17 An Introduction to HP-GL/2 Vector Graphics

The PCL 5 printer provides the ability to print vector graphics using the HP-GL/2 graphics language. HP-GL/2 graphics may be created within application software, or imported from existing applications. For various types of images (many technical drawings and business graphics, for example), it is advantageous to use vector graphics instead of raster graphics. The advantages include faster I/O transfer of large images and smaller disk storage requirements.

Note

As a guideline, use raster graphics for small, complex images, or those images that cannot be accomplished with HP-GL/2 (such as scanned photographs). Use HP-GL/2 for images that would involve a large amount of I/O data transfer if printed using raster graphics, or for drawings that are already in HP-GL/2 format. If the image is easier to describe using vectors instead of raster lines, the image usually prints faster using HP-GL/2.

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

Learning HP-GL/2

Read through this chapter and Chapter 18 for a general overview of the HP-GL/2 language and its relationship to the PCL printer language. Then, flip through the other HP-GL/2 chapters until you see an example that interests you or fits your objective. Read through the examples and try printing them using your choice of programming languages. If you need help converting the generic commands shown in the examples to a programming language, see “Using HP-GL/2 with Programming Languages” later in this chapter.

As you see unfamiliar commands, find the page number of the command description in the index and read about the command. Think of an application that you would like to program and then look for an example that uses some of the elements you desire. After trying some examples and seeing how the commands interact, you should be well on your way to learning the HP-GL/2 language.

This chapter describes the interaction between the PCL printer language and HP-GL/2 modes and introduces the following topics:

- HP-GL/2 Commands and Syntax
- Using HP-GL/2 with Programming Languages
- The HP-GL/2 Coordinate System
- HP-GL/2 and PCL Orientation Interactions
- The Vector Graphics Limits
- Units of Measure
- Pen Status and Location
- Defining the Image Area (PCL Picture Frame)
- Scaling
- Automatically Adjusting Image Size
- Absolute and Relative Pen Movement

Chapter 18 covers more HP-GL/2 fundamentals, and Chapters 19 through 23 discuss HP-GL/2 commands and their syntax.

HP-GL/2 Commands and Syntax

There are two classes of commands used to print vector graphics: PCL printer language commands and HP-GL/2 commands. As the name implies, the *PCL printer language commands* are used when in the PCL printer language mode. They define the area on the page where HP-GL/2 graphics are printed and provide a means to enter HP-GL/2 mode. The *HP-GL/2 commands* are used within HP-GL/2 mode. They define the image that is printed, and allow you to return to the PCL printer language mode. The HP-GL/2 language has its own syntax, and each command is listed in this section of the manual.

The vector graphics commands have been grouped into functional categories. The categories are designated as shown in Table 17-1 through . Each of the command categories is discussed in its own chapter, beginning with Chapter 19, *The Configuration and Status Group*.

Table 17-1 The HP-GL/2 Commands by Group (1 of 5)

CONFIGURATION GROUP	
CO	Comment
DF	Default Values
IN	Initialize
IP	Input P1 and P2
IR	Input Relative P1 and P2
IW	Input Window
PG[1]	Advance Page
RO	Rotate Coordinate System
RP[1]	Replot
SC	Scale

1. Ignored by HP LaserJet printers.

Table 17-2 The HP-GL/2 Commands by Group (2 of 5)

VECTOR GROUP	
AA	Arc Absolute
AR	Arc Relative
AT	Absolute Arc Three Point
BR	Bezier Relative
BZ	Bezier Absolute
CI	Circle
PA	Plot Absolute
PD	Pen Down
PE	Polyline Encoded
PR	Plot Relative
PU	Pen Up
RT	Relative Arc Three Point

Table 17-3 The HP-GL/2 Commands by Group (3 of 5)

POLYGON GROUP	
EA	Edge Rectangle Absolute
EP	Edge Polygon
ER	Edge Rectangle Relative
EW	Edge Wedge
FP	Fill Polygon
PM	Polygon Mode
RA	Fill Rectangle Absolute
RR	Fill Rectangle Relative
WG	Fill Wedge

Table 17-4 The HP-GL/2 Commands by Group (4 of 5)

CHARACTER GROUP	
AD	Alternate Font Definition
CF	Character Fill Mode
CP	Character Plot
DI	Absolute Direction
DR	Relative Direction
DT	Define Label Terminator
DV	Define Variable Text Path
ES	Extra Space
FI[1]	Select Primary Font
FN[1]	Select Secondary Font
LB	Label
LO	Label Origin
SA	Select Alternate Font
SB	Scalable or Bitmap Fonts
SD	Standard Font Definition
SI	Absolute Character Size
SL	Character Slant
SR	Relative Character Size
SS	Select Standard font
TD	Transparent Data

1. These commands are part of HP-GL/2's Dual Context Extensions.

Table 17-5 The HP-GL/2 Commands by Group (5 of 5)

