

# 6 Raster Graphics

## Introduction

A raster image is made up of a series of discrete picture elements—pixels. Pictures such as those in newspapers, television, and documents from Hewlett-Packard printers are examples of raster images. In comparison, pictures drawn with lines and geometric objects such as circles, rectangles, and polygons are termed vector graphic images. While some pictures could be produced using either vector or raster graphics, raster graphics is best suited for printing photographic images.

A raster image is invariably rectangular and divided into a gridwork of pixels. Each pixel represents a small area of the image. In monochrome images each pixel represents a black or white dot. In color images each pixel is a colored dot. Therefore, the most important characteristics of an image are its:

- Image width
- Image height
- Image resolution (the number of dots or pixels per inch)

An image is also divided into rows, where a row is a grouping of all the pixels in a horizontal strip of the image, the width of the image and one pixel high.

As noted before, a pixel represents a small area of the image. The size of the area depends on the resolution of the image. The pixel area for a 600 dot per inch (dpi) image is a square  $1/600$ th of an inch on a side. Coarser resolutions cover more picture area per pixel. For example, a 75 dpi pixel contains 0.0002 square inches, a 600 dpi pixel contains 0.000003 square inches. Some printing and scanning devices specify a resolution in the horizontal direction and another resolution in the vertical direction, 300 by 600 dpi, for example. However, PCL raster only supports identical resolutions in the horizontal and vertical directions, and therefore, PCL 5 raster graphics pixels are squares.

There is a direct connection between the resolution and the size of an image. The image size is the number of pixels in the image multiplied by the number of bits used to represent a pixel. The smaller the dpi, the smaller the size of the image; the larger the dpi, the larger the size of the image. There is a tradeoff in image size versus print quality. Larger pixels give poorer print quality since diagonal lines can start looking ragged or pixelated, but the image size is small. Smaller pixels give good print quality but larger image size. As a rule, 150 dpi provides a happy medium of good print quality at a relatively small image size.

A printer has a printing resolution, 600 dpi for example. An image with an image resolution smaller than the printer's resolution will undergo a process of scaling as the image is converted to the printer's resolution, also known as device resolution. For example, a 150 dpi image is converted to a 600 dpi image at device resolution by replicating each pixel four times in the horizontal direction and four times in the vertical direction. Therefore, a single pixel of a 150 dpi image is replicated by creating 15 new pixels. A 75 dpi pixel is replicated by creating 63 new pixels.

The large scale replication of pixels for low resolution images is one of the main reasons for poor print quality: first the image detail may be missing from the image because of the large pixel size, and second a blocky, pixelated effect occurs when the pixel is reproduced over and over again. Various techniques have been devised to compensate for this pixelated effect. For example, rather than copy or replicate the pixel, a bi-linear or bi-cubic interpolation algorithm could be used to smooth out the transitions by considering the surrounding pixels when making new pixels. However, this tends to blur edges or sharp transitions that are in the original image. As a rule, better print quality results from higher resolution images, such as 150 or 300 dpi, that use smaller pixels and cause less pixel replication.

Pixels represent specific spots within the picture. In a black and white image, where a pixel is either black or transparent (white), a single binary digit or bit can represent a pixel. This has been the standard raster format for many years. See Figure 6-2 for an example. However, in color images, or grayscale images, those with varying black dot sizes which yield a range of grays when viewed by the human eye, a single bit is not sufficient. Therefore, the general form of a pixel is:

- The number of components in the pixel, which is generally either one or three, and
- The number of bits per component.

This pixel format is sufficient to represent black and white images, gray scale images, and color images. Furthermore, this format can represent images where the color specification is directly in the image, or where the pixel is an index into the palette. The table below shows some possible combinations of the number of components and the bits per component.

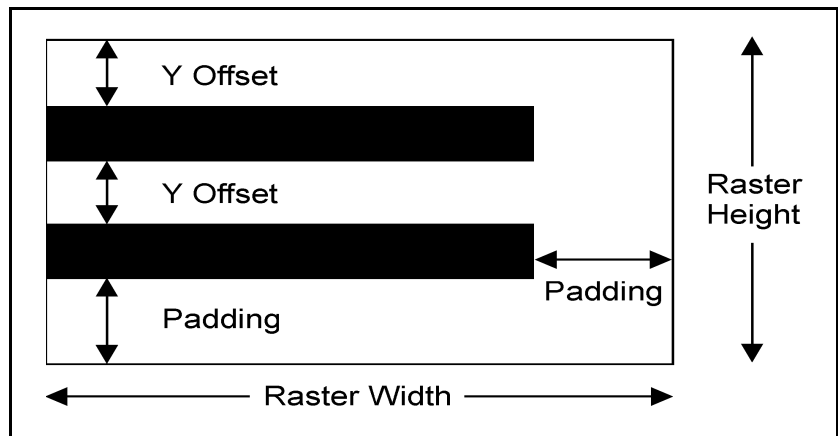
<b>Image Type</b>	<b>Number of Components</b>	<b>Bits per Component</b>
Black and White, direct	1	1
Black and White, indexed	1	1
Color, indexed	1	8
Color, direct	3	8

## PCL 5 Color Raster Graphics

The PCL 5 color raster graphics command set provides support for specifying an image's:

- Dimensions
- Resolution
- Pixel format, also known as the pixel encoding mode
- Compression mode

Large image size is one of the major concerns with raster images. For example, the size of an 8 x 10 inch monochrome image at 150 dpi is 225 Kbytes. The size of a 24-bit color image with the same size and resolution is 5.4 Mbytes. If the color image is 600 dpi, 24-bit color, its size is 86 Mbytes. Therefore, the PCL 5 language supports several compressions modes that exploit redundancy between adjacent pixels or adjacent rows. Furthermore, several shortcuts are available to avoid sending data that is really large blocks of zeros.



**Figure 6-1** Raster Area

An image's width and height define the extent of the image, that is, the number of pixels in a row and the number of rows in the image. The following commands and features eliminate the need to send blocks of zero pixel values:

- 1** The Y Offset command can be used to direct the printer to set all the pixel values in the given number of rows to zero.
- 2** If not enough data is sent for all the pixels in a row, the remaining pixels are set to zero.
- 3** If the image is ended before all rows have been received, the values of the pixels in the remaining rows will be set to zero.

Furthermore, if too much data is sent the data extending outside the raster area is clipped and discarded.

# Raster Graphics Command Sequence

PCL raster commands include: Start Raster Graphics and End Raster Graphics commands, Transfer Raster Data by Plane and Row, Raster Compression, Raster Presentation, Raster Resolution, Raster Height and Raster Width (which define the raster area), and Raster Y Offset commands. For printing well-behaved raster graphics, the normal sequence of execution for these commands is shown below. Hewlett-Packard strongly recommends that developers use this command sequence in their applications.

---

## Note

---

Although the source raster height and width commands are not necessary, they improve memory efficiency.

## Well-Behaved Raster Command Sequence

```
Raster Presentation
Raster Resolution
Raster Height
Raster Width
Start Raster Graphics
Y Offset
Raster Compression
Transfer Raster Data
...

Transfer Raster Data
Y Offset
Transfer Raster Data
...

Y Offset
Raster Compression
Transfer Raster Data
...

Raster Compression
Transfer Raster Data
End Raster Graphics
```

The emphasis in the previous command sequence is that the Raster Presentation Mode, Raster Resolution, Raster Height, and Raster Width are all set outside the *start..data..end* sequence of commands. Also, the entire image is sent during the *start..data..end* sequence, choosing the most effective compression method for each raster row of data.

Raster Presentation, Raster Resolution, Raster Height, Raster Width, and Raster Compression are all true modes. Once specified, the printer remains in that mode unless explicitly changed by issuing the command again, or reset to default values by a soft reset, self test, font printout, or power cycle.

