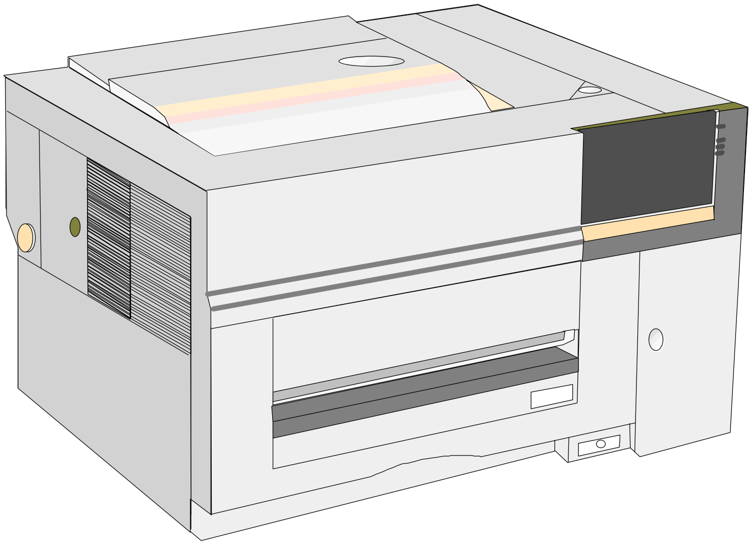
Developers Guide To The HP COLOR LASERJET



Chapter 1: Developer Information

Developer information

- Superset of LaserJet IIISi, IIIP, and DeskJet 1200c
- Font-compatible with the LaserJet 4.
- Extended 300 dpi PCL5 C support.
- PJL (Printer Job Language) for I/O and language switching and status/ustatus
- HP-GL/2 with dual context and palette extensions.
- Resolution enhancement (RET).
- Letter, executive and A4 color printing.
- Letter, A4, legal, A3, executive, tabloid black and white printing.
- No Envelope support.
- Plain paper, and transparency film (Special).
- 8 Mbytes resident memory, with up to 72 Mbytes more optional.
- Supports 16Mbyte SIMMs.
- Built in Centronics port and one MIO 5.2 with auto-sensing I/O (both I/Os hot).
- Optional JetDirect network interface cards: Novell, DLC, TCP/IP, EtherTalk in Ethernet or Novel, DLC in TokenRing.

Most black and white HP Color LaserJet commands are described in the following documents in your Driver Developer's Kit:

- *The PCL5 Printer Language Technical Reference Manual*
- *The PCL5 Comparison Guide*
- *PJL technical reference manual*

Most color commands are described in the following documents:

- *The PCL5 Color Extensions Technical Reference Manual*
- *The HP Color LaserJet Developers Guide.*

This Developer's Guide describes PCL5 color and other commands that have been modified or are of special interest for The HP Color LaserJet developers.

DeskJet 1200C Compatibility

From a language perspective, The HP Color LaserJet is a Seabreeze with part of LaserJet 4 PJL commands and increased control over the color qualities of the printer. To build a HP Color LaserJet driver from a Seabreaze, perform the following steps:

1. Add the PJL commands
2. Add any new color rendering support desired, for instance, a 24 bit color path, gamma correction, HP Color LaserJet halftones, and color lookup tables.
3. remove any media type, and print quality commands.
4. remove color support for page sizes greater the A4 (ledger and legal) for color documents.
5. remove any envelope support.
6. The printer can perform scaling and halftoning for your graphics object

LaserJet Compatibility

The PCL5 Comparison Guide contains a PCL Feature Support Matrix for the LaserJet III and 4 series printers.

