILL GRAPHICS

PCL includes sequences for filling or shading rectangular areas with pre-defined patterns. Rectangular areas are printed in the orientation of the addressable area. An area's width extends in the positive X-direction of the PCL coordinate system; an area's height extends in the positive Y-direction.

NOTE: Rectangular areas are not affected by the raster graphics resolution command.

Horizontal Rectangle Size (Decipoints)

This horizontal rectangle size sequence specifies the rectangle width in decipoints.

ESC*c#H
Decimal: 027 042 099 # 072
Hex: 1B 2A 63 # 48

#: Number of decipoints (1/720 inch)

The value field (#) is valid to four decimal places.

The printer converts the specified width to dots by rounding up to an integral number of dots. For example, 5 decipoints, which corresponds to 2.08 dots on the printer, is converted to 3 dots.

The factory default horizontal rectangle size is 0.

Horizontal Rectangle Size (Dots)

This horizontal rectangle size command specifies the rectangle width in dots.

ESC*c#A
Decimal: 027 042 099 # 065
Hex: 1B 2A 63 # 41

#: Number of dots

The factory default horizontal rectangle size is 0.

Vertical Rectangle Size

This vertical rectangle size sequence specifies the rectangle height in decipoints.

ESC*c#V
Decimal: 027 042 099 # 066
Hex: 1B 2A 63 # 42

#: Number of decipoints (1/720 inch)

The value field (#) is valid to four decimal places.

The printer converts the specified width to dots by rounding up to an integral number of dots. For example, 5 decipoints, which corresponds to 2.08 dots on the printer, is converted to 3 dots.

The factory default vertical rectangle size is 0.

Vertical Rectangle Size (Dots)

This vertical rectangle size sequence specifies the rectangle height in dots.

ESC*c#B
Decimal: 027 042 099 # 066
Hex: 1B 2A 63 # 42

#: Number of dots

The factory default vertical rectangle size is 0.

Area Fill (Pattern ID)

This sequence specifies the level of shading or type of pattern fill to be used when filling a rectangular area.

ESC*c#G	
Decimal:	027 042 099 # 071
Hex:	1B 2A 63 # 47

The value field (#) identifies the level of shading or type of HP-defined pattern. Eight shading levels are defined within PCL. To specify a shading level use any value between 1 and 100. Use a value within the range indicated in Figure 9-6 for the desired shading level.

Six HP-defined fill patterns are defined within PCL. To specify a fill pattern type use any value between 1 and 6. Use a value indicated in Figure 9-7 for the desired fill pattern. Note that the patterns are not rotated when the orientation changes.