---

**Note**

Only raster data appearing within the intersection of the logical page, the printable area, the raster width, and height is printed. If raster width and/or raster height have not been set, the intersection of the logical page and the printable area determines where raster graphics appear; raster data is clipped to the printable area.

---

# Raster Graphics Resolution Command

Raster graphics can be printed at various resolutions. This command designates the resolution of subsequent raster data transfers in dots per inch.

$E_C$  \* **t** # **R**

# = 75	- 75 dots-per-inch
100	- 100 dots-per-inch
200	- 200 dots-per-inch
150	- 150 dots-per-inch
300	- 300 dots-per-inch
600	- 600 dots-per-inch

**Default** = 75

**Range** = 75, 100, 150, 200, 300, 600

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

---

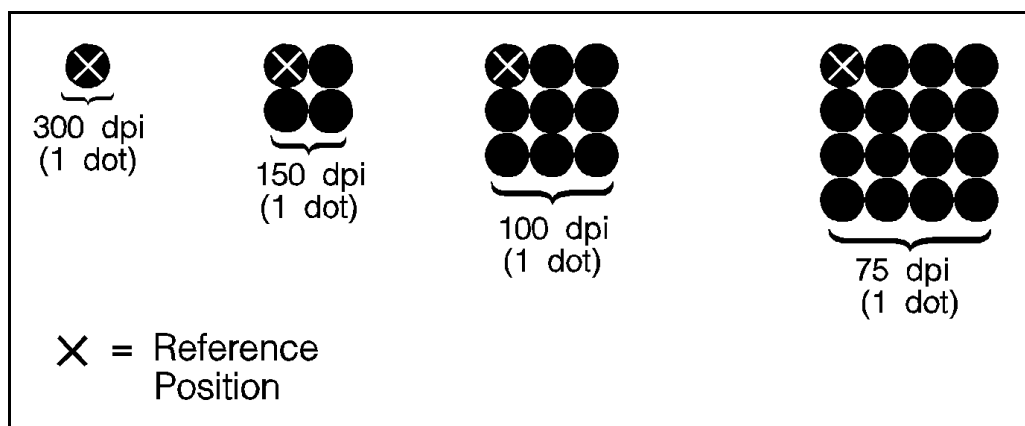
## Note

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 33,750. The same image at 300 dots-per-inch requires 540,000 bits. Note that lower resolution graphics may not give acceptable print quality.

---

When configured for 300 dpi resolution, the printer automatically expands raster graphics transferred at resolutions less than 300 dots-per-inch to 300 dots-per-inch during printing. Figure 6-2 illustrates how a single bit is translated into the corresponding printed dots in various graphics resolutions when the printer is configured for 300 dpi.





**Figure 6-2 Raster Graphics Expansion - at 300 dpi**

**Note**

Rectangular area fills and character data are not affected by changes in resolution. Rectangular Area fills and character data always print at the maximum resolution, regardless of the resolution setting.

# Raster Graphics Presentation Mode Command

The Raster Graphics Presentation command specifies the orientation of the raster image on the logical page.

$E_C$  \* **r** # **F**

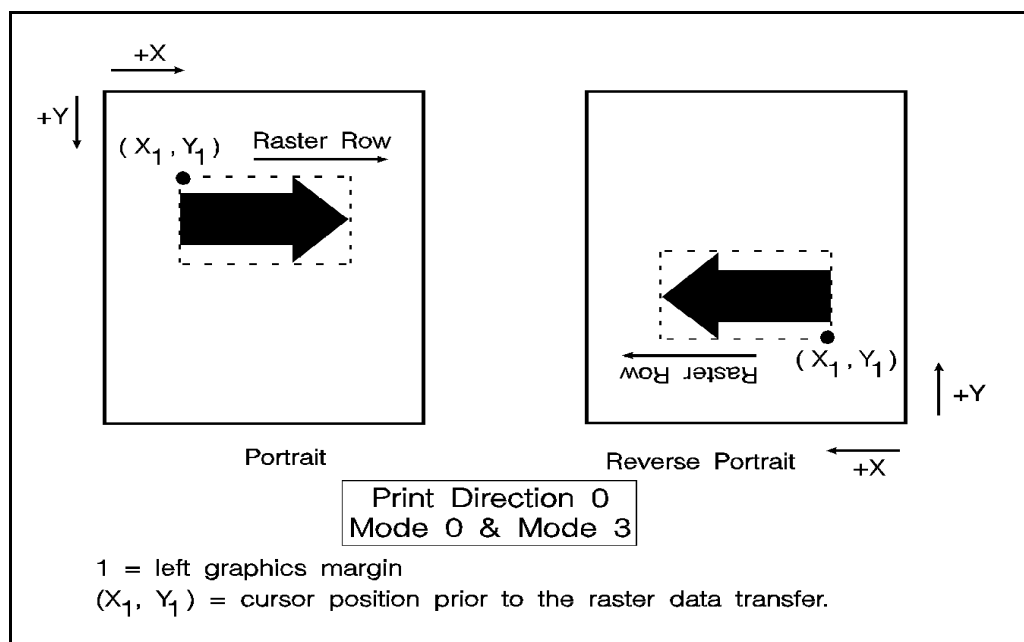
- # =    0 - Raster image prints in orientation of logical page  
      3 - Raster image prints along the width of the physical page