LINE AND FILL ATTRIBUTES GROUP	
AC	Anchor Corner
FT	Fill Type
LA	Line Attributes
LT	Line Type
PW	Pen Width
RF	Raster Fill Definition
SM	Symbol Mode
SP	Select Pen
SV[1]	Screened Vectors
TR[1]	Transparency Mode
UL	User-defined Line Type
WU	Pen Width Unit Selection

1. These commands are part of the Palette Extensions to HP-GL/2.

As shown in the tables above, each HP-GL/2 command is a two-letter mnemonic code designed to remind you of its function. For example, IN is the Initialize command, SP is the Select Pen command, and CI is the Circle command. Parameters are used with certain HP-GL/2 commands to tell the printer to complete the command in a particular way.

Understanding HP-GL/2 Syntax

HP-GL/2 commands have four components: a mnemonic, parameter(s), separator(s), and a terminator. Refer to the following illustration of a typical HP-GL/2 command and the description of its components.

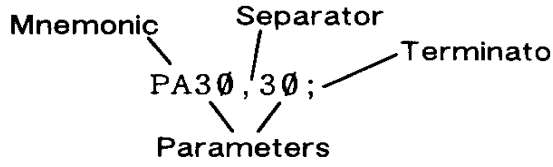

Figure 17-1 Typical HP-GL/2 Command

- **Mnemonic**--The two-letter mnemonic reminds you of the command's function. The mnemonic can be uppercase or lowercase.
- **Parameter(s)**--Some commands have no parameters; for those commands which have them, parameters can be either required or optional (as indicated in the description of that command).
- **Separator(s)**--When you use parameters, you must separate them with a comma or space, or in the case of a numeric parameter, with a + or - sign. (Commas are recommended because some computers eliminate spaces, especially when sending variables.)
- **Terminator**--All commands require a terminator. Most HP-GL/2 commands are terminated by a semicolon or the first letter of the next mnemonic, a white space, or a tab (exceptions: LB uses a user-defined terminator; PE cannot use the first letter of the next mnemonic). The last command prior to exiting HP-GL/2 mode *must* be terminated with a semicolon.

The following illustration shows the flexibility of the syntax. Each variation of the two-command sequence is permissible; however, the method shown on the left is recommended in most instances. The recommended method uses the first letter of the next mnemonic to terminate commands, uses no space between the mnemonic and its parameters, and separates parameters with a comma. (For clarity, examples in this HP-GL/2 section of the manual use semicolons as terminators, as shown in the middle example below.)

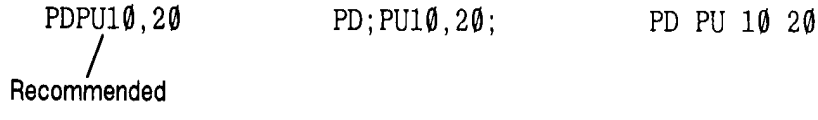

Figure 17-2 Illustration of Syntax Flexibility

The next section explains how the syntax of individual commands is presented.

Notations Used to Express Syntax

The following describes the notations used in the syntax section of each command description:

Mnemonic

For readability, the mnemonic is shown in uppercase and separated from the parameters and/or terminator.

parameters

Parameters are shown in italic.

[ ]

Parameters in square brackets are optional.

[param1,param2...[,param1,param2]]

These optional parameters must be paired.

params...params

These parameters may be given the number of times specified in the command description.

text...text

This parameter indicates that you can type in a range of ASCII characters, such as in the Label (LB) command.

(....)

Indicates that you can use a range of the previous parameter; however, all X coordinates must have a corresponding Y coordinate.

Note Remember that while X,Y coordinates are shown in parentheses in text [for example (3,4) or (0,0)], the parentheses are not part of the syntax. Do not enter these parentheses in your commands.

;

iCommand terminator. In most HP-GL/2 commands, a semicolon is optional, and is shown in parentheses in most command syntax.

Notes Three exceptions to the optional use of the semicolon as a command terminator occur in the following commands: Polyline Encoded (PE), Label (LB), and Comment (CO).

PE must be terminated by a semicolon. LB is terminated by the non-printing end-of-text character (ETX - decimal 3), or a user-defined character. The comment string of the CO command must be delimited by double quotes.

A semicolon terminator is **always required** following the last command prior to leaving HP-GL/2 mode.

,

A comma is always shown as the separator between parameters. A space, +, or - is also valid (although not preferred). (A + or - is a valid separator only for numeric parameters.)

Omitting Optional Parameters

Some commands have optional parameters that take on default values if they are omitted. When you omit a parameter, you must omit all subsequent parameters in the same command (the Define Label Terminator (DT) command is an exception).