LaserJet 4 Compatibility

To build a HP Color LaserJet driver from a LaserJet 4 driver, perform the following steps:

1. Support HP Color LaserJet's media size selections. Add ledger size support for Black and White.

2. Add 8-bit color (*Esc*v#W*) to text, raster graphics, and HP-GL/2 objects.

Note: *Adding color to a driver is no easy task. For each graphic object color information must be known. This information includes color, rendering algorithm, palette index, gamma correction, and any color correction information passed to the printer via Color lookup tables. The complexitiy of your driver will increase.*

LaserJet IIIP and IIISi Compatibility

To build a HP Color LaserJet driver from a LaserJet IIIP driver, perform the following steps:

1. Add the LaserJet 4 fonts, including the 35 Intellifont and 10 TrueType fonts.

2. Support HP Color LaserJet's media size selections. Add ledger size support for Black and White.

3. Add 8-bit color (*Exc*v#W*) to text, raster graphics, and HP-GL/2 objects.

Note: *Adding color to a driver is no easy task. For each graphic object color information must be known. This information includes color, rendering algorithm, palette index, gamma correction, and any color correction information passed to the printer via Color lookup tables. The complexitiy of your driver will increase.*

Note that the LaserJet 4, IIIP and The HP Color LaserJet contain the following commands for downloading symbol sets that are not available on the LaserJet IIISi:

Symbol Set ID	*Esc*c#R*
Symbol Set Control	*Esc*c#S*
Define Symbol Set	*Esc(f#W[data]*

Incompatibilities

- The HP Color LaserJet does not support Compression Method 9 (*Esc*b#M*) and will not print correctly if data is sent from a DeskJet 550C and prior driver using compression method 9.

- The HP Color LaserJet does not recognize the format for DeskJet 550C and prior soft fonts.

- The HP Color LaserJet does not support 600 dpi (PCL Units) commands.
- The HP Color LaserJet does not support envelopes.
- The HP Color LaserJet supports color on media sizes A4 and smaller.
- The HP Color LaserJet does **not** supports color on media sizes larger that A4.

Developer Guidelines

Job Control

- Support PJL (Printer Job Language) for language switching. Begin and end all jobs with the Universal Exit Language PJL command (*Esc%-12345X*). The UEL command resets the printer by executing Esc E.

- Never terminate a job in the middle of data transfer, such as *Esc*b#W[data]*.

Media

- Support all HP Color LaserJet's media types and sizes.

High-level Objects

- Utilize PCL5's ability to handle patterns, rectangular area fills, and vector graphics.

- Use Foreground Color (*Esc*v#S*) to colorize text, rules, patterns, and black and white raster objects.

- Use logical operations (*Esc*l#O*) for MS Windows imaging compatibility.

Fonts

- Select the appropriate symbol set. Support PC-8 and Roman-8 character sets in the US; and PC-850, ECMA 94 and PC-8 Danish/Norwegian in Europe. (See *The PCL5 Comparison Guide*)

- Support The HP Color LaserJet Intellifont and TrueType scalable fonts and downloadable symbol sets.

- Support Autofont/TFM (Tagged Font Metrics).

Programmable Color

- Use 8-bit programmable color (*Esc*v#W*) for text, raster, and HP-GL/2 objects. 8-bit indexed color tends to compress better than 3-bit indexed color, and it allows the printer to do the rendering and scaling.
- Use the printer's internal rendering (*Esc*t#J*) to optimize color quality and utilize different dithering algorithms, scaling, and color controls.
- Optimize color gamuts with white/black reference adjustment (*Esc*v#W*) and gamma control (*Esc*t#I*).

Optimizing

- Use raster compression methods 1-3 (*Esc*b#M*) and the PE command in HP-GL/2 for smaller output files and faster printing. Compression methods will significantly improve throughput.
- Use high-level HP-GL/2 objects, such as polygons, with a defined fill type. This will produce smaller output files than rasteriztion. Use the PE command for data compression.
- Use macros to store frequently used templates. This will reduce memory usage. (See *The PCL5 Technical Reference Manual*)
- Always send the Source Raster Width command (*Esc*r#S*) before sending raster to avoid storing unnecessary data.
- When scaling down to a smaller image, use the printer's internal scaling or send raster at 150 dpi. This will significantly reduce the data sent. (See *The PCL5 Technical Reference Manual* for HP-GL/2 scaling.)
- Rather than using raster for gradient fills, create a series of rectangles that gradually increment the colors. The file size will be substantially reduced.
- Use the printer's internal rendering (*Esc*t#J*) to quickly optimize color quality and utilize different dithering algorithms.
- Use a formfeed instead of *EscE* between pages of a multi-page job. This is more efficient than resetting after every page.