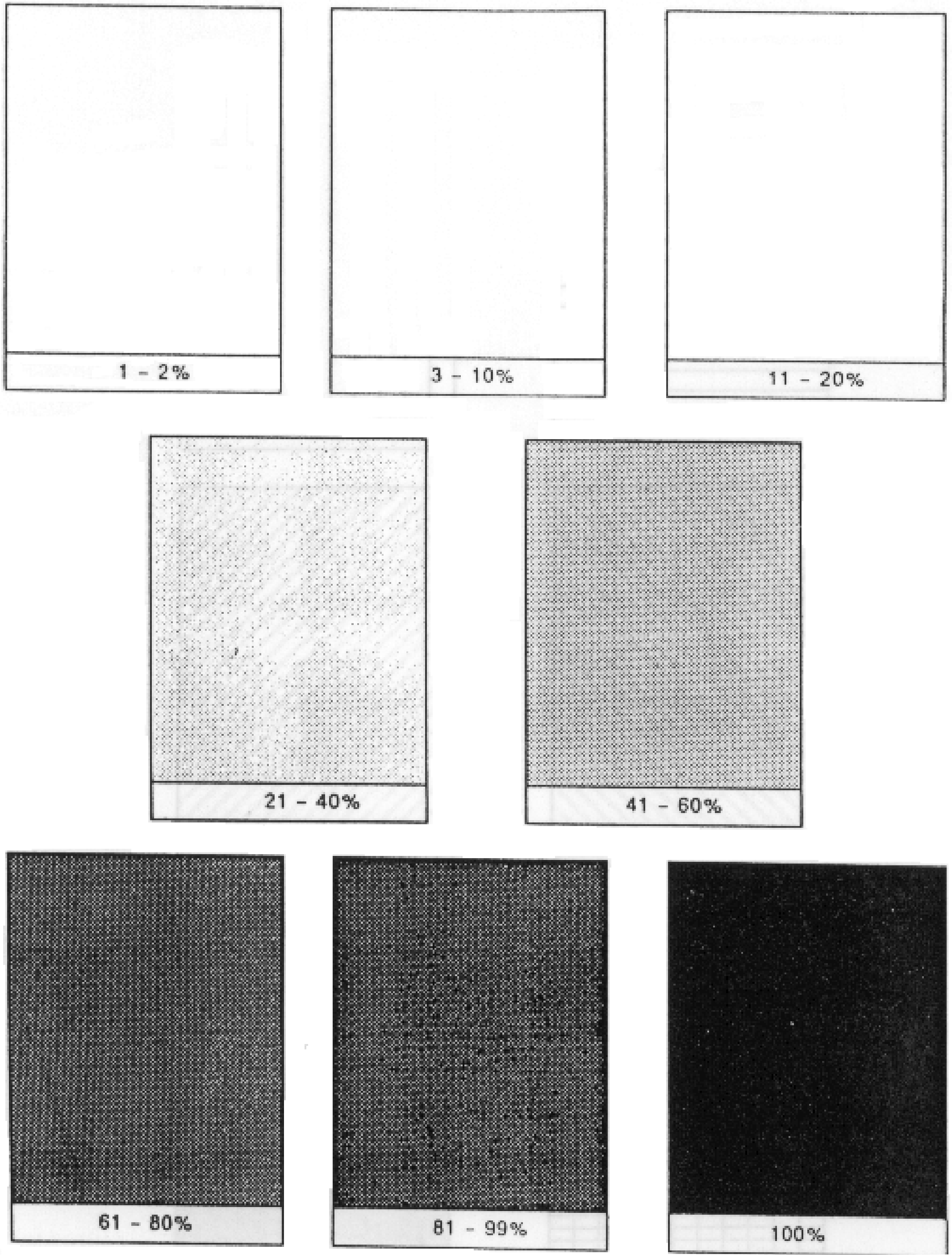


Figure 9-6 Shading Levels

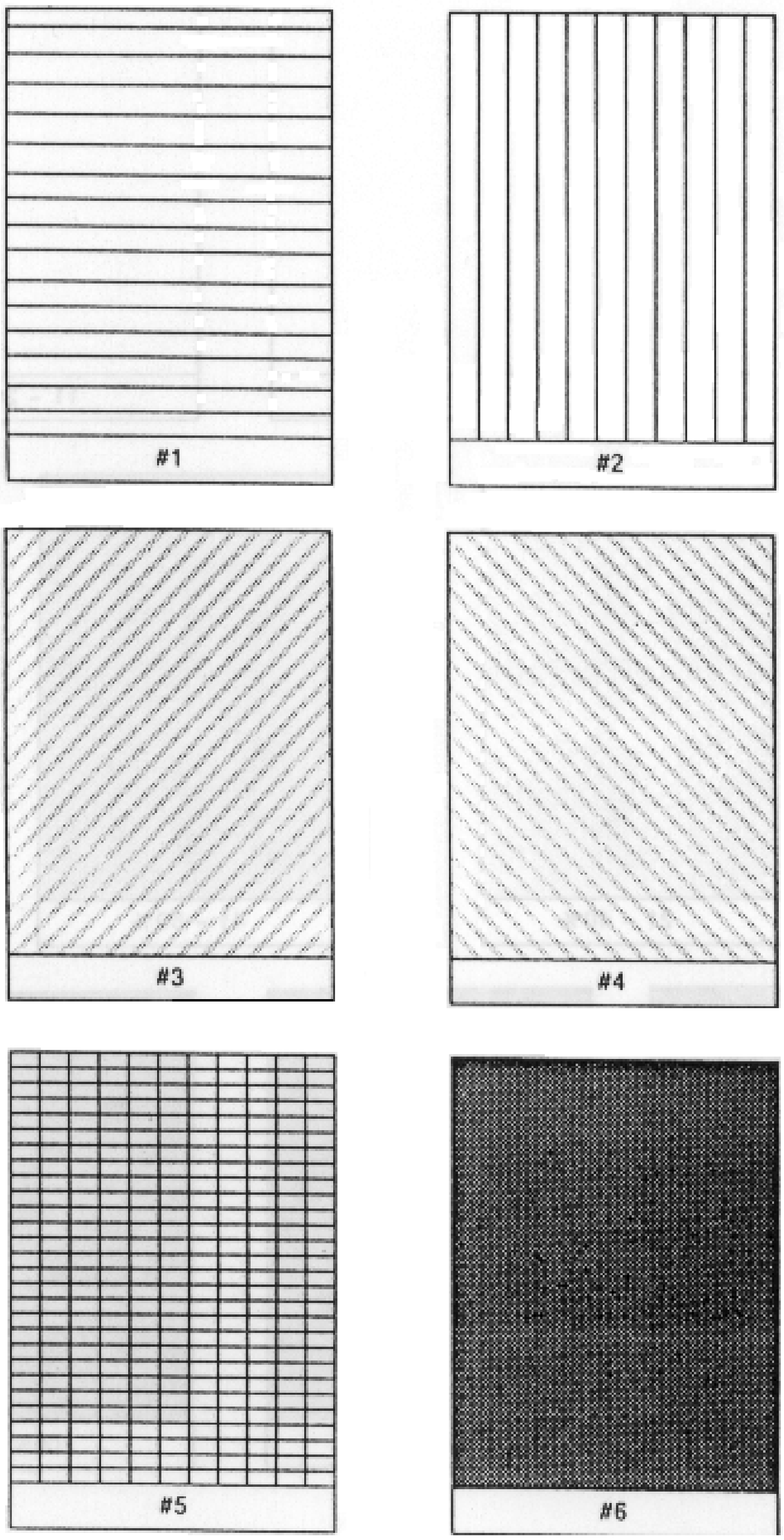


Figure 9-7 Fill Patterns

Fill Rectangle Area

This sequence fills a rectangular area of the specified width and height.

```
ESC*c#P
Decimal:  027 042 099 #..# 080  (#..# = 48 through 53)
Hex:      1B 2A 63 #..# 50  (#..# = 30 through 35)
```

#: 0 = Solid fill
1 = Erase (Solid white area fill)
2 = Shaded fill
3 = HP defined pattern fill
4 = User defined pattern fill
5 = Current pattern fill

The level of shading or type of patterned fill used when filling a rectangle is specified by the current area fill ID. A solid-filled rectangular area is also known as a black rule and does not require specification of an area fill ID.

The upper left corner of the rectangular area is located at the current active position. The current active position is not changed during the printing of a rectangular area.

Rectangular areas are independent of the text area and perforation skip mode, i.e., these boundaries are ignored.

Rectangular areas are limited to the addressable area; rectangular areas that extend outside the area are clipped.

Rectangle Area Fill Examples

To print a 3- x 5-inch black rule, perform the following steps.

1. Position the cursor:

   ```
   ESC*p300x400Y
   ```

   This moves the cursor to dot position (300, 400) within the PCL coordinate system.

2. Specify the width of the rule:

   ```
   ESC*c900A
   ```

   This sets the rule width to 900 dots (3 inches).

3. Specify the height of the rule:

   ```
   ESC*c1500B
   ```

   This sets the rule height to 1500 dots (5 inches).

4. Print the rule:

   ESC*c0P

   This example prints the following (see Figure 9-8):

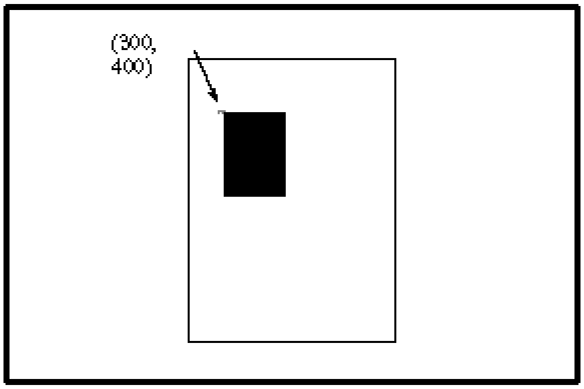


Figure 9-8 Solid Fill Example

To print a 3- x 5-inch 25% shaded rectangle, perform the following steps.

1. Position the cursor:

   ESC*p300x400Y

   This moves the cursor to dot position (300, 400) within the PCL coordinate system.

2. Specify the width of the rectangle:

   ESC*c900A

   This sets the rectangle width to 900 dots (3 inches).

3. Specify the height of the rectangle:

   ESC*c1500B

   This sets the rectangle to 1500 dots (5 inches).

4. Specify the area fill ID:

   ESC*c25G

   This sets the area fill ID to 25.

5. Print the rectangular shaded area:

   ESC*c2P

This example prints the following (see Figure 9-9):

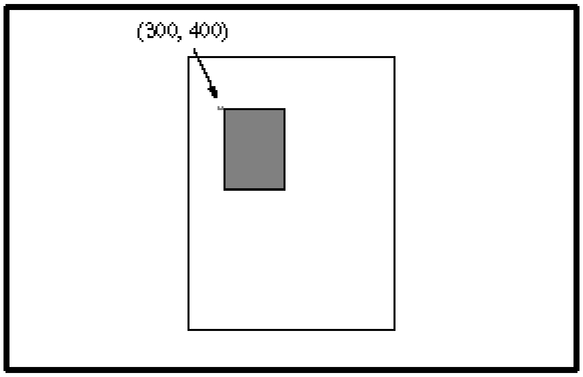


Figure 9-9 Shaded Fill Example

To print a 3- x 5-inch rectangular area filled with a horizontal line pattern, perform the following steps:

1. Position the cursor:

    ESC*p300x400Y

    This moves the cursor to dot position (300, 400) within the PCL coordinate systems.

2. Specify the width of the rectangle:

    ESC*c900A

    This sets the rectangle width to 900 dots (3 inches).

3. Specify the height of the rectangle:

    ESC*c1500B

    This sets the rectangle height to 1500 dots (5 inches).

4. Specify the area fill ID:

    ESC*c1G

    This sets the area fill ID to 1.

5. Print the rectangular pattern-filled area:

    ESC*c3P

    This example prints the following (see Figure 9-10):

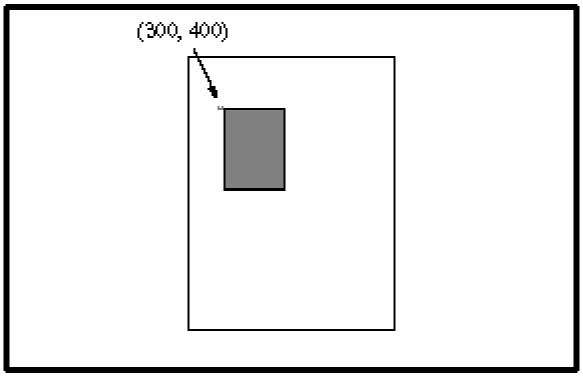


Figure 9-10 Patterned Fill Example

PCL PRINT MODEL

The Print Model feature allows images and characters to be filled with any of the printer's pre-defined shading or cross-hatched patterns. Images include any raster graphic, such as those described previously in this section (see "Raster Graphics"), a rectangular fill area (see "Fill Graphics"), or a character or characters selected from a font.

Print Model operations define a pattern, source image, and destination image. These are applied to each other using the Print Model's transparent and opaque modes to produce an image that is a combination of the others (see Figure 9-11).

- pattern

  The pattern is the design that is painted through the black ("1" bits) area of the source image onto the destination image. For patterns, Print Model uses one of the printer's internal pre-defined eight shading patterns, one of six cross-hatch patterns, or a user defined pattern.

- source image

  The source image is an image in which the black ("1" bits) are replaced by the specified pattern. This is like a stencil through which the pattern is applied to the destination image. The source image may be defined as a rectangular fill area, a raster graphics image, or characters.

- destination image

  The destination image is the image onto which the source image/pattern combination is placed. The destination image is the result of any previous operations.

- source transparency mode

  Source transparency mode is the transparency or opaqueness of the source image's white pixels (the "0" bits) as they are applied to the destination image. Setting the source transparency mode to 1 (opaque) applies the source image white pixels to the destination image; with a setting of 0 (transparent), these pixels have no effect on the destination.

- pattern transparency mode

  Pattern transparency mode is the transparency or opaqueness of the white pixels in the pattern. When set to 0 (transparent), these pixels have no effect on the destination; when set to 1 (opaque), they are applied through the black pixels of the source pattern to the destination.