For example, the Line Type (LT) command has three optional parameters: type, pattern length, and mode. The following command shows all three being used (*type* = 6, *pattern length* = 25, *mode* = 1).

```
LT6,25,1
```

If you omit the second parameter you must also omit the third parameter, as shown below:

```
LT6
```

The printer uses the most recently specified pattern length and mode. If you have not specified a length or mode since sending a Default Values (DF) or Initialize (IN) command, the printer uses the parameter's defaults.

For example, if you send the following command (omitting the second parameter), the printer interprets the "1" as the second parameter:

```
LT6,1
```

Parameter Formats

You must give parameters in the format (type of units) required by each HP-GL/2 command. The required format is stated in the parameter table of each command's description, and is described as follows.

1 *Integer*—An integer from –1,073,741,823 ($-2^{30}+1$) to 1,073,741,823 ($2^{30}-1$). The printer automatically rounds fractional parameters to the nearest integer within the range. Sending a number outside the parameter range may produce unexpected results.

2 *Clamped Integer*—An integer from –32,768 (-2^{15}) to 32,767 ($2^{15}-1$). The printer automatically rounds fractional parameters to the nearest integer. Sending a number outside this range does not cause an error, but the number is "clamped" to the limits of the range. For example, when parsing a clamped integer, the printer treats all numbers above 32,767 as 32,767.

Certain commands have parameters which are restricted to a smaller range. These ranges are listed in the parameter tables for each command. Sending a number outside the reduced parameter range may produce unexpected results.

3 *Real*—A number with an integer portion from –1,073,741,823 ($-2^{30}+1$) to 1,073,741,823 ($2^{30}-1$). You are assured of at least 6 significant digits (including integer and fractional portion). You may omit the decimal point when no decimal fraction is specified. Sending a number outside the parameter range may produce unexpected results.

4 *Clamped Real*—A number with an integer portion from –32,768 to 32,767; you are assured of at least 6 significant digits (including integer and fractional portion). You may omit the decimal point when no decimal fraction is specified. Sending a number outside this range does not cause an error, but the number is “clamped” to the limits of the range. For example, the printer treats all numbers above 32,767 as 32,767.

   Certain commands have parameters which are restricted to a smaller range. These ranges are listed in the parameter tables for each command. Sending a number outside the reduced parameter range may produce unexpected results.

5 *Label*—Any sequence of characters. In the HP-GL/2 language, text is described using the term “label.” Refer to the Label (LB) command in Chapter 23 for a complete description.

Note

Numbers within the above-mentioned ranges do not cause errors; however, the range may exceed the printer’s physical printing area. Numbers that move the pen position outside the *effective window* result in image clipping. This topic is discussed in more detail later in this chapter under “The Vector Graphics Limits.”

When you see the term “current units” in a parameter table, the unit system of that parameter depends on whether scaling is on or off. When scaling is on, the units are user-units; when scaling is off, the units are plotter units (described under “Units of Measure” later in this chapter).

Notes

The printer cannot use exponential format numbers (for example, 6.03E8). If you are using a computer or language that uses the exponential format, you must use integer variables or a formatting technique to output fixed-point real numbers.

Parameter values less than the range maximum are passed by the parser; these values may subsequently be unscaled into resolution units (e.g. 7200 units-per-inch) that exceed the device-dependent internally representable number range. If this occurs, the device enters a LOST mode; all relative drawing commands are ignored until a command is received which specifies an absolute move to a point within the internally representable number range.

Notes

When LOST mode is entered, the pen is raised and the following commands are ignored: AA, AR, AT, CI, CP, EA, ER, EW, LB, PE, PM, PR, RA, RR, RT, and WG.

The commands allowed in LOST mode are: AC, AD, CF, CO, DF, DI, DR, DT, DV, ES, FT, IN, IP, IR, IW, LA, LO, LT, PA, PD, PG, PU, PW, RF, RO, RP, SA, SB, SC, SD, SI, SL, SM, SP, SR, SS, TD, UL, WU, and the PM1/PM2 forms of PM.

The commands IN, PG, RP, and PA, with in-range parameters, clear LOST mode, PD and PU in absolute plotting mode, with in-range parameters, also clear LOST mode. When PD clears LOST mode, a line is drawn from the last valid current position to the first point in the PD parameter sequence. If PA clears LOST mode, the pen will not go down until a PD command is received.

Using HP-GL/2 With Programming Languages

The HP-GL/2 examples included in this manual are given in a “generic” format (they show the commands required to perform a specific function but usually do not use a specific programming language). In most cases, the commands are accompanied by a brief description of the command being used.

To see how HP-GL/2 commands are used in BASIC and the C programming language, see the following examples.