**Default** = 3

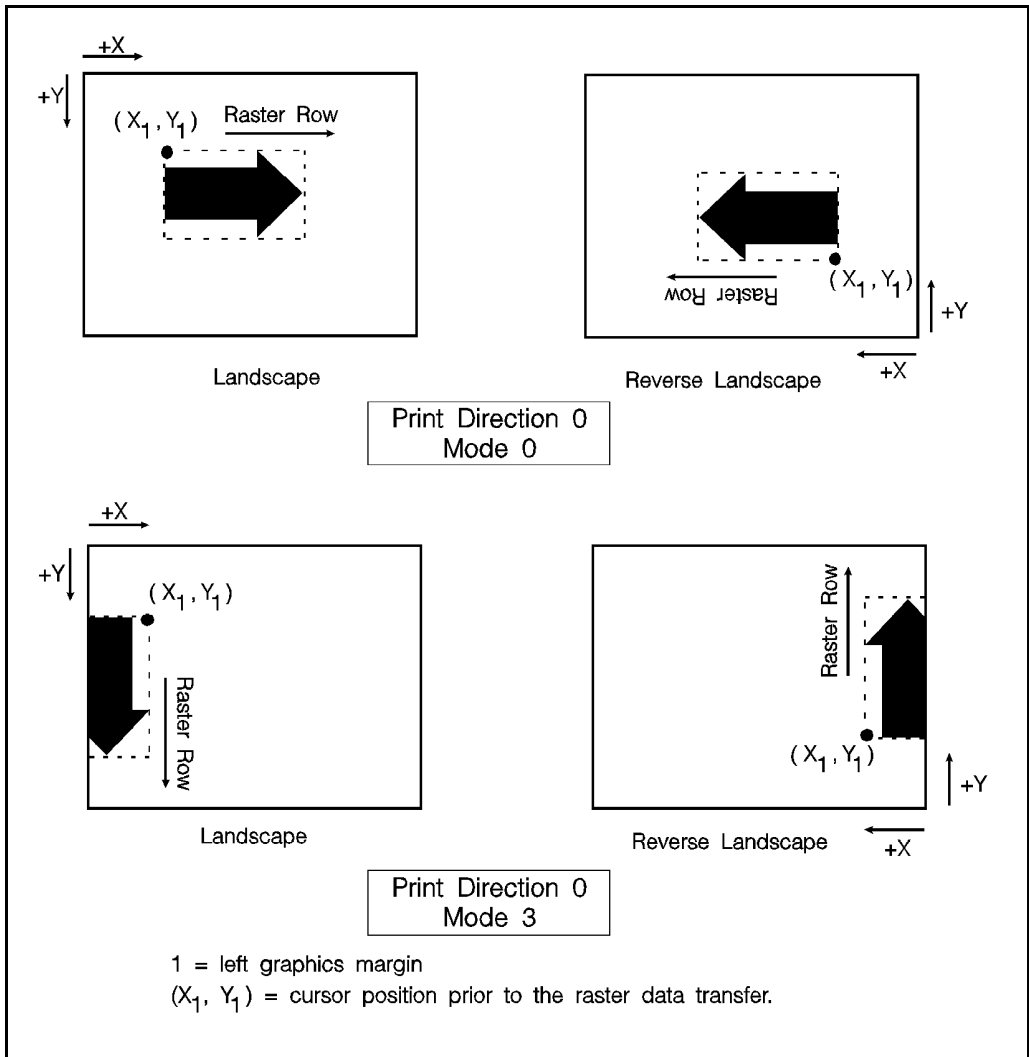
**Range** = 0, 3

- A value of **0** indicates that a raster row will be printed in the positive X-direction of the PCL coordinate system. (The print direction translates the PCL coordinate system.)
- A value of **3** indicates that the raster graphics will be printed along the width of the physical page, regardless of logical page orientation. In portrait orientation, a raster row is printed in the positive X-direction of the PCL coordinate system and a subsequent raster row is printed beginning at the next dot row position in the positive Y-direction. In landscape orientation, a raster row is printed in the positive Y-direction of the PCL coordinate system and a subsequent raster row is printed beginning at the next dot row position in the negative X-direction. Figures 6-3 and 6-4 illustrate presentation modes 0 and 3.

Raster Presentation Mode	Orientation	Default Graphics Margin
0	portrait	logical page left bound
0	reverse portrait	logical page left bound
0	landscape	logical page left bound
0	reverse landscape	logical page left bound
3	portrait	logical page left bound
3	reverse portrait	logical page left bound
3	landscape	50 dots in from the logical page <i>top</i> bound
3	reverse landscape	50 dots in from the logical page <i>top</i> bound



**Figure 6-3 Raster Graphics Presentation Mode for Portrait Orientation**



**Figure 6-4 Raster Graphics Presentation Mode for Landscape Orientation**

# Source Raster Height Command

The Raster Height 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, hence, height is subject to the current raster presentation mode and print direction (see Figure 6-5).

$E_C * r \# T$

# = Height in raster rows

**Default** = N/A

**Range** = 0 to (logical page length – current Y- position of the 0, cursor)\*

\* Greater values default to (logical page length - current Y cursor position).

---

## Note

Specifying the raster width and raster height improves memory usage. Therefore it is highly recommended.

This command fills the raster area to the full raster height with zeroed rows. Unspecified rows map to either white or transparent depending on the source transparency mode (this is true only if index 0 is white).

When a Transfer Raster Data command is received that causes any raster row to extend beyond the row boundary set by the Raster Height command, the row outside the boundary is clipped. This includes the case where the cursor is moved beyond the height boundary with a Raster Y Offset command and the printing of raster data is attempted.

If you have specified either a raster height or a raster width of 0 and a Start Raster Graphics (or Transfer Raster Data) command is received, then the entire raster graphic is clipped. If both a raster height and a raster width are specified (non-zero) and a Start Raster Graphics (or Transfer Raster Data) command is received, then the raster area is guaranteed to be filled.

---

## Note

For color printers, a zero fill is not necessarily white.

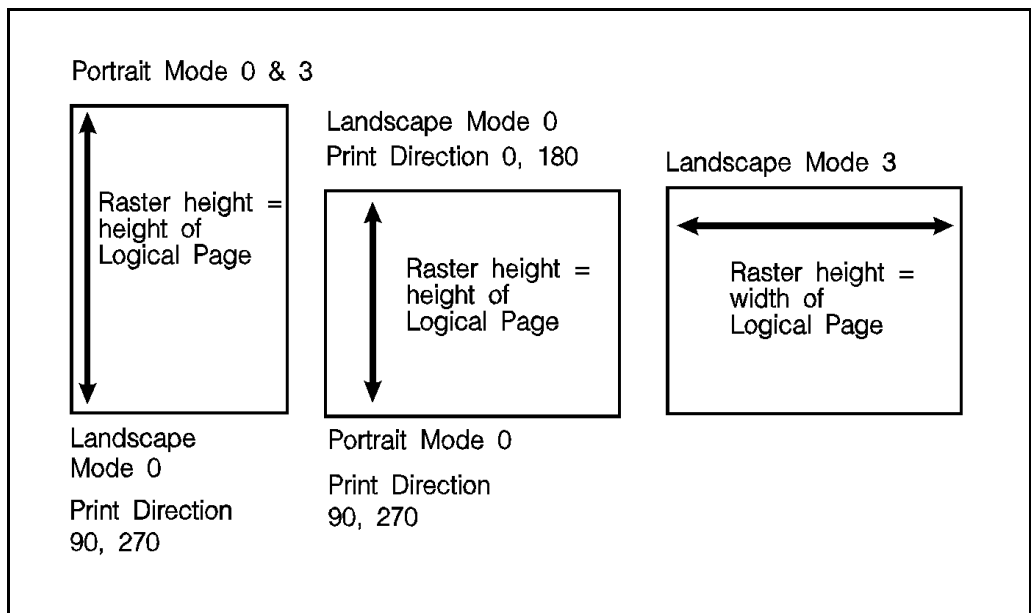
If the raster height is not set, no padding or clipping of rows takes place.

This command is ignored after the Start Raster Graphics or Transfer Raster Data commands until the next End Raster Graphics command.

#### Note

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.

Upon receiving an End Raster Graphics ( $E_c^*rC$ ) command, the cursor position is set to the left graphics margin of the next raster row after the raster height boundary.



**Figure 6-5 Maximum Raster Height**

# Source Raster Width Command

The Raster Width command specifies the width in pixels of the raster area. Width is in the direction that the raster rows are laid down, hence width is subject to the current raster presentation mode and print direction (see Figure 6-6).

$E_C * r \# S$

# = Width in pixels of the specified resolution

**Default** = depends on raster presentation mode: when presentation mode is 0 then width = width of logical page minus left graphics margin; when presentation mode is 3 then width = dimension of logical page along paper length minus left graphics margin.

**Range** = 0 to (logical page width minus left graphics margin)\*

\* Greater values default to the (logical page width – left graphics margin).

---

## Note

Specifying the raster width and raster height improves memory usage. Therefore it is highly recommended.

This command allows you to implicitly tell the printer to pad raster rows that are not specified for the full raster width with zeros. Unspecified data maps to either white or transparent depending on the source transparency mode (this is true only if index 0 is white).

---

## Note

For color printers, a zero fill is not necessarily white.

When a Transfer Raster Data command is received that specifies a row of data that is longer than the raster width, the data that extends beyond the raster width is clipped.

This command is ignored after the Start Raster Graphics or Transfer Raster Data commands, until the next End Raster Graphics command.

---

## Note

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.