Colorizing the PCL5 Page -- An Example

The following is an example of how to colorize a PCL5 page.

1. Start from an existing LaserJet 4 driver like 4P or 4si.

2. Build a shared RGB palette for both PCL and HP-GL/2.

For example, set up a business palette to match your Super VGA display. Since most charts look better with saturated colors, stay with primary and secondary colors to avoid dithered colors in text and bar charts (specks in small text are distracting).

Esc*v6W0,0,8,8,8,8	Build the palette with 256 colors — each entry is 24-bit RGB.
Esc*v0a0b0c0I	The zero entry has no color components; it is solid black.
Esc*v255a0b0c1I	The next entry has only a full red component; it is solid red.
Esc*v255a255b255c255I	The last entry has all color components; it is solid white.

3. Go into HP-GL/2 to build your bar or pie chart.

Symbolic description of synthetic graphs and clip art saves download time and memory, and is more resolution-independent.

Esc%2B	Go into HP-GL/2 with the same coordinate system as PCL.

Moving in HP-GL/2 user units is like moving in PCL dots on a PCL page. Use wedge commands to assemble your pie chart. Remember: selecting pen *n* in HP-GL/2 is the same as selecting foreground color *n* on the shared palette.

SP 1	Pick the red pen
PU 300, 300; PD	Absolute dot move to upper left part(1",1") of the PCL page.
WG 300, 270,90	Fill a 1" quarter wedge in lower right quadrant with red.
Esc%1A	Go back to PCL.

If you want to embellish, add an outline rectangle and some fancy font labeling. If you like using the shared palette, avoid using IN because it will create a color palette compatible with older plotter applications.

To save your business palette, "push" it with the push/pop command.

Esc*p0P	Save this palette for later use

4. Format the text report. Crisp black text on white paper provides maximum contrast for readability. One escape sequence can always get you back to a black/white palette.

Esc*r1U	Make a black/white LaserJet palette.
Esc*v1S	From the palette, choose black as the foreground color.
...	Insert your favorite LaserJet driver here to lay out the text.

5. After filling in the text, set up a 24-bit/pixel scanned natural image. A portrait of your face would be nice so people would remember who wrote the report. Handle continuous-tone images carefully: fleshtones are especially noticeable.

Start by setting up 24-bit pixel encoding of a direct raster bitmap. Sending more detail lets the printer technology make optimal tradeoffs of color and tone. Halftoning techniques like error diffusion provide near-photographic effects on a scanned bitmap.

Esc*v6W0,3,0,8,8,8	Configure 24-bit direct pixel encoding — no palette indexing.
Esc*r4J	Render the raster image using error diffusion.

You don't have to wait for a 24 Megabyte full page image to trickle through your PC Centronics port. Most multi-bit images should be scanned at 150 dpi or 100 dpi for optimal transmission time and excellent quality. For example, a 100 dpi bitmap would be reduced 9X, to only 2.7 Megabytes. The printer can scale the raster to its native resolution and then halftone the image. This technique creates a more device independent image that looks better...even on printers with different native resolutions.

Esc*t100R	Original resolution is 100 dots/inch.
Esc*r200s300T	Original raster width and height is 2" x 3"
Esc*t1440h2160V	Set new image raster width and height to 2" x 3"

You can still use raster compression techniques to send the raster. For continuous tone images, gradual color changes will help compress the image well. So a 24 bit 100 dpi image is the same original size as a 3 bit 300 dpi image; but it may actually compress better and provide better print quality. Plus device independence, sounds great!

Esc*b3M	Use Compression Method 3.
Esc*r3A	Start raster mode with printer scaling enabled.
Esc*b600W$r_1,g_1,b_1,...r_{200},g_{200},b_{200}$	Send each raster row in pixel order.
Esc*rC	End raster mode.

6. Restore the business palette, and put on the company logo.

Esc*p1P	Pop the business palette; it still has the same 256 colors.
Esc%2B	Go to HP-GL/2 to draw the company logo.
...	Draw the logo in HPGL/2: business colors are available.
Esc%A	Go back to PCL.