Example:BASIC

This example uses BASIC to print three lines forming a simple triangle (shown below).

```
10 LPRINT CHR$(27);“E”; :REM Reset the printer
20 LPRINT CHR$(27);“%0B”; :REM Enter HP-GL/2 Mode
30 LPRINT “IN”; :REM Initialize HP-GL/2 Mode
40 LPRINT “SP1PA10,10”; :REM Select Pen & move to 10,10
50 LPRINT “PD2500,10,10,1500,10,10;”; :REM Pen down & draw
60 LPRINT CHR$(27);“%0A”; :REM Enter PCL Mode
70 LPRINT CHR$(27);“E”; :REM Reset to end job/eject page
```

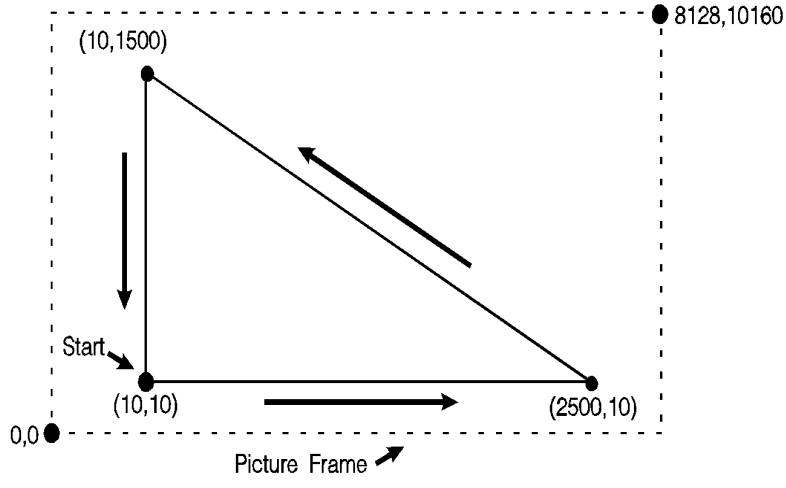

Figure 17-3

Example:C Programming Language

This example uses the C programming language to print the same three lines shown on the previous page.

Table 17-6

```
#include <stdio.h>

main()

{

FILE *prn;
prn = fopen("PRN","wb");                /* open the printer */
fprintf(prn,"033E");                    /* Esc E to reset printer */
fprintf(prn,"033%%>0B");                /* Enter HP-GL/2 */
fprintf(prn,"IN");                      /* Initialize HP-GL/2 Mode */
fprintf(prn,"SP1PA10,10");              /* Select pen 1 & move to 10,10 */
fprintf(prn,"PD2500,10,10,1500,10,10;"); /* Pen down & draw */
fprintf(prn,"033%%0A");                 /* enter PCL at previous CAP */
fprintf(prn,"033E");                    /* Reset to end job/eject page */
}
```

The HP-GL/2 Coordinate System

Both PCL and HP-GL/2 use a Cartesian Coordinate System. The Cartesian coordinate system is a grid formed by two perpendicular axes, usually called the X-axis and Y-axis (refer to Figure 17-4). The intersection of the axes is called the origin of the system and has a location of (0,0). The default HP-GL/2 coordinate system is different than the PCL coordinate system; +Y is down in PCL and up in HP-GL/2. In addition, the default origin is at the lower left in HP-GL/2 and at the upper left in PCL.

Note

The HP-GL/2 coordinate system can be set up to match the PCL coordinate system. See the example entitled “Adapting the HP-GL/2 Coordinate System to Match the PCL System” in Chapter 19.

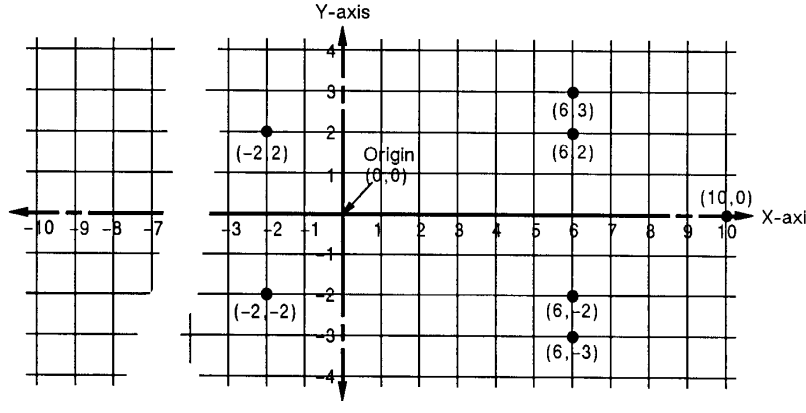

Figure 17-4 The HP-GL/2 Coordinate System

To locate any point on the grid (the printing area within the PCL Picture Frame), move from the origin a number of units along the X-axis, then move a number of units parallel to the Y-axis. The number of units you move matches a coordinate location. Each point is designated by the combination of its X-coordinate and Y-coordinate, known as an X,Y coordinate pair. In , positive X values are plotted to the right of the origin, and positive Y values are plotted above the origin.