---

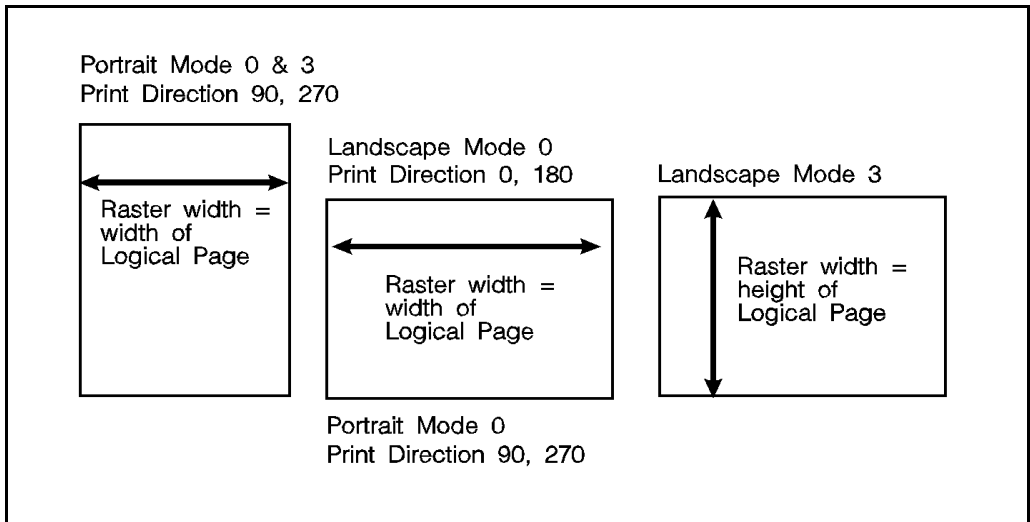


Figure 6-6 Maximum Raster Width



# Start Raster Graphics Command

The Start Raster Graphics command identifies the beginning of the raster data and also specifies the left graphics margin.

$E_C * r \# A$

- $\# =$
- 0 - Start graphics at default left graphics margin (X-position 0).
  - 1 - Start graphics at current cursor position (current X-position).
  - 2 - Raster scaling on—start at left boundary.
  - 3 - Raster scaling on—start at CAP.

**Default** = 0

**Range** = 0 - 3 (out-of-range values default to 0)

A value of 0 specifies that the left graphics margin is at the default left margin of the page (X-position 0). A value of 1 specifies that the left graphics margin is at the current X-position. Values of 2 or 3 are equivalent to 0 and 1, but also enable resolution-independent scaling. In presentation mode 3, the location of the left graphics margin varies depending on the orientation.

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 margin are fixed until an end raster graphics command is received.

Once in Raster Graphics Mode, PCL commands and text imply an End Raster Graphics ( $E_C * rC$ ) except for the following commands:

- Transfer Raster Data by Row or by Plane
- Set Raster Compression Method
- Raster Y Offset

In addition, the following commands are ignored (i.e., locked out) while in Raster Graphics Mode and do not imply an End Raster Graphics command:

- Start Raster Graphics
- Set Raster Width
- Set Raster Height
- Set Raster Presentation Mode

- Set Raster Graphics Resolution
- Palette Commands
- CID Command
- Color Treatment
- Render Algorithm

---

**Note**

An implied End Raster Graphics resets the Raster Compression Method 3 seed row, but does not reset the Raster Compression Method nor the left raster graphics margin.

If source and/or transparency modes have been set, frequent start/end graphics commands in an image can result in a memory overflow condition.

---

# Raster Y Offset Command

The Raster Y Offset command moves the cursor position vertically the specified number of raster lines from the current raster position in the raster area.

$E_C * b \# Y$

# = Number of raster lines of vertical movement

**Default** = N/A

**Range** = 0 - 32767

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

This command zero-fills the offset area. For color printers, zero-fills are filled with the color of index 0, which is not necessarily white.

For Delta Row compression (method 3), this command zeros the seed row. For Adaptive compression (method 5), this command applies to the entire raster data block.

---

## Note

Movement by this command is based upon the Raster Resolution setting ( $E_C * t \# R$ ) and also the printer's resolution setting (75, 150, or 300 dpi).

---

## Set Compression Method Command

The Set Compression Method command allows you to encode the data in a given raster transfer, either by plane or by row. These methods achieve data compression by exploiting byte-to-byte redundancies, either within a row or between rows, without regard to the pixel encoding mode of the data. While these methods reduce the amount of data transferred from the host to the printer, they do not reduce the memory requirements within the printer.

While PCL 5 compression methods are based on redundancies at the byte level, the pixel encoding mode must be considered when choosing a compression method. The direct by pixel mode transfers the three bytes which make up a pixel, one after the other. Therefore, compression formats that depend upon reducing redundancy within a row will not do well, since the pixel to pixel redundancy is disguised by the format. This is because, while two adjacent pixels have a fairly high probability of being identical, the probability that the components of a pixel are identical is very small. For example, consider a raster image whose background is all one color. While there are a large number of pixels that are identical, the only time the bytes within the pixel will be identical are if the color is a gray somewhere between white and black: such as (0, 0, 0), (128, 128, 128), (255, 255, 255). If the pixel represents a non-gray color, the bytes within the pixel will not be identical. Therefore, the only successful compression mode that exploits redundancy between the rows is delta row compression. Since the probability that a pixel is the same in the horizontal direction is usually equivalent to the probability that it is the same in the vertical direction, the bytes within a pixel will be redundant and therefore compressible from row to row.

The remaining pixel encoding modes: index by plane, index by pixel, and direct by plane, can be redundant within a row and are, therefore, compressed well by compression methods such as run-length encoding, and TIFF rev 4.0.

The Set Compression Mode command has the following form:

$E_C * b \# M$

- # = 0 - Unencoded
- 1 - Run-length encoding
- 2 - Tagged Image File Format (TIFF) rev. 4.0
- 3 - Delta row compression
- 5 - Adaptive compression

**Default** = 0

**Range** = 0 - 3, 5 (other values are ignored)

## Unencoded (Method 0)

This is a simple transfer of binary data without any compression.

## Run-length Encoding (Method 1)

Run-length encoding interprets raster data in pairs of bytes. The first byte of each pair is the repetition count for the data in the second byte. The second byte is the raster data to be printed. A repetition count of 0 signifies the pattern in the data byte is not repeated (it occurs only once). A repetition count of 1 signifies the pattern occurs twice. The repetition count can range from 0 to 255 for a repetition of 1 to 256 times.

**[(Repetition count 0-255)(pattern byte)] . [ . ] [ ]**

## Tagged Image File Format Encoding (Method 2)

Tagged Image File Format encoding interprets raster data as TIFF "Packbits." This format combines features of methods 0 and 1. A **control byte** precedes the raster data (pattern bytes). The control byte identifies whether the pattern byte(s) represent a byte that is to be repeated some number of times (up to 127), or represent some number of bytes (up to 127) which are to be printed as is (literally).

The sign of the number in the control byte identifies whether the byte or bytes that follow represent a literal pattern or byte to be repeated. A positive number (1 to 127) indicates that the bytes are literal. A negative number (-1 to -127), represented by the twos complement, indicates a repeated byte. The value of the number, if positive (literal), identifies the number of pattern bytes which follow the control byte; if negative (repeated), it identifies the number of times to repeat the following byte. A pattern byte may be repeated up to 127 times; or up to 127 literal bytes may follow the control byte.

As mentioned, for a byte to be repeated, the control byte must be a negative value as represented by the twos complement. For example, to repeat a pattern three times would require the twos complement of the number 3. The twos complement is computed as follows. The binary of 3 is 00000011. Complement each bit to get 11111100, then add one to this value to produce 11111101, the twos complement. The decimal value of this number, 253, used in the control byte, produces a repetition of 3 bytes for a total of 4 occurrences of the pattern.

The range of numbers for the control byte is shown below.

#### **Literal Pattern Values**

# of Bytes	Binary value	Decimal value
1	0000 0000	1
to	to	to
127	0111 1111	127

#### **No Operation Value**

NOP value	Binary value	Decimal value
128 (-128)	1000 0000	128

#### **Repeated Pattern Values**

# of Repetitions	Binary value*	Decimal value
1 (-1)	1111 1111	255
to	to	to
127 (-127)	1000 0001	129

\* These negative values are represented by taking the twos complement of the value of the number.

**Note** Another method to calculate the number needed in the control byte for some number of repetitions is to subtract the number of desired repetitions from 256. For example, the control value for 3 repetitions (4 occurrences) of a byte is 256 minus 3 = 253.