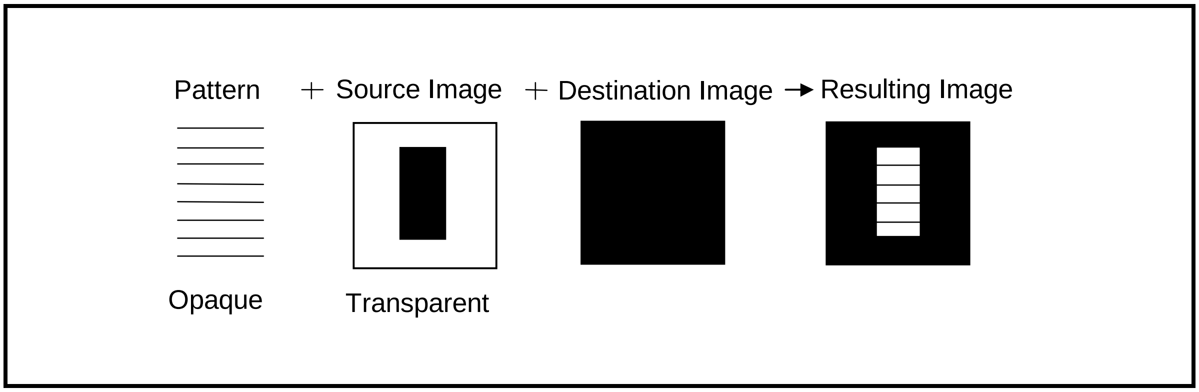


Figure 9-11 Print Model Imaging

Select Source Transparency Mode

ESC*v0N (transparent)
Decimal 027 042 118 048 078
Hex: 1B 2A 76 30 42

ESC*v1N (opaque)
Decimal 027 042 118 049 078
Hex: 1B 2A 76 31 42

This command sets the source image's transparency mode to transparent or opaque. A transparency mode of "0" (transparent) means that the white regions of the source image are not copied onto the destination. A transparency mode of "1" (opaque) means that the white pixels in the source image are applied directly onto the destination.

Select Pattern Transparency Mode

ESC*v0O (transparent)
Decimal 027 042 118 048 079
Hex: 1B 2A 76 30 43

ESC*v1O (opaque)
Decimal 027 042 118 049 079
Hex: 1B 2A 76 31 43

This command sets the pattern's transparency mode to transparent or opaque. A transparency mode of "0" (transparent) means that the white regions of the pattern image are not copied onto the destination. A transparency mode of "1" (opaque) means that the white pixels in the pattern are applied directly onto the destination.

Select Pattern Command

ESC*v#T
Decimal: 027 042 118 ### 084 (### = 048 through 051)
Hex: 1B 2A 76 ## 54 (## = 30 through 32)

#: 0 = Solid black (default)
1 = Solid white
2 = HP-defined shading pattern
3 = HP-defined cross-hatch pattern

This command identifies the type of pattern to be applied onto the destination. Use this command for selecting or changing the current pattern.

VECTOR GRAPHICS

The following are vector graphics commands within the PCL context.

HPGL/2 Plot Horizontal Size

ESC*C#K
Decimal: 027 042 #..# 075
Hex: 1B 2A 63 #..# 4B

# = horizontal size in inches

Default = width of the currently selected picture frame
Range = 0 tp 32767 (valid to 4 decimal places)

This command specifies the horizontal size of the HPGL/2 drawing being imported.

The horizontal HPGL/2 plot size determines the horizontal scaling factor used to fit the drawing into the PCL Picture Frame. For example, if the horizontal HPGL/2 plot size is specified as 12 inches and the PCL Picture Frame width is 4 inches, the horizontal scaling factor would be 3:1; the horizontal component of the image would be reduced to one-third its original size to fit into the PCL Picture Frame.

A parameter value of zero or a reset, page length, paper size, or orientation command defaults theHPGL/2 plot size to the width of the currently selected picture frame. This results in no scaling.

HPGL/2 Plot Vertical Size

ESC*C#L
Decimal: 027 042 #..# 076
Hex: 1B 2A 63 #..# 4C

# = vertical size in inches

Default = width of the currently selected picture frame
Range = 0 tp 32767 (valid to 4 decimal places)

This command specifies the vertical size of the HPGL/2 drawing being imported.

The vertical HPGL/2 plot size determines the vertical scaling factor used to fit the drawing into the PCL Picture Frame. For example, if the vertical HPGL/2 plot size is specified as 7 inches and the PCL Picture Frame height is 14 inches, the vertical scaling factor would be 1:2; the vertical component of the image would be enlarged to twice its original size to fit into the PCL Picture Frame.

A parameter value of zero or a reset, page length, paper size, or orientation command defaults theHPGL/2 plot size to the height of the currently selected picture frame. This results in no scaling.

Set Picture Frame Anchor Point

ESC*c0T
Decimal: 027 042 099 048 084
Hex: 1B 2A 63 30 54

Default = 0
Range = 0

This command sets the location of the PCL Picture Frame anchor point to the PCL cursor position.

The position of the picture frame anchor point defines the location of the upper left corner of the PCL Picture Frame. The “upper left” refers to the corner for which X and Y coordinates are minimized when the print direction is 0.

A parameter value of 0 (ESC*c0T) specifies that the picture frame anchor point should be set to the cursor position. Sending a cursor move command prior to sending this command places the picture frame anchor in the desired location. All parameter values other than zero are ignored, but if you do not send a Set Picture Frame Anchor Point command, the printer defaults the anchor point to the left edge of the logical page and the default top margin.

NOTE: The print direction command does not affect the physical location of the anchor point or the picture frame.

Using this command defaults the location of P1 and P2, resets the soft-clip window to the PCL Picture Frame boundaries, clears the polygon buffer, and updates the HPGL/2 pen position to the lower left corner of the picture frame (if entered with ESC%0B) as viewed from the current orientation.

Picture Frame Horizontal Size (Decipoints)

ESC*C#X
Decimal: 027 042 99 #..# 088
Hex: 1B 2A 63 #..# 58

# = horizontal size in decipoints (1/720th inch)

Default = width of the current logical page
Range = 0 tp 32767 (valid to 4 decimal places)

This PCL command specifies the horizontal dimension of the window to be used for printing an HPGL/2 plot.

NOTE: The horizontal dimension specified is parallel to the PCL X-axis when the print direction is set to 0 degrees (the default).

Using this command defaults the location of P1 to the lower left corner of the picture frame (and P2 to the upper right corner of the picture frame). This command also resets the soft-clip window to the PCL Picture Frame boundaries, clears the polygon buffer, and updates the HPGL/2 pen position to the lower left corner of the picture frame (P1) as viewed from the current orientation.

If no horizontal picture frame frame size command is used, the printer defaults the picture frame size to the logical page width. A parameter of 0 or the PCL reset, UEL, page length, paper size, or orientation commands default the horizontal picture frame size.

If an HPGL/2 plot size is specified, the horizontal picture frame size is used to determine the horizontal scaling factor used for scaling the image to fit in the picture frame.

Picture Frame Vertical Size (Decipoints)

ESC*C#Y
Decimal: 027 042 99 #..# 089
Hex: 1B 2A 63 #..# 59

# = vertical size in decipoints (1/720th inch)

Default = distance between the default top and bottom margins (the default text length)
Range = 0 tp 32767 (valid to 4 decimal places)

This PCL command specifies the vertical dimension of the window to be used for printing an HPGL/2 plot.

NOTE: The vertical dimension specified is parallel to the PCL Y-axis when the print direction is set to 0 degrees (the default).

HPGL/2 GRAPHICS

The IntelliBar provides the ability to print vector graphics using the HPGL/2 graphics language. HPGL/2 graphics may be created within application software or imported from existing applications. For various types of images, it is advantageous to use HPGL/2 vector graphics instead of the raster graphics described previously in this section. The advantages include faster I/O transfer of large images and smaller disk storage requirements.

Printing with HLGL/2 requires leaving the PCL printer language mode and entering HPGL/2 mode. Switching between modes involves only a few commands, and software applications may easily switch between the two modes as needed.

Enter HPGL/2 Mode

ESC %#B
Decimal: 027 037 ### 066 (### = 048 through 049)
Hex: 1B 25 ## 42 (## = 30 through 31)

#: 0 = Position pen at previous HPGL/2 pen position
1 = Position pen at current PCL cursor position

This command causes the printer to interpret subsequent commands as HPGL/2 commands, instead of PCL language commands. As soon as the printer receives this command, it switches to HPGL/2 mode, interpreting commands as HPGL/2 commands until it receives an Enter PCL Mode command, a Reset (ESC E) command, or until power is turned off and on.