Study Figure 17-4 to locate these points: (0,0); (-2,2); (6,2); (6,3); (10,0); (6,-3); (6,-2); (-2,-2); (0,0). Draw a straight line between each point in the order listed. (You should have drawn an arrow.) This is a simple demonstration of defining a vector image when in HP-GL/2 mode.

Note

To specify a point when programming an application, you must always give a complete X,Y coordinate pair; the X coordinate is first and the Y coordinate second. This manual shows coordinate pairs in parentheses (X,Y) for clarity. Do not use parentheses in your command sequence.

Using the default HP-GL/2 coordinate system, the origin is in the lower left corner of the PCL Picture Frame, as shown in Figure 17-5. Using the IP or IR commands, you can move the origin to other locations. Then, using the SC command, you can define practically any unit coordinate system. (This process is discussed in more detail later in this chapter under “Scaling,” and also in Chapter 19.)

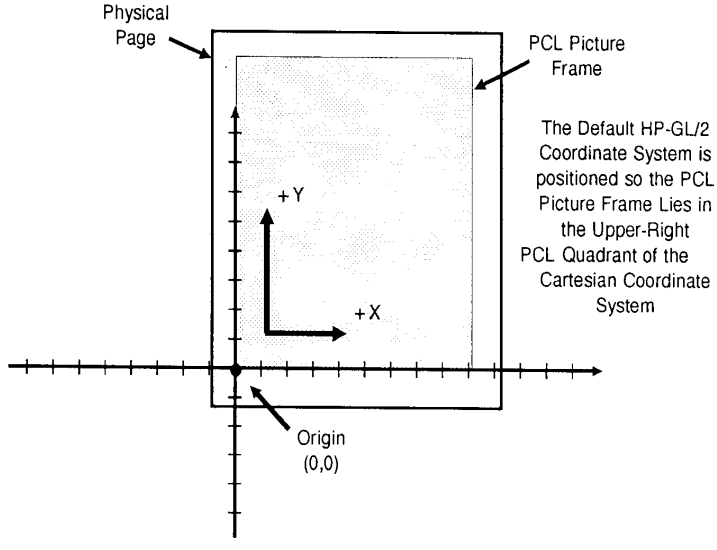

Figure 17-5 The Default HP-GL/2 Coordinate System

HP-GL/2 & PCL Orientation Interactions

The relationship between the orientation of the HP-GL/2 coordinate system and the PCL coordinate system is important. Figure 17-6 illustrates this relationship for the default HP-GL/2 orientation (RO 0) and the PCL logical page orientation. As shown in the illustration, in the default HP-GL/2 orientation, the origin of the HP-GL/2 coordinate system defaults to the lower-left corner of the PCL Picture Frame. (HP-GL/2 and PCL X-coordinates increase in the same direction, but the Y-coordinates increase in opposite directions.) Notice that a change in the PCL logical page orientation changes the orientation of the PCL coordinate system and the HP-GL/2 coordinate system.

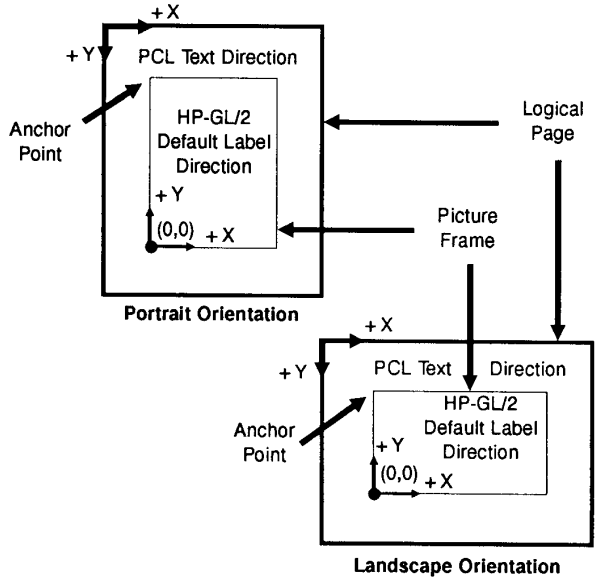

Figure 17-6 Orientation Interactions Between PCL and HP-GL/2

The relationship between the coordinate systems can be changed using the HP-GL/2 Rotate (RO) command. Rotations specified by the RO command are relative to the default HP-GL/2 orientation (which matches the PCL orientation). Figure 17-7 shows how the RO command modifies the default HP-GL/2 orientation.

Note A change in PCL print direction has no effect on the HP-GL/2 orientation, the physical position of the picture frame, or the picture frame anchor point.