A zero or positive value in the control byte means that the subsequent byte or bytes are non-replicated bytes of data. The value of the control byte *plus one* indicates the number of data bytes that follow. For example, a control byte of 0 means the following 1 byte is literal raster data. A control byte of 6 indicates that the following 7 bytes are literal raster data bytes.

TIFF encoding also allows you to include a non-operative (NOP) control byte, represented by the value -128. This byte is ignored, and the subsequent byte is treated as the new control byte.

**Note** It is more efficient to code two consecutive identical bytes as a repeated byte. If these bytes are preceded and followed by literal bytes, however, it is more efficient to code the entire group as literal bytes.

**Examples: Run-length and TIFF Compression**

The following examples show how a raster row can be coded using run-length and TIFF compression methods. Note that the compression examples use characters to represent the binary data stream.

Byte Number	#1	#2	#3	#4	#5	#6	#7
Bits	01010101	01010101	01010101	01010101	01000001	01010100	01010100
ASCII	U	U	U	U	A	T	T

**Unencoded**

E<sub>C</sub>\*r1A  
E<sub>C</sub>\*b0m7WUUUUATT  
E<sub>C</sub>\*rC

## Run-length Encoding

$E_C^*r1A$   
 $E_C^*b1m6W(3)U(0)A(1)T$   
 $E_C^*rC$

## TIFF Encoding

$E_C^*r1A$   
 $E_C^*b2m6W(-3)U(0)A(-1)T$  or  $E_C^*b2m6W(-3)U(2)ATT$   
 $E_C^*rC$

In the TIFF encoding example above, parenthetical expressions are used to identify control bytes. For example, the byte (-3) is shown to represent the control byte for a repetition (minus value) of 3. The actual value for this position is the decimal value 253. Additional “encoded” control bytes in this sequence include: (0) for decimal 0, (-1) for decimal 255, and (2) for decimal 2. The raster data (pattern) bytes are represented by the ASCII character.

## Delta Row Compression (Method 3)

Delta row compression identifies a section of bytes in a row that is different from the preceding row, and then transmits only that data that is different (the delta data). If a row is completely different from its preceding row, then the entire row must be sent as the delta, which is not very efficient; if only one bit is different, then only one byte is identified and sent. To reassemble the raster data rows, the printer takes the current row (the seed row) and makes the changes indicated by the delta data, to create the new row. The new row (which becomes the new seed row) is used by the next delta compression data to create another row.

A delta compression row consists of two parts, a command byte and the replacement bytes, as shown below:

**[(Command byte)(1 to 8 Replacement bytes)]**



The command byte identifies two things: 1) the number of replacement (delta) bytes that follow; and 2) where to position the replacement byte string (the left offset). The replacement bytes are some number (up to eight bytes) of consecutive bytes that are used to create the new row from the seed row.

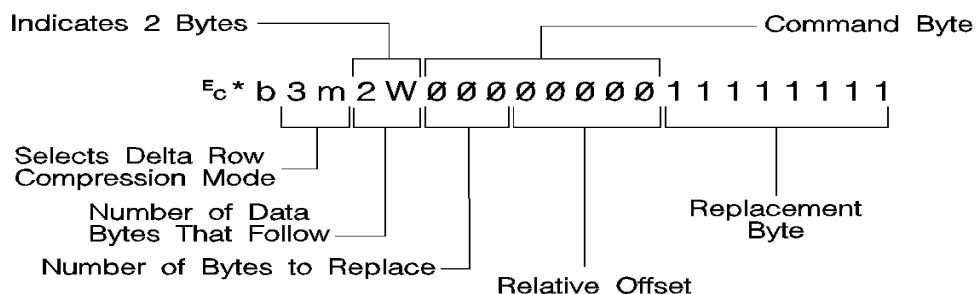
7	5	4	0
Number of bytes to replace (1-8)		Relative offset from last untreated byte	

If more than eight replacement (delta) bytes are needed, additional command byte/replacement bytes may be added, as shown below:

$E_C * 3m \# W$  [(Command Byte)(1 to 8 Replacement Bytes)][(Command Byte)(1 to 8 Replacement Bytes)]. . .

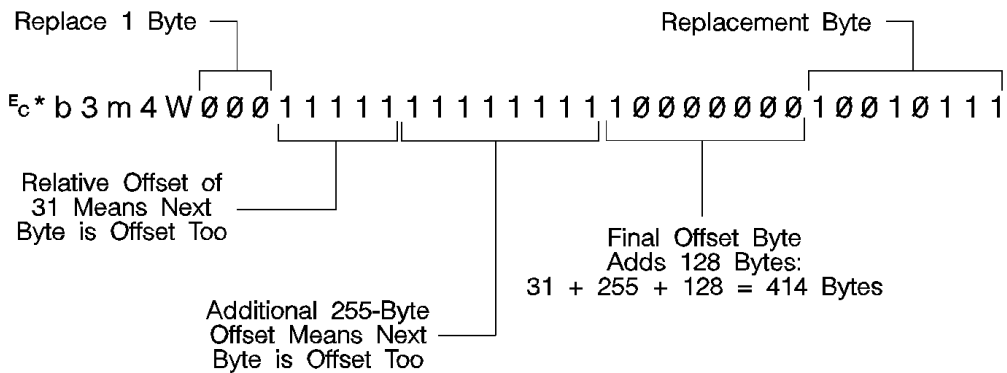
In the command byte, the upper three bits identify the number of replacement (delta) bytes (which can be 1 to 8 bytes). The lower five bits identify the location the replacement bytes are to be positioned. This position is identified as the **offset**, or the number of bytes from the treated byte. For example, if there are 5 replacement bytes and the offset is 7, then the replacement bytes replace bytes 7, 8, 9, 10, and 11 (the five bytes beginning at byte 7 from the seed row).

If there is more than one replacement in a row, the second offset is counted from the next untreated byte in the row: the first byte following the last replacement byte.



As mentioned, the offset contained in the lower five bits of the command byte allows for offset values from 0 to 31. Compression mode allows offsets larger than 31 bytes as follows:

- An offset value of 0-30 indicates that the replacement bytes are offset from the 1st byte to the 31st byte.
- A value of 31 indicates that the next byte following the command byte is an additional offset byte which adds to the first (32) offset value. This allows offset values larger than 31. Also, if this second offset byte is set to 255 (all ones), additional offset bytes follow until the required offset value is obtained. When the formatter detects an offset byte less than 255, it is assumed to be the last offset value and the offset bytes are then totaled (added). The following example shows an offset larger than 31:



The total offset is 414, which is the sum of the three offset values:  
 $31 + 255 + 128$ .

## Seed Row

The seed row is basically the current raster data row, the row being printed. It is maintained by the printer for use by delta row compression. The delta compression replacement bytes are applied to the seed row to create the new row. This new data row is printed and becomes the new seed row.

For color raster images, the printer operates on each plane independently, and a separate seed plane is maintained for each graphic plane. A Y offset, however, affects all planes and seed rows simultaneously.

The seed row is updated by every raster graphic transfer, regardless of the compression method. This allows the delta compression method to be mixed with other methods to achieve better compression performance.

### Repeating a Row

$E_C^* b 0 W$

When using the delta compression method, it is possible to repeat or copy the previous raster row using the Raster Data Transfer command. This is accomplished by setting the Raster Data Transfer command value field to zero.

### Printing A Zeroed Row (Setting the Seed Row to Zero)

$E_C^* b 1 Y$

It is possible to print a row of all zeros using the Raster Y-Offset command. Sending a Raster Y Offset command with a value field of 1 sets the seed row to zero and prints the zeroed row. Note that the next delta row is applied to a zeroed seed row.

Other cursor position moves set the seed row to zeros. (Remember, non-graphic cursor moves have the same effect as an end graphics command.)