7. Return to the black and white LaserJet palette for the next job.

Esc*r1U	The new palette destroys the business palette.
EscE	Reset can also restore the B/W LaserJet palette.

Optimizing the Color PCL5 Page

1. Minimize language and palette swaps. In the above example, do the business chart and logo once, while you are in HP-GL/2 with the programmed palette. Then do all the text layout with the black/white palette.

2. Save large programmable palettes (push/pop palette or palette ID's). Simple color and black and white palettes — which are small and non-programmable — need not be saved.

3. Use direct pixel raster transfer and raster scaling for continuous-tone images. Error diffusion or device best improves quality on most scanned raster images. Scatter give good image quality and increases speed. Experiment with the different render algorithms for optimal quality and speed.

4. Cache host-rasterized fonts as soft fonts, so characters need not be sent more than once. This saves transmission time and printer memory — especially when several characters are reused on a page. Higher printer resolutions make this more imperative.

5. Use vector graphics rather than bitmaps. This creates more resolution-independent page descriptions. It is also faster and more compact in most cases.

6. Use raster logical operations (ROPS) to achieve arbitrary clip paths if you need color fills for arbitrary shapes. For example, a three-step process can be used with a rectangular color-fill bitmap.

1. Place the original bitmap on the page with an XOR operation.
2. Draw a black character or polygon shape within the rectangular area.
3. Merge the original bitmap to the page with an XOR operation. This step erases the original bitmap except for areas touched by the arbitrary shape — your printer does the clipping instead of the application.

7. Use PCL macros to reduce download time. For example, store the above rectangular bitmap as a macro. Instead of downloading the bitmap twice, you execute the macro twice. This really reduces download time if you apply the same fill to several objects.

Esc&f1Y	Define the color fill raster bitmap as PCL macro 1.
Esc&f0X	Start macro.
...	Download the color fill raster bitmap normally.
Esc&f1X	End macro.
*Esc*v1N*	Neutralize PCL print model — source opaque.
*Esc*v1O*	Neutralize PCL print model — pattern opaque.
*Esc*l102O*	XOR the color bitmap source to page destination.
Esc&f2X	Apply bitmap by executing macro.
*Esc*l252O*	Restore default ROP, then fill a polygon or text normally.
*Esc*l102O*	XOR the color bitmap source to page destination.
Esc&f2X	Erase the bitmap by executing the macro.

DOS Color Matching

Since RGB screen values do not correspond to RGB printer values, printed colors and monitor colors will look different. However, customers may desire printer-to-screen color matching. The Driver Developer Kit contains an interpolation program and color matching tables for DOS drivers. (HP provides a Microsoft Windows driver). The interpolation program converts screen RGB colors to The HP Color LaserJet RGB colors. Printed colors will then appear like screen colors.

The interpolation for most monitors in very small changes in color. The device RGB mode of the Configure Image Data command has colors that match closly to many monitors. You may not need to proform any interpolations to get desired output.

The Print Environment

Print environments are described in Chapter 3 of *The PCL5 Printer Language Technical Reference Manual*. Some of the more important default settings (active after power-up or reset) for The HP Color LaserJet are shown below:

- 10-pitch font (Courier)
- Black text
- 80 characters per line
- 6 lines per inch
- 60/64 lines per page (back panel English/metric)
- Perforation skip ON
- Autowrap OFF
- Manual feed OFF
- Symbol set (back panel selectable, i.e., PC-8 or Roman-8)

NOTE: Always start and end PCL jobs with *EscE* and put commands that access desired functions at the beginning of a file, since the printer may be in an unknown state from a previously-used file. Outside of the *Esc E* commands send PJL commands to set language and certian job configurations.

Internal Fonts

The HP Color LaserJet has the same internal fonts as the HP LaserJet 4, as shown in the following tables.

Bitmap Fonts

The HP Color LaserJet has 1 internal bitmap font.

Typeface	Type Style	Pitch	Point Size	Orientation
Line Printer	Medium	16.67	8.5	Portrait & Landscape *

* All characters, including downloaded characters, can be automatically rotated to any of the four orientations.