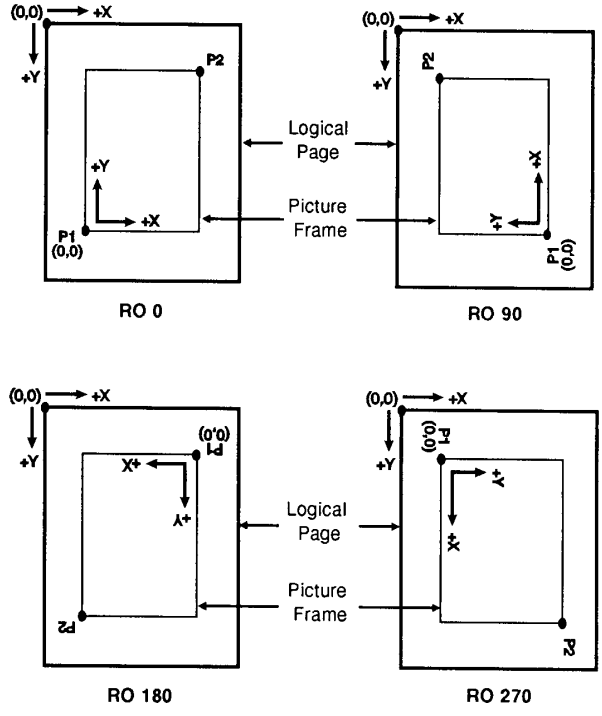

Figure 17-7 Modifying HP-GL/2 Orientation on a Portrait Page

The Vector Graphics Limits

The area on the page where a vector graphics image can be printed is determined by the intersection of the following four boundaries:

- Hard-clip Limits
- Soft-clip Window
- PCL Logical Page
- PCL Picture Frame

The *hard-clip limit* refers to the boundaries resulting from the physical limits of the printer (in PCL mode, this is referred to as the *printable area*). The *soft-clip limit* refers to the area defined using the HP-GL/2 *Input Window* (IW) command. The intersection of all these areas is the *effective window.* An HP-GL/2 graphic appears on the page only if it falls within the effective window.

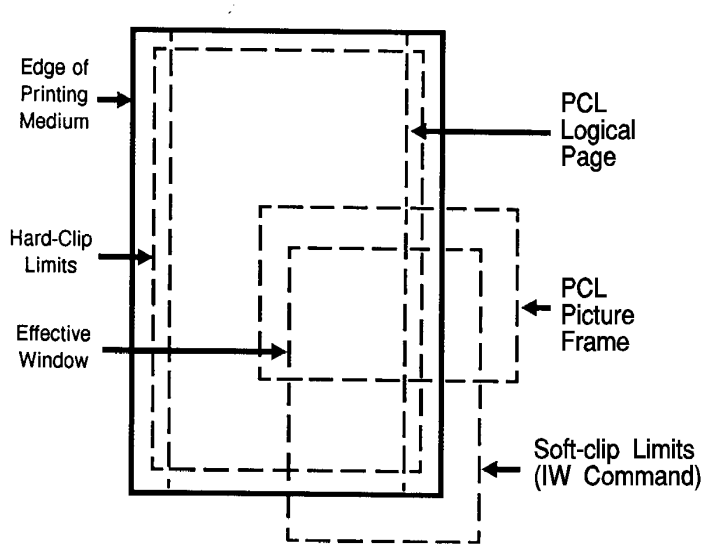

Figure 17-8 The Effective Window

Note For more information on the PCL coordinate system and the PCL 5 printer's printable limits, see Chapter 2.

HP-GL/2 Units of Measure

In HP-GL/2 mode, you can measure along the X,Y axes and express coordinates using two types of units: *plotter units* and *user-units*.

Plotter Units

One *plotter unit* equals 0.025 mm. When specifying distances in plotter units, the printer converts the number of plotter units to equivalent dot coordinates before printing. Under default conditions, the printer uses plotter units.

The following table lists equivalent measurements for plotter units.

Table 17-7

PlotterUnits	EquivalentValue
1 plu =	0.025 mm (≈ 0.00098 in.)
40 plu =	1 mm
1016 plu =	1 in.
3.39 plu =	1 dot @ 300 dpi

User-units

The size of units along the X and Y axes may be redefined using the Scale (SC) command. User-units allow you to customize the coordinate system to represent any value. For example, you could plot the moon cycle for the year by dividing the X-axis into 31 units for days of the month and the Y-axis into 12 units for months of the year. To mark a point on December 25, you would give the coordinate (25,12) rather than calculating the exact location in plotter units.

Before printing, the printer internally converts user-units to dot locations.

Pen Status and Location

Since printing vector graphics has traditionally been performed with plotters, the terms *pen* and *pen position* are used to described the HP-GL/2 cursor, the current active position (CAP) when in HP-GL/2 mode. Like a physical pen, this imaginary pen must be selected if you want to draw images. Commands such as Pen Up (PU) or Pen Down (PD), and phrases such as “current pen position” or “moving the pen” apply to the imaginary pen just as they would a physical pen on a plotter.