**Note**

If the byte count of the Transfer Raster Data command value field is less than the number of bytes that can be replaced, the byte count has precedence. Also, if the last byte is a control byte, it is ignored. Therefore,  $E_C^* b 1 W$  does not affect the seed row, but causes the previous row to be replicated.

### Example: Delta Row Compression

The following example demonstrates how to compress the following data using the delta row compression. (The bytes highlighted in bold type indicate those bytes needing replacement – those bytes that are different from the previous row, the seed row.)

Byte No.	0	1	2	3	4
Row 1	00000000	<b>11111111</b>	00000000	00000000	00000000
Row 2	00000000	11111111	<b>11110000</b>	00000000	00000000

Row 3      **00001111 11111111 11110000 10101010 10101010**

$E_C^*r1A$  – The *start raster graphics* command initializes the seed row to all zeros.

**Row 1** –  $E_C^*b3m2W(00000001)(11111111)$

The **3m** selects the delta row compression method and the **2W** indicates 2 bytes of data to follow. The first three bits of the first data byte, the command byte, signify a single byte replacement (all three bits are 0). The next five bits indicate an offset of 1 byte from the current position. The replacement byte follows and contains **11111111**.

**Row 2** –  $E_C^*b2W(00000010)(11110000)$

The first three bits of the command byte indicate that one byte will be replaced, and the next five bits indicate a relative offset of 2, so the replacement will occur 2 bytes from the current position. The replacement byte follows and contains 11110000.

**Row 3** –  $E_C^*b5W(00000000)(00001111)(00100010)(10101010)(10101010)$

As in the other rows, the first three bits of the command byte are zero, indicating a single byte replacement. The five offset bytes indicate a relative offset of zero bytes. The replacement byte follows and is 00001111. The third byte is another command byte and the first three bits signify the replacement of two bytes (the top three bits are 001). The offset bits indicate an offset of two bytes from the current position. The fourth and fifth bytes are the two replacement bytes.

## Adaptive Compression (Method 5)

Adaptive compression enables the combined use of any of the four previous compression methods (0 through 3), and it includes the ability to print empty (all zeros) rows or to duplicate rows.

Adaptive compression interprets a raster image as a block of raster data rather than as individual rows. The result of this interpretation is that the Transfer Raster Data ( $E_C^*b\#W$ ) command is sent only once at the beginning of a raster data transfer, and the value field (**#**) identifies the number of bytes in the block of rows. For the other compression methods, the Transfer Raster Data command is sent at the beginning of each row and the value field (**#**) identifies the number of bytes for that row only.

The size of a block is limited to 32,767 bytes. (32,767 bytes is the number of compressed bytes and not the size of the uncompressed data). To transfer greater than 32,767 bytes, send multiple blocks.

Adaptive compression uses three control bytes at the beginning of each row within the block. The first of these bytes, the command byte, identifies the type of compression for the row. The two following bytes identify the number of bytes or rows involved. The format for adaptive compression raster rows is shown below:

<command byte><# of bytes/rows - upper byte><# of bytes/rows - lower byte> ...  
... <first raster row byte> ... <last raster row byte>

The command byte designates the compression method, empty row, or row duplication. Command byte values are shown below.

Value	Compression Operation
0 –	Unencoded
1 –	Run-Length Encoding
2 –	Tagged Image File Format (TIFF) rev 4.0
3 –	Delta row
4 –	Empty row
5 –	Duplicate row

For command byte values 0 - 3, the two **<# of bytes/rows>** bytes specify the number of bytes (row length) for the row. For command byte values 4 and 5, these bytes identify the number of empty or duplicate rows to print. The maximum value for these two bytes is 65,535; however, the image is clipped to the logical page. Thus, the value of these bytes should not exceed the maximum number of bytes/rows that can be printed on the current logical page size.

If an out-of-range command byte is encountered, the remainder of the block is skipped, the cursor is not updated, and the seed row is cleared.

Compression methods 0 - 3 are the compression methods used by the Set Compression Method command. Value fields 4 and 5 are features for the adaptive compression method and are explained below.

## **Empty Row**

A command byte of 4, empty row, causes a row of zero's to be printed. The number of rows printed depends on the value contained in the two **<# of bytes/rows>** bytes following the command byte. The empty row operation resets the seed row to zero and updates the cursor position.

## **Duplicate Row**

A command byte of 5, duplicate row, causes the previous row to be printed again. The row can be duplicated the number of times indicated by the value contained in the **<# of bytes/row>** byte. Duplicate Row updates the cursor position but does not change the seed row.

## Adaptive Compression Operation Hints

---

### Note

---

Some HP LaserJet printers perform internal compression techniques to support full-page graphics. Refer to Chapter 1 of the *PCL 5 Comparison Guide* for specifics.

- The compression methods cannot be mixed within one raster row. A raster row must be compressed using only one method.
- The cursor position is updated with each row of the raster block. The cursor position is also incremented when a block count of less than 3 is sent.
- A Raster Y-Offset command moves the entire block of raster data and initializes the seed row to zeros. The seed row is set to zero even if the y-offset is zero.
- Block size takes precedence over row length. If the row length of any line exceeds the block size, the row length is truncated to the block size.
- For duplicate and empty rows, a row length value of zero does not update the cursor, however the seed row is initialized to zero.
- If an unsupported command byte for a raster row is encountered, the remaining bytes for the block are skipped, the seed row is cleared, and the cursor is not incremented.
- For method 1, run length encoded, if the row length is odd, the cursor is incremented, the row data is skipped (thrown away), and the seed row is left unchanged.
- For method 1, a row length value of zero increments the cursor and zero fills the seed row.
- For method 2, TIFF, if row length terminates the data before the control byte value is satisfied (literal byte count greater than row length), the data following the control byte, if any, is printed as text. The cursor is incremented.
- For Method 2—if row length is equal to one, the one byte is consumed from the I/O and the cursor is incremented. The data is ignored and the seed row is zeroed.
- For Method 3—delta row compression, within an adaptive compression block, the seed row is updated by every raster compression method or type of row. For example, a row compressed with Method 2, TIFF, updates the seed row, while the effect of an empty row initializes the seed row to zeros. Maintaining the seed row allows Method 3 to be mixed with other methods to achieve optimal compression performance.

- For Method 3—since delta row compression requires that the seed row be available whenever raster graphics mode is entered, the seed row is initialized to zeros upon raster graphics mode entry ( $E_C*r\#A$ ). The seed row is also initialized upon receipt and completion of each raster block.
- For Method 3—if the row length terminates the data before the control byte value is satisfied (literal byte count greater than row length), the data following the control byte, if any, is printed as text. The cursor is incremented.
- For Method 3—if the row length is equal to one, the current row is duplicated and the cursor is incremented.

## Transfer Raster Data Commands

There are two Transfer Raster Data commands: *Transfer Raster Data by Plane* and *Transfer Raster Data by Row*.

- **Transfer Raster Data by Plane**—This command ( $E_C*b\#V$ ) is used when the raster data is encoded by plane as specified by the Simple Color command ( $E_C*r\#U$ ) or the Configure Image Data command ( $E_C*v\#W$ ). The *Transfer Raster Data by Plane* command is used to send each plane in the row except the last; the *Transfer Raster Data by Row* command ( $E_C*b\#W$ ) must be used to send the last plane and advance the cursor to the beginning of the next row.
- **Transfer Raster Data by Row**—This command ( $E_C*b\#W$ ) moves the current active cursor position to the next pixel row after its execution. It is used for monochrome printers for the last plane in a multi-plane row, or for color raster transfer when the data is encoded by pixel.

Both commands are described in detail in the following paragraphs. Chapter 2 provides additional descriptions and examples using the Transfer Raster Data commands to print color images.



## Transfer Raster Data by Plane

This command sends a plane of data to the printer and advances to the next plane (not to the next row).