The bitmap Line Printer font is available in the following 6 resident symbol sets:

Symbol Set ID	Symbol Set Name
8U	Roman-8
0N	ISO 8859/1 Latin 1 (ECMA-94)
10U	PC-8
11U	PC-8 Danish/Norwegian
12U	PC-850
1U	Legal
2N	ISO 8859/2 Latin 2
5N	ISO 8859/9 Latin 5

Intellifont Scalable Fonts

The HP Color LaserJet has 35 resident Intellifont scalable fonts.

Typeface	Type Styles
Courier	Medium, Bold, italic, Bold Italic
CG Times	Medium, Bold, Italic, Bold Italic
CG Omega	Medium, Bold, Italic, Bold, Italic
Coronet	Medium
Clarendon Condensed	Bold
Letter Gothic	Medium, Bold, Italic
Univers	Medium, Bold, Italic, Bold Italic
Univers Condensed	Medium, Bold, Italic, Bold Italic
Antique Olive	Medium, Bold, italic
Garamond Antiqua	Antiqua, Halbfett, Kursive, Kursive Halbfett
Marigold	Medium
Albertus	Medium, Extra Bold

TrueType Scalable Fonts

The HP Color LaserJet has 10 resident TrueType scalable fonts.

Typeface	Type Styles
Arial	Medium, Bold, Italic, Bold Italic
Times New	Medium, Bold, Italic, Bold Italic
Symbol	Medium
Wingdings	Medium

Internal Symbol Sets

The HP Color LaserJet supports the following 34 symbol sets for Intellifont and TrueType fonts.

Symbol Set ID	Symbol Set Name
0D	ISO 60 Norwegian v1
0I	ISO 15 Italian
0N	ISO 8859/1 Latin 1 (ECMA-94)
0S	ISO 11 Swedish: names
0U	ISO 6 ASCII
1E	ISO 4 United Kingdom
1F	ISO 69 French
1G	ISO 21 German
1U	Legal
2N	Latin 2 (ISO 8859/2)
2S	ISO 17 Spanish
5M	PS Math
5N	Latin 5 (ISO8859/9)
5T	Windows 3.1 Latin 5
6J	Microsoft Publishing
6M	Ventura Math
7J	Desktop
8M	Math-8
8U	Roman-8
9E	Windows 3.1 Latin 2
9T	PC-8 Turkish
9U	Windows
10J	PS Text
10U	PC-8
11U	PC-8 Danish/Norwegian
12J	MC Text
12U	PC-850
13J	Ventura International
14J	Ventura US
15U	Pi Font
17U	PC-852 Latin 2
19M	Symbol
19U	Windows 3.1 Latin 1
579L	Wingdings

PCL Commands

The kernel definition of the PCL language provides a large core of features which must be consistently implemented **in total** in all products. (***The HP Color LaserJet does not support 2 - Primary and Secondary Font Quality)***

Kernel features are grouped in five *segments*: movement, rendering, fonts, page presentation, and device control. A detailed listing of the features in each kernel segment is given below. Note that kernel features are those previously included in former "level" classifications I thru III, as well as some appropriate added features. The kernel contains more than half of the escape sequence groups currently supported in the language.

In addition to the commands included in the kernel, some fundamental requirements for fonts are defined. Fonts have not historically been included in the PCL language umbrella, causing a loss of compatibility.

The kernel standard also includes some device default and mode setting expectations. The customer may alter these defaults using the front panel; but standard default expectations increase the consistency of products as they are shipped from the factory.

In summary, the PCL kernel is:

- Linked with the past: PCL III is the base.
- Expanded to include basic font requirements.
- Expanded to include basic defaults and mode settings.