Pen Status

Pen status refers to whether the “pen” is up or down. Use the Pen Up (PU) command with X,Y coordinates to move the pen to the desired printing location without drawing a line. Use the Pen Down (PD) command with X,Y coordinates to lower the pen and begin drawing from the current location to the first specified X,Y coordinate.

Upon entering HP-GL/2 mode for the first time following a reset ($^{E}_{C}$E) command, no pen has been selected and the pen is up. *This means that no lines are drawn when HP-GL commands are given until a pen is selected.* This can be done using the Select Pen (SP) command.

Most drawing commands require that the pen be lowered to produce marks on the page. Once lowered with a Pen Down (PD) command, the pen remains down for subsequent HP-GL/2 printing commands until a Pen Up (PU) or Initialize (IN) command is issued. The pen remains selected until a new SP command is received. You must be aware of the pen’s up/down status to avoid drawing stray lines between parts of your picture.

Note

Upon entry into HP-GL/2 mode, a good programming practice is to select a pen and command a pen-up move to the initial starting position. This ensures that a pen is selected and is in the proper position to begin drawing.

Every time you use a PU or PD command, the printer updates the pen up/down status. The following table shows the commands that include an automatic PD command as part of their function. After performing their complete function, they return the pen to its previous up/down state.

Table 17-8 Commands That Include an Automatic Pen Down

Command		Group
CI	Circle	*The Vector Group*
EA	Edge Rectangle Absolute	*The Polygon Group*
EP	Edge Polygon	
ER	Edge Rectangle Relative	
EW	Edge Wedge	
FP	Fill Polygon	
RA	Fill Rectangle Absolute	
RR	Fill Rectangle Relative	
WG	Fill Wedge	
LB	Label	*The Character Group*
SM	Symbol Mode	*The Line and Fill Attributes Group*

Notes

Whenever the printer receives a Pen Down command, it produces a dot at the current pen location. If the pen is already down when the printer receives a command with an automatic Pen Down, the unnecessary dot can mar your final output. For best results, include a Pen Up (PU) command before any command with an automatic Pen Down.

Only the portion of the pen falling within the effective window is printed. The pen is centered on a line between the beginning and end points, with half of the pen width falling on either side of this line.

The definition of each command tells you whether it has an automatic pen down. If you find that part of your image is not drawn, make sure your command sequence uses the PD command before the affected commands.

Pen Location

Pen location refers to the X,Y coordinates of the current active position (CAP — the point at which the next HP-GL/2 command begins). Most commands, when completed, update the pen location. The next command then begins at that location. Some commands do not update the current pen location. The definition of each command tells you whether the current pen location is updated or restored. Use the Pen Up (PU) command with the desired X,Y coordinates to lift the pen and move it to a new location.

The Default Values (DF) command does not reset the current pen location; the Initialize (IN) command moves it to the lower-left corner of the PCL Picture Frame. You should specify your beginning pen location for each HP-GL/2 drawing.

Scaling

When you *scale* a drawing, you define your own units of measurement instead of using plotter units; the printer converts your units (*user-units*) to dot positions for placing the image on the page. *Scaling* allows control of the printer using units that are easy for you to work with.

For example, you can scale your drawing to divide the drawing area into 100 squares. As you plan the drawing, you can think in terms of 100 squares rather than plotter units. Here is another example of scaling: since 400 plotter units equals 1 centimeter, you can establish this scale to print in user-units equal to 1 centimeter each.

Scaling begins with the scaling points, P1 and P2. P1 and P2 act as two points marking opposite corners of a rectangle. You can make this rectangle any size and place it anywhere in relation to the origin, depending on the plotter unit coordinates you specify for P1 and P2. (P1 and P2 default to the lower left and upper right corners of the picture frame, respectively, but you can change their locations using the Input P1 and P2 (IP) or Input Relative P1 and P2 (IR) commands.)

After you have defined the positions for P1 and P2, or have accepted the default, use this imaginary rectangle to set up scaling for your drawing. With the Scale (SC) command you specify how many sections the rectangle divides into horizontally (the X-axis) and how many sections the rectangle divides into vertically (the Y-axis). With this process you have created your user-units.

Scaling also allows you to enlarge or reduce your image by changing the locations of P1 and P2. P1 and P2 represent physical locations in relation to the PCL Picture Frame. When the imaginary rectangle formed by P1 and P2 is enlarged or reduced with the IP or IR commands, the HP-GL/2 image is also enlarged or reduced to fit the new P1/P2 rectangle. (For a more detailed explanation of scaling and the Scale (SC) command, see Chapter 19.)

For importing existing HP-GL/2 images, another method of enlarging or reducing drawings exists. It involves varying the size of the PCL Picture Frame and is described next. This method allows you to scale an image while maintaining the aspect ratio of all elements (including fonts). The Scale command does not affect the size of fonts.