$E_C * b \# V$  [*raster data*]

**Default** = N/A

**Range** = 0 to 32767

The value field (#) identifies the number of bytes in the plane. The number of planes per row is specified by the Simple Color command ( $E_C * r \# U$ ) or the Configure Image Data command ( $E_C * v \# W$ ), depending on which color mode is used. The first plane sent represents the least significant bit in the pixel.

Since  $E_C * b \# V$  does not advance the cursor to the beginning of the next raster row, it cannot be used for the last plane or for single-plane rows. Only  $E_C * b \# W$  can advance the cursor to the next row.

The amount of data sent varies from plane to plane and is independent of raster width. Planes shorter than the raster width are zero-filled. Empty planes can be sent using  $E_C * b0V$ .

---

### Note

For monochrome printers, zero indicates a white pixel. For color printers, the color indicated by zero depends on the current palette.

---

## Transfer Raster Data By Row/Block Command

The Transfer Raster Data command is used to transfer a row of raster data to the printer. This command is used for sending all raster graphics data to monochrome printers. It is also used for color printers when encoding the raster data by pixel rather than by plane. When encoding color raster data by plane, this command is used for single-plane rows, or for the last plane in a multi-plane row, since this command advances the cursor position to the beginning of the next raster row.

$E_C * b \# W$  [*raster data*]

**Default** = N/A

**Range** = 0 to 32767

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

Raster graphics are independent of the text area and perforation skip mode—these boundaries are ignored.

Raster graphic images, raster height, and raster width are limited to the printable area; images that extend beyond the printable area are clipped.

## Byte Counts and the TIFF v4.0 Compression Mode

The byte count of the value field in the Transfer Raster Data command has precedence over the literal, or the command byte, byte count of the TIFF v. 4.0 compression mode. For example, the command,

Ⓔ<sub>C</sub>\*b2m3W [*binary data*]

sets compression method=2 (TIFF v. 4.0) and sends 3 bytes of raster data for the row. Suppose the binary data appears as follows:

00000010 00000001 00000001 00000001

The control (first) byte value of 2 indicates that 3 bytes of literal (unencoded) raster data will follow. The Transfer Raster Data command, however, specified only three bytes total (including the control byte) in the raster row. The control byte and the following two data bytes are read, and the remaining data byte is ignored.

If the last byte indicated by the value field in the Transfer Raster Data command is a control byte, that byte is ignored.

---

### Note

If a Transfer Raster Data command is received without an accompanying Start Raster Graphics command, any preceding start raster values are used (such as left graphics margin, raster height and width, etc.).

---

# End Raster Graphics Command

The End Raster Graphics command signifies the end of a raster graphic data transfer.

$E_C * r C$

Receipt of this command causes 5 operations:

- Resets the raster compression seed row to zeros.
- Moves the cursor to the raster row immediately following the end of the raster area (if a source raster height was specified).
- Allows raster commands which were previously locked out to be processed.
- Sets compression mode to 0 (no compression).
- Defaults the left graphics margin to X-position 0.

---

## Note

This command is a modified version of the  $E_C * r B$  End Raster Graphics command. The newer version ( $E_C * r C$ ) performs two additional operations: it resets the compression mode to 0 and defaults the left graphics margin to 0.

This command ( $E_C * r C$ ) is not supported by the HP LaserJet III or the HP LaserJet IIID printers. Use the  $E_C * r B$  End Raster Graphics command to terminate raster graphic data transfers for these printers.

Refer to the “PCL Feature Support Matrix” in Chapter 1 of the *PCL 5 Comparison Guide* for specific printers which support these commands.

---

# Raster Scaling

Raster scaling provides the ability to enlarge or reduce raster images using the Destination Raster Width and Destination Raster Height commands. The Start Raster command ( $E_c*r\#A$ ) with a value field of 2 or 3 turns on scale mode. Scaling is independent of device resolution.

---

## Note

To use raster scaling, the Configure Image Data command ( $E_c*r\#W$ ) must be sent prior to the Start Raster command ( $E_c*r\#A$ ), which must have a value field of 2 or 3 to enable scaling.

---

There are two types of raster scaling: Resolution and Arbitrary. The scaling type is selected by the argument to the Start Raster command. If either 0 or 1 are used then resolution scaling is performed. If either 2 or 3 is used then arbitrary scaling is performed.

## Resolution Scaling

This scaling is performed using the raster resolution in combination with the device resolution. For example, if the raster resolution is set to 300 dpi and the device resolution is 600 dpi, then the image is scaled by a factor of two. This type of scaling does not depend on the source or destination raster dimensions.

## Arbitrary Scaling

This scaling is performed without regard to the raster resolution which is unaffected by scale calculations. The scaling factor is determined by the source and destination raster dimensions. If the destination dimensions are not specified the graphics margin and printable area are used to calculate destination dimensions while maintaining isotropic scaling so that the entire image fits on the page. If only one destination boundary is specified the other is calculated to maintain isotropic scaling.

The destination width and height commands accept real numbers with up to four decimal places of precision. This level of precision is necessary when converting from decipoints to pixels so that round off errors will not create visible print artifacts such as lines within the image.

## Destination Raster Width

The Destination Raster Width command defines the width in decipoints of the destination raster picture denoted by the next Start Raster command, which must have a value field of 2 or 3 ( $E_C^*r2A$  or  $E_C^*r3A$ ).

$E_C^*t\#H$

$\#$  = Width (in decipoints)

**Default** = Right logical page boundary minus left graphics margin

**Range** = 0 – 32767.0000 (values outside the range are ignored)

Zero or absent values default the destination width to a value that preserves isotropic scaling.

A specified width that would cross the right physical page boundary is clipped at the right physical page boundary, but the scale factor is maintained.

## Destination Raster Height

The Destination Raster Height command defines the height in decipoints of the destination raster picture denoted by the next Start Raster command, which must have a value field of 2 or 3 ( $E_C^*r2A$  or  $E_C^*r3A$ ).

$E_C^*t\#V$

$\#$  = Height (in decipoints)

**Default** = Bottom logical page boundary minus vertical CAP

**Range** = 0 – 32767.0000 (values outside the range are ignored)

Zero or absent values default the destination height to a value that preserves isotropic scaling.

A specified height that is longer than the physical page is clipped at the bottom of the physical page, but the scale factor is maintained.

# Raster Graphics Example

To transfer an unencoded arrow-shaped raster graphic image (see Figure 6-7) in the shape of an arrow, perform the following steps (notice the well-behaved raster sequence is utilized):

**1** Position the cursor:

<code>E<sub>C</sub>*p300x400Y</code>	Move the cursor to PCL Unit position (300, 400) within the PCL coordinate system.
--------------------------------------	---

**2** Specify Raster Presentation Mode 0:

<code>E<sub>C</sub>*r0F</code>	Print raster graphics in the orientation of the logical page.
--------------------------------	---

**3** Specify the raster graphics resolution:

<code>E<sub>C</sub>*t75R</code>	Set raster resolution to 75 dpi.
---------------------------------	----------------------------------

**4** Specify the raster graphics height:

<code>E<sub>C</sub>*r32T</code>	Specify a raster height of 32 pixels (32 rows of raster data).
---------------------------------	--

**5** Specify the raster graphics width:

<code>E<sub>C</sub>*r32S</code>	Specify a 32-pixel raster width.
---------------------------------	----------------------------------

**6** Specify the left raster graphics margin:

<code>E<sub>C</sub>*r1A</code>	Set the left graphics margin to the current X position (300).
--------------------------------	---

**7** Specify the Y offset:

<code>E<sub>C</sub>*b0Y</code>	This specifies a Y offset of 0 (this command is not necessary here but shows the proper command sequence).
--------------------------------	--

**8** Specify the raster compression mode:

$E_C*b0M$                       No compression (unencoded).