Enter PCL Mode

ESC%0A (use previous PCL cursor position)
Decimal: 027 037 048 65
Hex: 1B 25 30 41

ESC%1A (use current HPGL/2 pen position for cursor position)
Decimal: 027 037 049 65
Hex: 1B 25 31 41

This command causes the printer to return to PCL mode from HPGL/2 mode. Sending this command causes the printer to stop interpreting the incoming data as HPGL/2 commands and to begin interpreting the data as PCL commands.

Line and Fill Attributes Group

The HPGL/2 commands described below allow you to

- enhance your drawings with various line types
- enhance your drawing with various fill types
- position fill type patterns

Line Type (LT)

LT *line type*[,*pattern length*[,*mode;*] or LT [;] or LT99 [;]

This command specifies the line pattern to be used when drawing lines. Use LT to vary lines and enhance your plot. Note that the ends of dashed line segments in a line pattern are affected by current line attributes (see the LA command).

Table 9-1 Line Type Command Parameters

Parameter	Format	Functional Range	Default
line type (fixed or adaptive)	clamped integer	-8 to 8	solid line
		99	restores previous line type
pattern length	clamped real	> 0	4% of the distance between scaling points P1 and P2
mode (relative or absolute)	clamped integer	0 or 1	0 (relative)

The LT command applies to lines drawn by the AA, AR, AT, CI, EA, ER, EW, FP, PA, PD, PE, PR, RA, RR, RT, and WG commands. Line types are drawn using the current line attributes set by the line attribute (LA) command. For example, if you have used LA to specify rounded ends, the printer draws each dash in a dashed line pattern with rounded ends.

If no parameters are entered, the LT command defaults the line type to solid and saves the previous line type, pattern length, and any unused portion of the pattern (residue). LT99 restores the previous line type (and residue if it is a fixed-line type).

Line Attributes (LA)

LA *kind, value*[,*kind,value*[,*kind,value;*]] or LA [*;*]

This command specifies how line ends and line joins are physically shaped. Use this command when drawing lines thicker than 0.35 mm.

Table 9-2 Line Attributes Command Parameters

Parameter	Format	Functional Range	Default
kind	clamped integer	1 through 3	1
value	clamped integer	Kind 1: 1 - 4	1 (butt)
	clamped integer	Kind 2: 1 - 6	1 (mitered)
mode	clamped real	Kind 3: 1 to 32,767	5

There are three line attributes: *line ends*, *line joins*, and the *miter limit*. The LA command parameters are used in pairs: the first parameter, *kind*, selects a line attribute; the second, *value*, defines the appearance of that attribute. The printer uses the current line attribute when the optional parameter pairs are omitted.

No parameters - Defaults the line attributes to butt ends, mitered joins, and a miter limit of 5. Equivalent to (LA1,1,2,1,3,5).

Kind - Specifies the line attribute for which you are setting a value.

Value - Defines the characteristics of the attribute specified by the kind parameter.

Number of Pens (NP)

NP [*n*;]

n: 2, 4, or 8

This command is not supported in the Intellibar.

Pen Width (PW)

PW *width*[,*pen;*] or PW [*;*]

This command specifies a new width for the logical pen. Subsequent lines are drawn in this new width. Use the PW command to vary your lines and enhance your drawings. Pen width can be specified as a fixed value or relative to the distance between P1 and P2. The pen width units are selected by means of the WU command (the default is metric - millimeters).

Table 9-3 Pen Width Command Parameters

Parameter	Format	Functional Range	Default
width	clamped real	-32768 to 32767	Dependent*
pen	integer	0 or 1	1 (black)

*Dependent on the mode set by the Pen Width Unit Selection (WU) command; if mode is metric, default width is 0.35 mm, if mode is relative, default width is 0.1% of the diagonal distance from P1 to P2.

Pen Width Unit Selection (WU)

WU *type[;]*] or WU [*;*]

This command specifies how the width parameter of the pen width (PW) command is interpreted (whether metric or relative units).

Table 9-4 Pen Width Unit Selection Command Parameters

Parameter	Format	Functional Range	Default
type	clamped integer	0 to 1	0 (metric)

Since using WU, with or without parameters, defaults all pen widths, send the WU command *before* a PW command (which sets a new pen width).

No parameters - Defaults type parameter to 0 (metric) and all pen widths to 0.35 mm.

Type - Specifies how the width parameter of the pen width (PW) command is interpreted.

0: Metric. Interprets the pen width parameter in millimeters. Specifying type 0 defaults all pen widths to 0.35 mm.

1: Relative. Interprets the pen width parameter as a percentage of the diagonal distance between P1 and P2. Specifying type 1 defaults all pen widths to 0.1% of the diagonal distance from P1 to P2.

If the specified type parameter is not 0 or 1, the printer ignores the command.

Select Pen (SP)

SP *pen number*[;] or SP [;]

This command selects the printer's "logical pen" for subsequent plotting. An SP command must be included at the beginning of each command sequence to enable the printer to draw.

Table 9-5 Select Pen Command Parameters

Parameter	Format	Functional Range	Default
pen number	integer	0 to 1	No pen

Although your printer does not have physical pens, for the purpose of compatibility it has a "logical" pen which you must select to print your drawing.

No parameters - Cancels pen selection; subsequent plotting commands are not drawn. Equivalent to (SP0).

Pen number - Selects the printer's "logical" pen. The printer will not draw unless and SP is sent.

0: Selects the white pen. To see a white pen on a non-white background, you must set the transparency mode to off (TR0;).

1: Selects the black pen; numbers greater than 1 are also interpreted as 1.

Use the pen width (PW) command to change the line width. You may change line widths as often as you like without sending an SP command again.

NOTE: If you are not using the transparency mode (TR) command, white is always transparent. For more information on the transparency mode command, see the TR command later in this section.

Symbol Mode (SM)

SM *character*[*;*] or SM [*;*]

This command draws the specified symbol at each X, Y coordinate point using the PA, PD, PE, PR, and PU commands. Use the SM command to create scattergrams, indicate points on geometric drawings, and differentiate data points on multiline graphs.

Table 9-6 Symbol Mode Command Parameters

Parameter	Format	Functional Range	Default
character	label (text)	Most printing characters (decimal codes 33 to 58, 60 to 126, 161, and 254)	–

The SM commands draws the specified symbol at each X, Y coordinate point for subsequent PA, PD, PE, PR, and PU commands. The SM command includes an automatic pen down; after the symbol is drawn, the pen position and any dashed-line residue are restored.

No parameter - Terminates symbol mode.

Character - Draws the specified character centered at each subsequent X, Y coordinate. The symbol is drawn in addition to the usual function of each HPGL/2 command.

The character is drawn in the font selected at the time the vectors are drawn. If you change to a new symbol set, the character changes to the corresponding character from the new symbol set, The size (SI and SR), slant (SL), and direction (DI and DR) commands affect how the character is drawn. Specifying a non-printing character cancels symbol mode.

An SM command remains in effect until another SM command is executed or the printer is initialized or set to the default conditions.

Fill Type (FT)

FT *fill type*[,*option1*[,*option2;*]] or FT [*;*]

This command selects the shading pattern used to fill polygons (FP), rectangles (RA or RR), wedges (WG), or characters (CF). Use the FT command to enhance drawings using solid fill, shaded fill, parallel lines (hatching), cross-hatch, patterned (raster) fill, or PCL user-defined patterns.

Table 9-7 Fill Type Command Parameters

Parameter	Format	Functional Range	Default
fill type	clamped integer	1 - 4, 10, 11, 21, 22	1
option1, option2	clamped real	type dependent	type dependent

There are eight forms of fill types. The type parameter tells the printer which form you are using. If the fill type is specified but the option1 and/or option2 parameter is omitted, values previously given for the specified fill type are assumed, or the defaults are assumed if none is specified.

No parameters - Defaults all FT parameters and sets the fill type to solid fill. Equivalent to (FT1).

Type - Selects the fill pattern (see Table J-8).

Option1, Option2 - The definition of these optional parameters depends on the type of fill selected. Table 9-8 lists the options available for each fill type.

Table 9-8 Fill Patterns and Options

Fill Type	Description	Option1	Option2
1 and 2	Solid black	Ignored	Ignored
3	Hatching (parallel lines)	Spacing of lines	Angle of lines
4	Cross-hatch	Spacing of lines	Angle of lines
10	Shading	Shading level	Ignored
11	HPGL/2 user-defined	Raster-fill index	Ignored
21	PCL cross-hatch patterns	Pattern type	Ignored
22	PCL user-defined	Pattern ID	Ignored

Anchor (AC)

AC *X,Y*[*;*] or AC [*;*]

This command positions the starting point of any fill pattern. Use AC to ensure that the selected fill pattern is positioned as expected within the figure.