The PCL Kernel

Movement Segment

Command	Sequence	Range
Backspace	BS	~
Carriage Return	CR	~
Form Feed	FF	~
Horizontal Tab	HT	~
Horizontal Motion Index (HMI)	Esc&k#H	0-32767
Line Feed	LF	~
Move CAP Horizontal (decipoints)	Esc&a#H	0-Page Right
Move CAP Horizontal (PCL Units)	Esc*p#X	0-Page Right
Move CAP Horizontal (Columns)	Esc&a#C	0-Page Right
Move CAP Vertical (Decipoints)	Esc&a#V	(CAP-1/3")-32767
Move CAP Vertical (PCL Units)	Esc*p#Y	(CAP-1/3")-32767
Move CAP Vertical (Rows)	Esc&a#R	(CAP-1/3")-32767
Space	SP	~
Vertical Motion Index (VMI)	Esc&l#C	0-Page Length

Rendering Segment

Command	Sequence	Range
Compression Method	Esc*b#M	0-3
End Raster	Esc*r#C	~
Raster Resolution	Esc*t#R	Device Specific
Start Raster	Esc*r#A	0-1
Transfer Raster by Row/Block	Esc*b#W	0-32767
Underline Mode	Esc&d#D	0,1,3
Underline Mode Off	Esc&d@	~

Font Segment

Command	Sequence	Range
Primary Font (Shift In)	SI	~
Primary Font Symbol Set	Esc(ID	Device Specific
Primary Font Spacing	Esc(s#P	Device Specific
Primary Font Pitch	Esc(s#H	Device Specific
Primary Font Height	Esc(s#V	Device Specific
Primary Font Style	Esc(s#S	Device Specific
Primary Font Stroke Weight	Esc(s#B	Device Specific
Primary Font Typeface	Esc(s#T	Device Specific
Secondary Font (Shift Out)	SO	~
Secondary Font Symbol Set	Esc)ID	Device Specific
Secondary Font Spacing	Esc)s#P	Device Specific
Secondary Font Pitch	Esc)s#H	Device Specific
Secondary Font Height	Esc)s#V	Device Specific
Secondary Font Style	Esc)s#S	Device Specific
Secondary Font Stroke Weight	Esc)s#B	Device Specific
Secondary Font Typeface	Esc)s#T	Device Specific

Page Presentation Segment

Command	Sequence	Range
Clear Margins	Esc9	~
Left Margin	Esc&a#L	0-Right Margin
Line Spacing	Esc&l#D	0-Page Length
Page Length	Esc&l#P	Device Specific
Perforation Skip Mode	Esc&l#L	0-1
Right Margin	Esc&a#M	Left margin-Right bound
Text Length	Esc&l#F	0-(Page Length-Top Margin)
Top Margin	Esc&l#E	0-Page Length

Device Control and Diagnostics Segment

Command	Sequence	Range
Configure AppleTalk	Esc&b#W	~
Display Functions Mode On	EscY	~
Display Functions Mode Off	EscZ	~
End-of-Line Wrap	Esc&s#C	0-1
Line Termination	Esc&k#G	0-3
Reset	EscE	~
Self-test	Escz	~
Transparent Data Mode	Esc&p#X	0-32767

Kernel Intrinsics

- Negative movement is limited to 1/3 of an inch above the lowest CAP.
- Format is line buffered rather than page buffered.
- Kerning of Fonts.
- The following Symbol Sets:

8U	-	Roman8
10U	-	PC-8 [IBM Code Page 437]
11U	-	PC-8 D/N [IBM Code Page 437 N]
12U	-	PC-850 [IBM Code Page 850]
0N	-	ECMA-94 Latin 1

Extension Blocks

Printer functionality extends beyond the kernel based upon two current markets: the *Office* market, and the *Technical Graphics* market.

Extensions are identified by *segment* and *extension block*. Extensions parallel kernel segments: i.e., movement, rendering, fonts, page presentation, and device control segments and are then further subdivided into blocks. **Each extension block must be implemented in its entirety**. Products targeted for the same market should be implemented with compatible feature sets. However, the chosen blocks may vary between markets. The process is: Define and adhere to standards based upon customer needs.

The Extension Process

Use the following two goals to define and implement extensions:

- Minimize the number of extensions.
- Do not implement subsets of Extension Blocks.

Minimize the number of extensions — Use a tools approach rather than a solution approach to define new extensions. The intent is to add as few extensions to the language as possible, supplying device driver writers with building blocks for a solution. For example, it is probably unnecessary to add a special command to print color separation marks, since commands to print small rules in specific positions are already available.