**9** 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 on the following page.

**10** Signify the end of the raster image transfer:

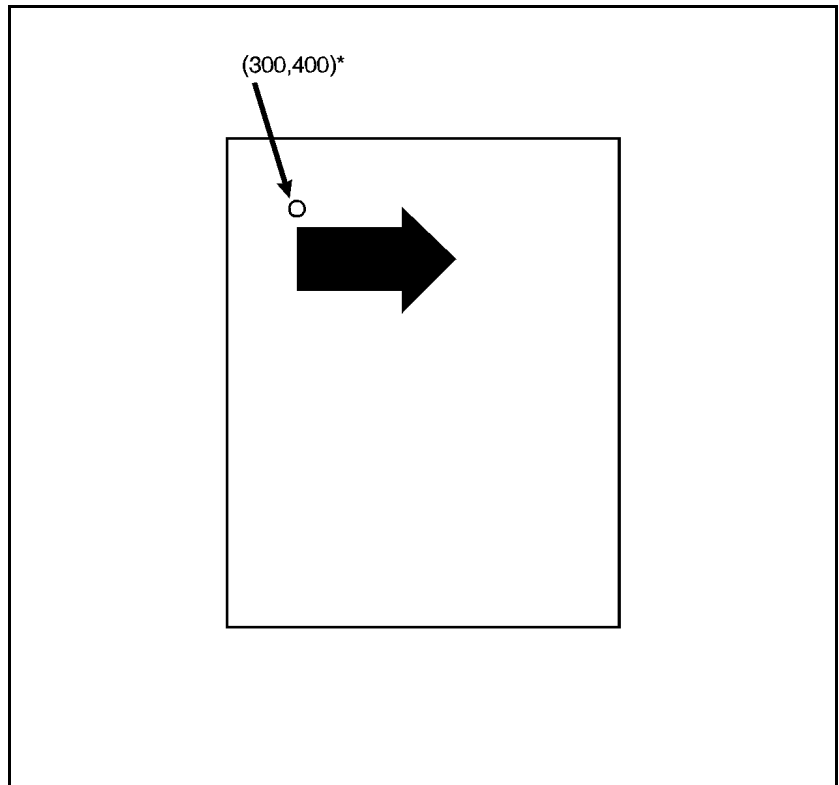
$E_C*rC$                       This example prints the arrow as shown in Figure 6-7.

Example of Raster Graphic Image Data					
Raster Image Data				Command Data	
Dot Row	byte 1	byte 2	byte 3	byte 4	Decimal Equivalent
1	00000000	00000000	10000000	00000000	$E_C*b4W[0, 0, 128, 0]$
2	00000000	00000000	11000000	00000000	$E_C*b4W[0, 0, 192, 0]$
3	00000000	00000000	11100000	00000000	$E_C*b4W[0, 0, 224, 0]$
4	00000000	00000000	11110000	00000000	$E_C*b4W[0, 0, 240, 0]$
5	00000000	00000000	11111000	00000000	$E_C*b4W[0, 0, 248, 0]$
6	00000000	00000000	11111100	00000000	$E_C*b4W[0, 0, 252, 0]$
7	00000000	00000000	11111110	00000000	$E_C*b4W[0, 0, 254, 0]$
8	00000000	00000000	11111111	00000000	$E_C*b4W[0, 0, 255, 0]$
9	00000000	00000000	11111111	10000000	$E_C*b4W[0, 0, 255, 128]$
10	11111111	11111111	11111111	11000000	$E_C*b4W[255, 255, 255, 192]$
11	11111111	11111111	11111111	11100000	$E_C*b4W[255, 255, 255, 224]$
12	11111111	11111111	11111111	11110000	$E_C*b4W[255, 255, 255, 240]$
13	11111111	11111111	11111111	11111000	$E_C*b4W[255, 255, 255, 248]$
14	11111111	11111111	11111111	11111100	$E_C*b4W[255, 255, 255, 252]$
15	11111111	11111111	11111111	11111110	$E_C*b4W[255, 255, 255, 254]$
16	11111111	11111111	11111111	11111111	$E_C*b4W[255, 255, 255, 255]$

Example of Raster Graphic Image Data					
Raster Image Data				Command Data	
Dot Row	byte 1	byte 2	byte 3	byte 4	Decimal Equivalent
17	11111111	11111111	11111111	11111111	E <sub>C</sub> *b4W[255,255,255,255]
18	11111111	11111111	11111111	11111110	E <sub>C</sub> *b4W[255,255,255,254]
19	11111111	11111111	11111111	11111100	E <sub>C</sub> *b4W[255,255,255,252]
20	11111111	11111111	11111111	11111000	E <sub>C</sub> *b4W[255,255,255,248]
21	11111111	11111111	11111111	11110000	E <sub>C</sub> *b4W[255,255,255,240]
22	11111111	11111111	11111111	11100000	E <sub>C</sub> *b4W[255,255,255,224]
23	11111111	11111111	11111111	11000000	E <sub>C</sub> *b4W[255,255,255,192]
24	00000000	00000000	11111111	10000000	E <sub>C</sub> *b4W[ 0, 0,255,128]
25	00000000	00000000	11111111	00000000	E <sub>C</sub> *b4W[ 0, 0,255, 0]
26	00000000	00000000	11111110	00000000	E <sub>C</sub> *b4W[ 0, 0,254, 0]
27	00000000	00000000	11111100	00000000	E <sub>C</sub> *b4W[ 0, 0,252, 0]
28	00000000	00000000	11111000	00000000	E <sub>C</sub> *b4W[ 0, 0,248, 0]
29	00000000	00000000	11110000	00000000	E <sub>C</sub> *b4W[ 0, 0,240, 0]
30	00000000	00000000	11100000	00000000	E <sub>C</sub> *b4W[ 0, 0,224, 0]
31	00000000	00000000	11000000	00000000	E <sub>C</sub> *b4W[ 0, 0,192, 0]
32	00000000	00000000	10000000	00000000	E <sub>C</sub> *b4W[ 0, 0,128, 0]

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





**Figure 6-7** Example of Raster Graphic Image Data

## Color Raster Graphics Example

This example demonstrates the Indexed by Pixel PEM of the CID command. It produces a one inch wide by one half inch high image of one inch high by 0.1 inch wide colored vertical bars. The pixels in the image are eight byte indices, repeatedly ranging from 0 to 7, thus giving the different colored vertical bars.

The PCL commands are decoded as follows:

PCL Command	Description	Command Arguments
<ESC>E	Reset	
<ESC>*v6W\x00\x01\x08\x08\x08\x08	Configure Image Data	Color space: device RGB Pixel Encoding Mode: indexed by pixel Bits/Index: 8 Bits/Primary 1: 8 Bits/Primary 2: 8 Bits/Primary 3: 8
<ESC>*t600R	Change graphics resolution	600
<ESC>*r600S	Set Source Raster Width	600 pixel units
<ESC>*r300T	Set Source Raster Height	300 pixel units
<ESC>*r0A	Activate graphics mode	
<ESC>*b0M	Compression method	unencoded
<ESC>*b#W	Transfer raster data row (1)	600 bytes
<ESC>*b3M	Compression method	delta row
<ESC>*b#W	Transfer raster data (299)	600 bytes
<ESC>*rC	End raster graphics	
<ESC>E	Reset	

In the textual equivalent of the PCL 5 commands shown below, the following character representations are used:

- \e stands for the ASCII escape character
- \r is a carriage return
- \x00 to \x07 are the single byte, binary values from 00 to 07

Looking at the PCL commands, you'll notice that the first row is sent unencoded and then the remaining rows replicated by switching to delta row encoding and sending 299 zero rows. This method depends on two features of the PCL 5 command set:

- 1 The 1st row became the seed row at the end of the first row transfer command.
- 2 When in Delta Row compression mode, zero length transfers cause the seed row to be replicated.

The first row is actually a good candidate for run length or TIFF pack bits encoding but is not encoded for clarity.

[illegible]