Table 9-9 Anchor Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates	currents units	-2^{30} to $2^{30} - 1$	None

The ‘anchor corner’ is the point at which any fill pattern starts. Setting the anchor corner guarantees that a corner point of the selected fill pattern is at the specified coordinate, aligned vertically and horizontally.

No parameters - Defaults the anchor corner to the lower-left corner of the PCL Picture Frame (relative to the current coordinate system). Equivalent to (AC0,0).

X, Y coordinates - The coordinate position defines the position of the starting point for any fill pattern.

Raster Fill Definition (RF)

RF *index,width,height,pen number*[, ... *pen number,*] or RF *index*[;] or RF [*;*]

This command defines a rectangular pattern that may be used as area fill and for screened vectors (see the SV command) later in this section. Use the RF command to create your own fill types and screen patterns.

Table 9-10 Raster Fill Definition Command Parameters

Parameter	Format	Functional Range	Default
index	clamped integer	1 to 8	1 (solid)
width	clamped integer	1 to 255	–
height	clamped integer	1 to 255	–
pen number	integer	0 or 1	–

The RF command does not *select* a fill type; use the fill type (FT) command with a type parameter of 11 and the corresponding raster fill index number for the second parameter (for example, [FT11,3]) for an index number of 3.

No parameters - Defaults all raster fill patterns to solid fill.

Index - Specifies the index number of the pattern being defined. Eight patterns can exist concurrently. When you send RF with an index parameter only (Rfn), the corresponding pattern is defaulted to solid fill.

Width, Height - Specifies the width and height (in pixels) of the pattern being defined.

Pen number - Represents a pixel in the pattern being defined and indicates its color (black or white)

0: white

>0: black

The pen number parameter defines pixels left to right, top to bottom. The total number of pen number parameters should be equal to the width time height parameters. For example, to define a pattern that is 8 x 16 pixels, you need 128 pen number parameters. If you do not include enough pen number parameters, the rest of the pixels are assumed to be white (zero). Patterns are printed in rows parallel to the plotter-unit X-axis.

User Defined Line Type (UL)

UL *index* [,*gap1*, ... ,*gap20*;] or UL [;]

This command creates line types by specifying gap patterns that define the lengths of spaces and lines comprising a line type.

Table 9-11 User-Defined Line Type Command Parameters

Parameter	**Format**	**Functional Range**	**Default**
index	clamped integer	1 through 8	–
gaps	clamped real	0 through 32767	Default line types

The UL command allows you to define and store your own line types, This command does not itself select a line type. Use the line type (LT) command to select the line type once you have defined it with the UL command.

No parameters - Defaults all line types (refer to the LT command in this section).

Index - Identifies the number of the line type to be defined. Specifying an index number without gap parameters sets the line type identified by the index to the default pattern for that number. The index number may not be 0.

The index parameter uses absolute values; therefore, (UL-n) is the same as (ULn). Redefining a standard fixed line type automatically redefines the corresponding adaptive line type.

Gaps - Specifies alternate pen-down and pen-up stretches in the line type pattern; if gaps are numbered starting with 1, odd numbered gaps are pen-down moves, even numbered gaps are pen-up moves. The first gap is a pen-down move. Gap values are converted to percentages of the LT command's pattern length parameter. A maximum of 20 gaps are allowed for each user-defined line type.

Screened Vectors (SV)

SV [*screen_type*[,*shading*[,*index*]]][;] or SV [;]

screen_type: selects the types of screening as follows

0 - No screening
1 - Shaded fill
2 - HPGL/2 user-defined raster fill (RF command)
21- Pre-defined PCL cross-hatch patterns
22 - PCL user-defined raster fill (RF command)

This command selects the type of screening area (area fill) to be applied to vectors. Options include lines, hatching patterns (fill types 3 and 4), arcs, circles, edges of polygons, rectangles, wedges, and PCL user-defined patterns. The SV command does not affect solid fill types, stroked characters, or edges of characters.

There are four types of screen fill: shaded fill, HPGL/2 user-defined raster fill, pre-defined PCL cross-hatch patterns, and PCL user-defined patterns.

The SV command defaults to no parameters (no screening or solid fill, which is equivalent to SV0;). All parameters are optional. If all parameters are omitted, screening is turned off (the vectors are solid).

Table 9-12 Screened Vector Parameters

Parameter	Format	Functional Range	Default
screen_type	clamped integer	0, 1, 2, 21, 22	No screening (solid)
shading	clamped integer	type dependent	type dependent

For type 1 (shaded fill), specify the shading percentage using a number from 0 to 100. For example, to print vectors that are shaded 15%, specify (SV1,15;).

For type 2, (HPGL/2 user-defined raster fill), specify the index number of the fill pattern using the RF (raster fill definition) command.

Transparency Mode (TR)

TR [*n*][;] or TR [;]

n: 0 - Transparency mode is off
1 - Transparency mode is on
No parameter - Defaults to transparency mode = on. Equivalent to (TR1;).

This command defines how the white areas of the source graphics image affect the destination graphics image.

Table 9-13 Transparency Mode Parameters

Parameter	Format	Functional Range	Default
n	clamped integer	0 or 1	1 (on)

When transparency mode is on (default), the portion of a source image which is defined by white pixels does not affect the destination; whatever was already written to the page “shows through” the white areas in the new image.

When transparency mode is off, all source pixels are written to the destination, obscuring any underlying images.

The transparency mode is defaulted by the ESC E (reset), IN (initialize), and DF (default values) commands.

Vector Group Commands

The HPGL/2 vector group commands described below enable you to achieve the following results in your programs

- use absolute and relative coordinates when plotting
- draw lines and arcs
- encode coordinates to increase your printer's throughput.

Arc Absolute (AA)

AA $X_{center,}Y_{center,}$ *sweep angle*[, *chord angle*;]

This command draws an arc using absolute coordinates which starts at the current pen location and pivots around the current center point.

Table 9-14 Arc Absolute Command Parameters

Parameter	Format	Functional Range	Default
$X_{center,}$ Y_{center}	currents units	-2^{30} to 2^{30} - 1	None
sweep angle	clamped real	-32768 to 32767	None
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The AA command draws an arc starting at the current pen location using the current pen up/down status, line type, and attributes. After drawing the arc, the pen location remains at the end of the arc.

Arc Relative (AR)

AR $X_{increment}$,$Y_{incremententent}$, *sweep angle*[, *chord angle*;]

This command draws an arc using relative coordinates which starts at the current pen location and pivots around the current center point.

Table 9-15 Arc Relative Command Parameters

Parameter	Format	Functional Range	Default
$X_{increment}$, $Y_{increment}$	currents units	-2^{30} to $2^{30} - 1$	None
sweep angle	clamped real	-32768 to 32767	None
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The AR command draws an arc starting at the current pen location using the current pen up/down status, line type, and attributes. After drawing the arc, the pen location remains at the end of the arc.

Absolute Arc Three Point (AT)

AT X_{inter},Y_{inter}, X_{end},Y_{end}[, *chord angle*;]

This command draws an arc segment using absolute coordinates from a starting point, through an intermediate point, to an end point. Use the AT command when you know these three points of arc.

Table 9-16 Absolute Arc Three Point Command Parameters

Parameter	Format	Functional Range	Default
X_{inter}, Y_{inter}	currents units	-2^{30} to $2^{30} - 1$	None
X_{end}, Y_{end}	current units	-32768 to 32767	None
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The AT command uses the current pen location and two specified points to calculate a circle and draw the appropriate arc segment of its circumference. The arc starts at the current pen location using the current pen, line type, line attributes, and pen up/down status. You specify the intermediate and end points. After drawing the arc, the pen location remains at the end of the arc.

Plot Absolute (PA)

PA *X*,*Y* [, ... *;*] or PA[*;*]

This command establishes absolute plotting and moves the pen to the specified absolute coordinates from the current pen position.

Table 9-17 Plot Absolute Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates	currents units	-2^{30} to $2^{30} - 1$	None

The printer interprets the parameters as follows:

No parameters - Establishes absolute plotting for subsequent commands.

X, Y coordinates - Specifies the absolute location to which the pen moves. When you include more than one coordinate pair, the pen moves to each point in the order given using the current pen up/down status. If the pen is up, PA moves the pen to the point; if the pen is down, PA draws a line to the point. Lines are drawn using the current line width, type, and attributes.

Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

Plot Relative (PR)

PR *X*,*Y* [, ... *;*] or PR[*;*]

This command establishes relative plotting and moves the pen to the specified points, with each move relative to the current pen position.

Table 9-18 Plot Relative Command Parameters

Parameter	Format	Functional Range	Default
X, Y (increments)	currents units	-2^{30} to $2^{30} - 1$	None

The PR (and PE) command has extended ranges of -2^{30} to $2^{30} - 1$ plotter units. If the current pen position goes out of this range, the printer ignores HPGL/2 commands until it receives an absolute PA (or PE) coordinate within the extended range.

The printer interprets the parameters as follows:

No parameters - Defaults to relative plotting mode for subsequent commands.

X, Y coordinates - Specifies incremental moves relative to the current pen location. When you include more than one relative coordinate pair, the pen moves to each point in the order given (relative to the previous point) using the current pen up/down status. If the pen is up, PR moves the pen to the point; if the pen is down, PR draws a line to the point. Lines are drawn using the current line width, type, and attributes.

Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

Pen Down (PD)

PD *X*,*Y* [... ;] or PD [;]

This command lowers the printer's "logical pen" and draws subsequent graphics commands.

Table 9-19 Pen Down Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates/increments	currents units	-2^{30} to $2^{30} - 1$	None

This command emulates a pen plotter that must lower the pen to draw lines on the page.

The printer interprets the parameters as follows:

No parameters - Prepares the printer to draw subsequent graphics commands.

X, Y coordinates - Draws (in current units) to the point specified. You can specify as many X, Y coordinates as you want. When you include more than one coordinate pair, the printer draws each point in the order given.

Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

Whether the PD command uses coordinates or increments depends on the most recently executed PA or PR command. If no PA or PR command is issued, absolute plotting (PA) is used.

Pen Up (PU)

PU *X*,*Y* [... ;] or PU [;]

This command moves to subsequent points without drawing. Use the PU command to move to another location without drawing a connecting line.

Table 9-20 Pen Up Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates/increments	currents units	-2^{30} to $2^{30} - 1$	None

This command emulates a pen plotter that must raise the pen to prevent drawing stray lines on the page.

The printer interprets the parameters as follows:

No parameters - Prevents drawing subsequent graphics commands (unless the command contains an automatic pen down).

X, Y coordinates - Moves to the point(s) specified. You can specify as many X,Y coordinates as you want. When you include more than one coordinate pair, the printer moves to each point in the order given.

Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

Whether the PU command uses absolute coordinates or relative coordinates (increments) depends on the most recently executed PA or PR command. If no PA or PR command is issued, absolute plotting (PA) is used.

Relative Arc Three Point (RT)

RT $X_{incr\ inter,}Y_{incr\ inter,}$ $\mathrm{X}_{incr\ end}$ $\mathrm{Y}_{incr\ end}$[, *chord angle*;]

This command draws an arc segment using relative coordinates from a starting point, through an intermediate point, to an end point. Use the RT command when you know these three points of arc.

Table 9-21 Relative Arc Three Point Command Parameters

Parameter	Format	Functional Range	Default
$\mathrm{X}_{\mathrm{incr\ inter,}}$ $\mathrm{Y}_{\mathrm{incr\ inter}}$	currents units	-2^{30} to $2^{30} - 1$	None
$\mathrm{X}_{\mathrm{incr\ end,}}$ $\mathrm{Y}_{\mathrm{incr\ end}}$	current units	-32768 to 32767	None
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The RT command uses the current pen location and two specified points to calculate a circle and draw the appropriate arc segment of its circumference. The arc starts at the current pen location using the current pen, line type, line attributes, and pen up/down status. You specify the intermediate and end points. After drawing the arc, the pen location remains at the end of the arc.

Polyline Encoded (PE)

PE [*flag*[*value*]]*coord_pair* ... [*flag*[*value*]]*coord_pair* ; or PE;

This command incorporates the PA, PR, PU, PD, and SP commands into an encrypted format that substantially decreases the size of your file and the time required for data transmission. (This command is especially useful when using the RS-232C interface.)

Table 9-22 Polyline Encoded Command Parameters

Parameter	Format	Functional Range	Default
flag	character	‘:’, ‘<‘, ‘>‘, ‘=‘, or ‘7’	None
value	character	flag dependent	
coordinate pair	character	-2^{30} to 2^{30} - 1	None

NOTE: Parameter values are self-terminating. Do not use commas with this command. Also, you must use a semicolon to terminate the PE command.

No parameters - Updates the Carriage Return point. The PE command without parameters does not affect the pen’s current location or up/down status.

Flag - Indicates how the printer interprets subsequent values. Flags are ASCII characters and are not encoded. The printer disregards the eighth bit of a flag.

Flag	Meaning
:	Select pen
<	Pen up
>	Fractional data
=	Absolute
7	7-bit mode

NOTE: Because the select pen (SP) command is not allowed in ploygon mode, if you select a pen with PE while in polygon mode, the SP command is ignored (see the following subsection for polygon group commands).

Value - Specifies data according to the preceding flag. For example, a value following a select-pen flag should be a pen number.

Table 9-23 Polyline Encoded Command Value Parameters

Value	Format	Range
pen number	integer	0 to 1
number of fractional binary bits	integer	-26 to 26

X, Y coordinates - Specifies a coordinate pair encoded into a base 64 (default) or base 32 equivalent. Use base 64 if the system sends 8 bits of data without parity; use 7-bit mode and base 32 coordinate values if the system requires a parity bit.

Polygon Group Commands

The polygon group commands described below use the polygon buffer, which is a temporary data storage area in printer memory. The polygon group commands enable you to achieve the following results in your program:

- draw circles, wedges, and rectangles
- use polygon mode for drawing polygons, subpolygons, and circles.

Circle (CI)

CI radius[,*chord angle*;]

This command draws the circumference of a circle using the specified radius and chord angle. If you want a filled circle, refer to the WG or PM commands described later in this section.

Table 9-24 Circle Command Parameters

Parameter	Format	Functional Range	Default
radius	currents units	-2^{30} to $2^{30} - 1$	None
chord angle	clamped real	0.5° to 180°	5°

The CI command includes an automatic pen down. When a CI command is received, the pen lifts, moves from the center of the circle (the current pen location) to the starting point on the circumference, lowers the pen, draws the circle, then returns with the pen up to the center of the circle. After the circle is drawn, the previous pen up/down status is restored. To avoid leaving a dot at the center of the circle, move to and from the circle’s center with the pen up

Radius - Measured from the current pen location. Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

Chord angle - Specifies the chord angle used to draw the arc. The default chord angle is 5°. The chord angle specifies, in degrees, the maximum angle created when lines from each of the chord intersect the center point of the circle. The smaller the chord angle, the smoother the curve.

Edge Rectangle Absolute (EA)

EA *X,Y*[*;*]

This command defines and outlines a rectangle using absolute coordinates. Use the EA command when drawing charts or schematic diagrams that require rectangles.

Table 9-25 Edge Rectangle Absolute Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates	currents units	-2^{30} to 2^{30} - 1	None

The EA command defines the edges of a rectangle using absolute coordinates and the current pen, line type, and line attributes. The EA command performs an automatic pen down. When command execution completes, the original pen location and up/down status are restored.

The X, Y coordinates specify the opposite corner of the rectangle from the current pen location. The current pen location is the starting point of a rectangle. Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

The only difference between the EA command and the RA (fill rectangle absolute) command is that the EA command produces an outlined rectangle; RA produces a filled rectangle.

Fill Rectangle Absolute (RA)

RA *X,Y*[*;*]

This command defines and fills a rectangle using absolute coordinates. Use the RA command to fill rectangular shapes in drawings. (To outline a rectangle using absolute coordinates, use the EA command).

Table 9-26 Fill Rectangle Absolute Command Parameters

Parameter	Format	Functional Range	Default
X, Y coordinates	currents units	-2^{30} to 2^{30} - 1	None

The RA command defines and fills a rectangle using absolute X, Y coordinates and the current pen, current line, and current fill types. The RA command performs an automatic pen down. When command execution completes, the original pen location and up/down status are restored.

The X, Y coordinates specify the opposite corner of the rectangle from the current pen location. The current pen location is the starting point of a rectangle. Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

The only difference between the RA command and the EA (edge rectangle absolute) command is that the RA command produces a filled rectangle; EA produces an outlined rectangle.

Fill Rectangle Relative (RR)

RR *X*,*Y*[*;*]

This command defines and fills a rectangle using relative coordinates (increments). Use the RA command to fill rectangular shapes in drawings. (To outline a rectangle using relative coordinates, use the ER command).