Do not implement subsets of Extension Blocks — Each extension block should be composed of escape sequences that are not only similar in functionality, but also in the cost and difficulty of implementation. Therefore, subsets of functional categories should not be implemented in a product; the entire extension block should be implemented to maintain consistency and maximize software support. Of course, to satisfy customer needs for both functionality and low cost, added functionality may vary between customer classifications.

PCL Extension Blocks

Movement Segment

Full Movement Block

Command	Sequence	Range
Move CAP Vertical (Decipoints)	Esc&a#V	(-32767)-(+32767)
Move CAP Vertical (PCL Units)	Esc*p#Y	(-32767)-(+32767)
Move CAP Vertical (Rows)	Esc&a#R	(-32767)-(+32767)

Rendering Segment

Rectangular Area Block

Command	Sequence	Range
Fill Rectangular Area	Esc*c#P	0-5
Horizontal Rectangle Size (decipoints)	Esc*c#H	0-32767
Horizontal Rectangle Size (PCL Units)	Esc*c#A	0-32767
Vertical Rectangle Size (decipoints)	Esc*c#V	0-32767
Vertical Rectangle Size (PCL Units)	Esc*c#B	0-32767
Pattern ID	Esc*c#G	0-32767

Pattern Block

Command	Sequence	Range
Current Pattern	Esc*v#T	0-4
Pattern Reference Point	Esc*p#R	0,1
Fill Rectangular Area	Esc*c#P	0-5
Horizontal Rectangle Size (Decipoints)	Esc*c#H	0-32767
Horizontal Rectangle Size (Dots)	Esc*c#A	0-32767
Vertical Rectangle Size (Decipoints)	Esc*c#V	0-32767
Vertical Rectangle Size (Dots)	Esc*c#B	0-32767
Pattern ID	Esc*c#G	0-32767

Print Model Block

Command	Sequence	Range
Current Pattern	Esc*v#T	0-4
Download Pattern	Esc*c#W	0 to (2**31) - 1
Foreground Color	Esc*v#S	0 to size of current palette - 1
Logical Operation	Esc*l#O	0-255
Source Raster Height	Esc*r#T	0-Logical Page Bound
Source Raster Width	Esc*r#S	0-Logical Page Right
Source Transparency Mode	Esc*v#N	0-1
Pattern Control	Esc*c#Q	0,1,2,4,5
Pattern ID	Esc*c#G	0-32767
Pattern Transparency Mode	Esc*v#O	0-1
Pattern Reference Point	Esc*p#R	0,1
Pixel Placement	Esc*l#R	0,1

Raster Block

Command	Sequence	Range
Raster Width	Esc*r#S	0-Logical Page Right
Y Offset	Esc*b#Y	0-Logical Page Bound

Raster Scaling Block

Command	Sequence	Range
Destination Raster Width	Esc*t#H	0-32767
Destination Raster Height	Esc*t#V	0-32767
End Raster Graphics	Esc*rB	0-32767
End Raster Graphics	Esc*rC	0-32767
Scale Algorithm	Esc*t#K	0-3
Source Raster Height	Esc*r#T	0-Logical Page Bound
Source Raster Width	Esc*r#S	0-Logical Page Right
Start Raster	Esc*r#A	2-3

Picture Frame Block

Command	Sequence	Range
Picture Frame Anchor Point	Esc*c#T	0
Picture Frame Horizontal Size (Decipoint)	Esc*c#X	0-32767
Picture Frame Vertical Size (Decipoints)	Esc*c#Y	0-32767
Plot Horizontal Size	Esc*c#K	0-32767
Plot Vertical Size	Esc*c#L	0-32767

Color Block

Command	Sequence	Range
Color Component 1	Esc*v#A	0-32767
Color Component 2	Esc*v#B	0-32767
Color Component 3	Esc*v#C	0-32767
Color Modification Lookup Tables	Esc*l#W[data]	
Color Index	Esc*v#I	0-(2**#bits/index)-1
Simple Color	Esc*r#U	-4,-3,1,3
Set Viewing Illuminant	Esc*i#W[data]	
Source Raster Width	Esc*r#S	0-Logical Page Right
Transfer Raster by Plane	Esc*b#V	0-32767