Absolute and Relative Pen Movement

The Plot Absolute (PA) and Plot Relative (PR) commands allow you to set whether you want to draw using absolute or relative “pen” moves. *Absolute* pen movement uses X,Y coordinates to specify an exact, fixed point relative to the origin (0,0). In Figure 17-9, the coordinates (3,8), (5,4), and (8,1) are always in the same place with respect to the origin, no matter where the pen is when the coordinates are issued.

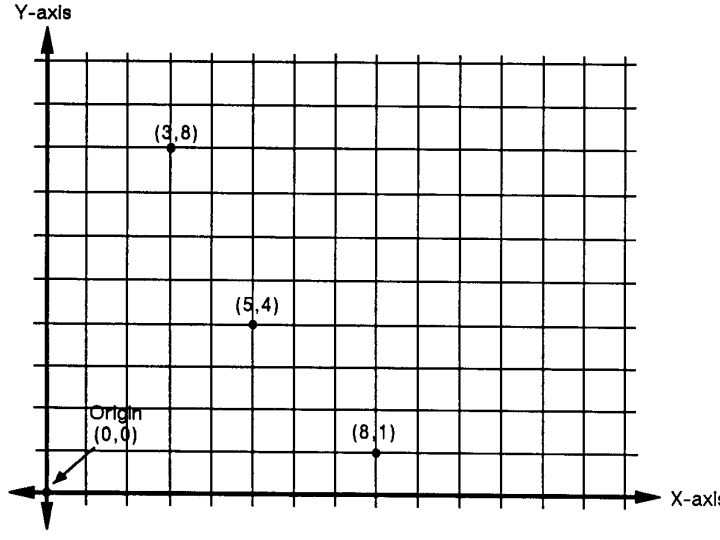

Figure 17-9 Absolute Coordinates

Relative pen movement uses X,Y *increments* to specify the number of units the pen moves from its current pen location. All commands that use relative increments include “relative” in their name (except the PE command). (An example is the Edge Rectangle Relative (ER) command.

In Figure 17-10 for example, assume that the pen is currently at the origin (0,0). To move to the absolute points shown in Figure 17-9 using relative coordinates, count 3 units to the right and 8 units up from the current pen location; these are both positive directions with respect to the origin. This is the relative location (3,8). Now move 5 positive X-units and 7 negative Y-units from this location to the lower point; this is the relative location (5,-7). From this location, move to the last point by moving 3 negative X-units and 3 positive Y-units (-3,3).

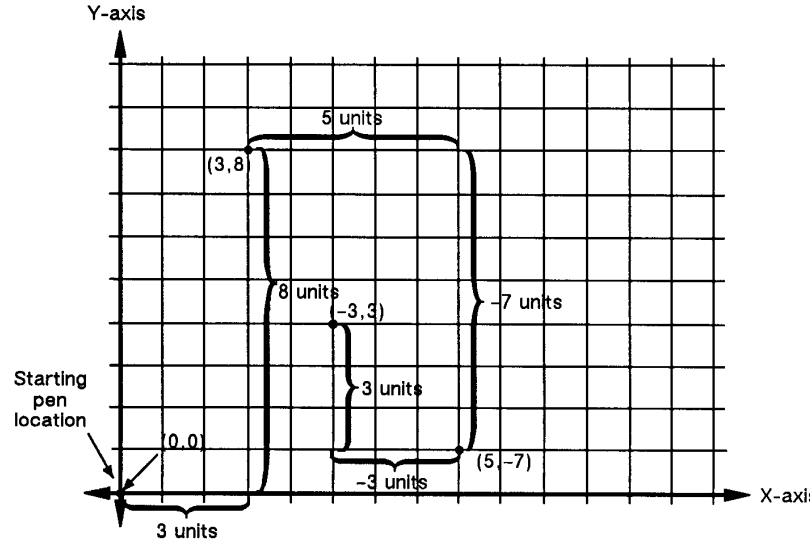

Figure 17-10 Relative Coordinates

Relative movement is useful in many applications where you know the dimensions of the shape you want, but do not want to calculate the absolute coordinates. For example, if you want a box 4 X-units by 8 Y-units, you can use the Edge Rectangle Relative (ER) command to draw the box without having to calculate the absolute coordinates of the opposite corner. (The ER command draws a rectangle using the current pen location as one corner, and the specified relative coordinates as the opposite corner.)

Absolute pen movement is the default mode; coordinates received within a PU (Pen Up) or PD (Pen Down) command are interpreted as absolute plotter units unless a PR (Plot Relative) command establishes relative mode. As with absolute coordinates, the relative units can be either user-units or plotter units, depending on whether the SC command is in effect.

Note

Relative increments add to the current pen location. The printer automatically converts the new relative location to absolute coordinates and updates the current pen location. Using relative coordinates can be faster in cases where the I/O speed limits your print speed, since relative coordinates are generally smaller numbers and therefore transmit less data over the I/O.