Table 9-27 Fill Rectangle Relative Command Parameters

Parameter	**Format**	**Functional Range**	**Default**
X, Y increments	currents units	-2^{30} to $2^{30} - 1$	None

The RR command defines and fills a rectangle using relative X, Y coordinates and the current pen, current line, and current fill types. The RR command performs an automatic pen down. When command execution completes, the original pen location and up/down status are restored.

The X, Y coordinates specify the opposite corner of the rectangle from the current pen location. The current pen location is the starting point of a rectangle. Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

The only difference between the RR command and the ER (edge rectangle relative) command is that the RR command produces a filled rectangle; ER produces an outlined rectangle.

Edge Rectangle Relative (ER)

ER *X*,*Y*[*;*]

This command defines and outlines a rectangle using relative coordinates (increments). Use the ER command when drawing charts or schematic diagrams that require rectangles.

Table 9-28 Edge Rectangle Relative Command Parameters

Parameter	Format	Functional Range	Default
X, Y increments	currents units	-2^{30} to $2^{30} - 1$	None

The ER command defines and edges a rectangle using relative X, Y coordinates and the current pen, line type, and line attributes. The ER command performs an automatic pen down. When command execution completes, the original pen location and up/down status are restored.

The X, Y coordinates specify the opposite corner of the rectangle from the current pen location. The current pen location is the starting point of a rectangle. Coordinates are interpreted in current units: as user-units when scaling is on; as plotter units when scaling is off.

The only difference between the ER command and the RR (fill rectangle relative) command is that the ER command produces an outlined rectangle; RR produces a filled rectangle.

Fill Wedge (WG)

This command defines and fills any wedge. Use the WG command to draw filled sections of a pie chart.

WG *radius,start angle,sweep angle*[,*chord angle;*]

Table 9-29 Fill Wedge Command Parameters

Parameter	Format	Functional Range	Default
radius	currents unit	-2^{30} to $2^{30} - 1$	–
start angle	clamped real	-32768 to 32767	–
sweep angle	clamped real	$\pm 360^{o}$	–
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The WG command defines and fills a wedge using the current pen, fill type, and line types. The WG command includes an automatic pen down. When the operation completes, the original pen location and up/down status are restored.

The only difference between the WG command and the EW (edge wedge) command is that the WG command produces a filled wedge, and the EW command produces an outlined wedge.

Edge Wedge (EW)

This command outlines any wedge. Use the EW command to draw sections of pie charts.

EW *radius,start angle,sweep angle*[,*chord angle*;]

Table 9-30 Edge Wedge Command Parameters

Parameter	Format	Functional Range	Default
radius	currents unit	-2^{30} to $2^{30} - 1$	None
start angle	clamped real	-32768 to 32767	None
sweep angle	clamped real	$\underline{+}\ 360^{o}$	None
chord angle	clamped real	0.5^{o} to 180^{o}	5^{o}

The EW command defines and edges a wedge using the current pen, fill type, and line types. The EW command includes an automatic pen down. When the operation completes, the original pen location and up/down status are restored.

The only difference between the EW command and the WG (fill wedge) command is that the EW command produces an outlined wedge, and the WG command produces a filled wedge.

Polygon Mode (PM)

This command enters polygon mode for defining shapes, such as block letters or any unique area, and exits for subsequent filling and/or edging. Fill polygons using the fill polygon (FP) command; outline polygons using the edge polygon (EP) command.

PM *polygon definition* [*;*] or PM[*;*]

Table 9-31 Polygon Mode Command Parameters

Parameter	Format	Functional Range	Default
polygon definition	clamped integer	0, 1, and 2	0

In polygon mode, you define the area of the polygon(s) using graphics commands. These commands (and associated X, Y coordinates) are stored in the polygon buffer. The polygon is not printed until you exit polygon mode and fill and/or outline the area.

No parameters - Clears the polygon buffer and enters polygon mode. Equivalent to (PM0).

Polygon definition - Defines polygon mode status as follows:

0: Clears the polygon buffer and enters polygon mode

1: Closes the current polygon (or subpolygon) and remains in polygon mode; all commands sent following PM1 but before PM2 (or the next PM1) are stored as one subpolygon.

2: Closes the current polygon (or subpolygon) and exits polygon mode.

Fill Polygon (FP)

This command fills the polygon currently in the polygon buffer. Use the FP command to fill polygons defined in polygon mode or with edge rectangle or edge wedge commands (ER, EW, RA, RR, or WG).

FP *fill method* [;] or FP [;]

Table 9-32 Fill Polygon Command Parameters

Parameter	Format	Functional Range	Default
polygon definition	clamped integer	0 or 1	0 (odd-even fill)

Fill method - Specifies the algorithm used to determine which portions of a polygon are “inside” the polygon and therefore are to be filled:

0: Even/odd fill algorithm (default)

1: Non-zero winding fill algorithm

The FP command fills any polygon that is currently in the polygon buffer. FP accesses the data in the polygon buffer but does not clear the buffer or change the data in any way.

The FP command fills between points defined with either the pen down or the pen up. The polygon is filled using the current pen, fill type, line type, and attributes (if the fill type is not raster). The FP command includes an automatic pen down. When command execution completes, the original pen location and up/down status are restored.

Edge Polygon (EP)

This command outlines the polygon currently stored in the polygon buffer. Use the EP command to edge polygons defined in polygon mode or with fill rectangle and wedge commands (RA, RR, and WG).

```
EP [;]
```

The EP command fills any polygon that is currently in the polygon buffer. FP accesses the data in the polygon buffer. This includes wedges and rectangles defined using the EA, ER, EW, RA, RR, and WG commands.

EP accesses the data in the polygon buffer but does not clear the buffer or change the data in any way.

The EP command only edges between points defined with the pen down. When command execution completes, the original pen location and up/down status are restored.

Configuration and Status Group

The configuration and status group commands help you:

- establish default conditions and values for HPGL/2 features
- scale images in the dimensional units you want to use
- enlarge/reduce images for different media sizes
- establish a window (soft-clip limits)
- draw equal-sized and mirror-imaged drawings
- rotate the HPGL/2 coordinate system
- add comments to your HPGL/2 command sequence.

Scale (SC)

This command establishes a user-unit coordinate system by mapping user-defined coordinate values onto the scaling points P1 and P2. The SC command determines the number of user-units along the X- and Y-axes between P1 and P2. The actual size of the units depends on the locations of P1 and P2 and the range of user-units set up by the SC command.

SC $X_{MIN}X_{MAX}Y_{MIN}Y_{MAX}$ [,*type*[,*left*,*bottom*;]] or
SC $X_{MIN}X_{FACTOR}Y_{MIN}Y_{FACTOR}$ [,*type*[;] or
SC [;]

Table 9-33 Scale Command Parameters

Parameter	Format	Functional Range	Default
$X_{MIN}X_{MAX}$,	real	-2^{30} to $2^{30} - 1$	None
$Y_{MIN}Y_{MAX}$	real	-2^{30} to $2^{30} - 1$	None
type	clamped integer	0, 1, or 2	0
left	clamped real	0 to 100%	50%
bottom	clamped real	0 to 100%	50%
$X_{FACTOR,}$ $Y_{\ FACTOR}$	real	-2^{30} to $2^{30} - 1$	None

Scaling allows you to establish units of measure with which you are familiar or which are more logical to your drawing. There are three types of scaling: anisotropic (type 0), isotropic (type 1), and point factor (type 2). The type parameter tells the printer which form you are using. If no parameters are entered, scaling is turned off, and subsequent coordinates are in plotter units.

Anisotropic scaling indicates that the size of the units along the X-axis may be different than the size of the units along the Y-axis. Isotropic scaling, then, indicates that the units are the same size on both axes. Point-factor scaling sets up a ratio of plotter units to user units.

The SC command does not change the locations of P1 and P2, only their coordinate values. Also, scaling is not limited to the rectangular area defined by P1 and P2, but extends across the entire printing area within the PCL Picture Frame.

Input Window (IW)

This command defines a rectangular area, or window, that establishes soft-clip limits. Subsequent HPGL/2 drawing is restricted to this area. Use the IW command to restrict printing to a specified area on the page.

IW $X_{LL}Y_{LL}X_{UR}Y_{UR}$[;] or IW [;]

Table 9-34 Input Window Command Parameters

Parameter	Format	Functional Range	Default
$X_{LL}Y_{LL}$, $X_{UR,}Y_{UR}$	current units	-2^{30} to $2^{30} - 1$	PCL Picture Frame

The printer interprets the command parameters as follows.

No parameters - Resets the soft-clip limits to the PCL Picture Frame limits.

X, Y coordinates - Specifies the opposite, diagonal corners of the window area, usually the lower left (LL) and upper right (UR) corners. Coordinates are interpreted in the current units: as user-units when scaling is on; as plotter units when scaling is off.

When scaling is on, subsequent changes to P1 and P2 move the window in relation to the physical page, but keep the same user coordinate locations. However, sending a subsequent SC command binds the window to its equivalent plotter units. The window does not change with any subsequent IP or IR commands.

When you turn on the printer, the window is automatically set to the PCL Picture Frame boundaries. You can define a window that extends beyond this picture frame; however, the printer cannot print vector graphics beyond this effective window. All programmed pen motion restricted to this area.

Input Relative P1 and P2 (IR)

This command establishes new or default locations for the scaling points P1 and P2 relative to the PCL Picture Frame size. P1 and P2 are used by the scale (SC) command to establish user-unit scaling. The IR command can also be used in advanced techniques such as printing mirror-images, enlarging or reducing drawings, enlarging/reducing relative character size, or changing text direction.

IRX_{P1},Y_{P1}[X_{P2},Y_{P2};] or IR [;]

Table 9-35 Input Relative P1 and P2 Command Parameters

Parameter	**Format**	**Functional Range**	**Default**
$X_{P1}Y_{P1}$ [X_{P2},Y_{P2}	clamped real	0 to 100%	0,0,100,100%

When P1 and P2 are set using the IR command, the scaled area is page-size independent. As the PCL Picture Frame changes size, P1 and P2 keep the same relative position within the PCL Picture Frame boundaries.

No parameters - Defaults P1 and P2 to the lower left and upper-right corners of the PCL Picture Frame, respectively.

X, Y coordinates - Specifies the location of P1 (and, optionally, P2) as percentages of the PCL Picture Frame limits (specifying P2 is not required). If P2 is not specified, P2 tracks P1; the P2 coordinates change so that the distances of X and Y between P1 and P2 remain the same. This tracking process can cause P2 to locate outside the effective window. Used carefully, however, the tracking function can be useful for preparing more than one equal-sized drawing on a page.

Neither X, Y coordinate of P1 can equal the corresponding coordinate of P2. If either coordinate of P1 equals the corresponding coordinate of P2, the coordinate of P2 is incremented by 1 plotter unit.

Default Values (DF)

This command returns the printer's HPGL/2 settings to the factory default paremters. Use the DF command to return the printer to a known state while maintaining the current locations of P1 and P2 (unlike the IN command described next in this section). When you use the DF command at the beginning of a command sequence, graphics parameters such as character size, slant, or scaling are defaulted.

DF[;]

The DF command resets the printer to the conditions listed in Table 9-36.

Table 9-36 Default Settings

Function	Command	Default Condition
Anchor Corner	AC	Anchor corner (not the same as the picture frame anchor point) set to lower-left corner of PCL Picture Frame, relative to the current coordinate system.
Alternate Font Direction	AD	Stick Font (11.5 point, 9 characters per inch, upright, medium)
Character Fill Mode	CF	Solid fill, no edging
Asolute Direction	DI1,0	Character direction parallel to X-axis
Define Label Terminator	DT	End of text (ETX) and non-printing mode
Define Variable Text Path	DV	Text printed left to right with normal Line Feed
Extra Space	ES	No extra space
Fill Type	FT	Solid fill
Input Window	IW	Set equal to PCL Picture Frame Window
Line Attributes	LA	Butt caps, mitered joins, and miter limit=5
Label Origin	LO1	Standard text printing starting at current location
Line Type	LT	Solid line, relative mode, pattern length = 4% of diagonal distance from P1 to P2
Plotting Mode	PA	Absolute plotting

Table 9-36 Default Settings (cont'd)

Function	Command	Default Condition
Polygon Mode	PM	Polygon buffer cleared
Raster Fill	RF	Solid black
Scalable or Bitmap Fonts	SB0	Scalable fonts only
Scale	SC	User-unit scaling off
Screened Vectors	SV	No screening
Standard Font Definition	SD	Stick Font (11.5 point, 9 characters per inch, upright, medium)
Absolute Character Size	SI	Turns off size transformation
Character Slant	SL	No slant
Symbol Mode	SM	Turns off symbol mode
Select Standard Font	SS	Standard font selected
Transparency Mode	TR1	Transparency mode on
Transparent Data	TD	Normal printing mode
User-defined line type	UL	Defaults all 8 line types

In addition, the printer updates the carriage return point for text printing to the current pen location.

The DF command does not affect the following HPGL/2 conditions:

- locations of P1 and P2
- current pen, its location, width, width unit selection, and up/down position
- HPGL/2 drawing rotation

Initialize (IN)

This command resets all programmable HPGL/2 functions to their default settings. Use the IN command to return the printer to a known HPGL/2 state and to cancel settings that may have been changed by a previous command sequence. (The ESC E reset command issues an automatic IN command.)

```
IN [;]
```

NOTE: Once HPGL/2 mode is entered and commands are issued, the HPGL/2 conditions are no longer initialized. To place HPGL/2 into the default state, send the IN command.

The IN command sets the printer to the same conditions as the default (DF) command, plus the following:

- raises the pen (PU)
- returns the pen location to the lower-left corner of the PCL Picture Frame (PA0,0)
- cancels drawing rotation (RO)
- sets P1 and P2 to the lower-left and upper-right corners, respectively, of the PCL Picture Frame (IP).
- sets pen width mode to metric; units are millimeters (WU)
- sets the pen width to 0.35 mm (PW)
- sets number of pens to 2 (black [1] and white [0]

Rotate Coordinate System (RO)

This command rotates the printer's coordinate system relative to the default HPGL/2 coordinate system in the following increments of rotation: 90^{o}, 180^{o}, and 127^{o}. Use the RO command to orient your drawing vertically or horizontally, or to reverse the orientation.

RO *angle*[;] or RO [;]

Table 9-37 Rotate Coordinate System Command Parameters

Parameter	Format	Functional Range	Default
Angle	clamped integer	0^{o}, 90^{o} , 180^{o}, or 270^{o}	0^{o}

The printer interprets the command parameters as follows:

No parameters - Defaults the orientation of the coordinate system to 0^{o}. Equivalent to (RO0). This is the same as the PCL's current orientation.

Angle - Specifies the degree of rotation:

0: Sets the orientation to PCL's current orientation

90: Rotates and shifts the coordinate system 90 degrees in a positive angle of rotation from PCL's current orientation.

180: Rotates and shifts the coordinate system 180 degrees in a positive angle of rotation from PCL's current orientation.

270: Rotates and shifts the coordinate system 270 degrees in a positive angle of rotation from PCL's current orientation.

Advance Full Page (PG)

This HPGL/2 command is ignored by the printer since it could cause undesirable results when importing plots. A page eject can be accomplished only from the PCL printer language mode.

The following PCL commands cause a conditional page eject, meaning that a page is ejected if there is any printable data in the print buffer:

- ESC E (reset)
- UEL (Universal Exit Language)
- Flush All Pages
- Page Length
- Page Size
- Orientation
- Paper Source

When a page is ejected using one of the above commands, the PCL cursor position is set to the top of form on the new page.

An alternative method of ejecting a page from PCL is the Form Feed control code. A Form Feed causes an unconditional page eject and advances the current active cursor position to the top of form on the next page. The horizontal cursor position remains the same as before the page eject.

NOTE: The HPGL/2 pen position is not affected by the Form Feed; it occupies the same position on the next page.

Replot (RP)

This command is ignored by the printer; to eject a page, the printer must be in the PCL printer language mode. The following commands cause a conditional page eject, meaning that a page is ejected if there is any printable data in the print buffer:

- ESC E (reset)
- UEL (Universal Exit Language)
- Flush All Pages
- Page Length
- Page Size
- Orientation
- Paper Source

The PCL Form Feed control code causes an unconditional page eject and advances the current active cursor position to the top of form on the next page.

A page eject caused by any of the above commands except paper source defaults the HPGL/2 pen position.

To print more than one plot, use the Number of Copies command described in Section 3.

[*] "This information is subject to change without notice. This information is provided "as is" without either express or implied warranty. IntelliTech International, Inc. disclaims any and all warranties with regard to this information. IntelliTech shall not be liable in any event for any special, indirect or consequential damages or any damages whatsoever resulting from loss of data, profits or use, for any reason or in any action, arising out of or in connection with the use or performance of this information. "