Extended Color Block

Command	Sequence	Range
Color Component 1	Esc*v#A	0-32767
Color Component 2	Esc*v#B	0-32767
Color Component 3	Esc*v#C	0-32767
Color Index	Esc*v#I	0-(2**#bits/index)-1
Configure Image Data	Esc*v#W	6, 18
Destination Raster Height	Esc*t#V	0-32767
Destination Raster Width	Esc*t#H	0-32767
Download Dither Matrix	Esc*m#W	7-32767
Finish Mode	Esc&b#F	0,1
Foreground Color	Esc*v#S	0-(size of current palette - 1)
Gamma Correction	Esc*t#I	0.0-32767.0
Monochrome Print Mode	Esc&b#M	0,1
Push/Pop Palette	Esc*p#P	0,1
Render Algorithm	Esc*t#J	0-10
Scale Algorithm	Esc*t#K	0-3
Simple Color	Esc*r#U	-3,1,3
Source Raster Width	Esc*r#S	0-Logical Page Right
Source Raster Height	Esc*r#T	0-Logical Page Bound
Start Raster Graphics	Esc*r#A	2-3
Transfer Raster by Plane	Esc*b#V	0-32767
Y Offset	Esc*b#Y	0-Logical Page Bound

Fonts Segment

Download Fonts Block

Command	Sequence	Range
Character Code	Esc*c#E	0-255
Download Character	Esc(s#W	0-32767
Download Font	Esc)s#W	0-32767
Font Control	Esc*c#F	0-6
Font ID	Esc*c#D	0-32767
Font Selection by ID (Primary)	Esc(#X	0-32767
Font Selection by ID (Secondary)	Esc)#X	0-32767

Download Symbol Sets Block

Command	Sequence	Range
Symbol Set Code	Esc*c#R	0-65535
Symbol Set Control	Esc*c#S	0,1,2,4,5
Download Symbol Set	Esc(f#W	0-32767

Page Presentation Segment

Macros Block

Command	Sequence	Range
Macro Control	Esc&f#X	0-10
Macro ID	Esc&f#Y	0-20
Push/Pop CAP	Esc&f#S	0-1

Orientation Block

Command	Sequence	Range
Orientation	Esc&l#O	0-3

Print Direction Block

Command	Sequence	Range
Print Direction	Esc&a#P	0, 90, 180, 270

Device Control and Diagnostics Segment

Duplexing Block

Command	Sequence	Range
Left Registration	Esc&l#U	(-32767)-(+32767)
Orientation	Esc&l#O	0-3
Top Registration	Esc&l#Z	(-32767)-(+32767)

Language Switching Extension GROUP

Command	Sequence	Range
Enter PCL mode	Esc%#A	0
Enter HP-GL/2 mode	Esc%#B	-1 to 3
Universal Exit Language/Start PJL	Esc%#X	-12345

Print Mechanism Adjustment Extension GROUP

Command	Sequence	Range
Copies	Esc&l#X	1-32767
Media Source	Esc&l#H	(-1)-6
Page Orientation	Esc&l#O	0-3
Page Size	Esc&l#A	Device specific
Presentation Mode	Esc*r#F	0,3
Underline Mode	Esc&d#D	2,4

HP-GL/2 Commands

The following categories specify the HP-GL language commands recognized and acted upon by the HP Color LaserJet controller. These commands will perform their documented functionality with any exceptions as outlined in their respective chapters.

Dual Context Extensions

• EscE	Reset
• Esc%#A	Enter PCL5 Mode
• Esc%#B	Enter HP-GL/2 Mode
• Esc%#X	Exit Language / Start PCL5
• FI	Primary Font Selection by ID
• FN	Secondary Font Selection by ID
• SB	Scalable or Bitmapped Raster Fonts

Palette Extensions

• TR	Transparency Mode
• NP	Number of Pens
• SV	Screened Vectors

HP-GL/2 Kernel

Line and Fill Attributes Group

• LT	Line Type
• LA	Line Attributes
• PW	Pen Width
• WU	Pen Width Unit Selection
• SP	Select Pen
• SM	Symbol Mode
• FT	Fill Type
• AC	Anchor Corner
• RF	Raster Fill Definition
• UL	User Defined Line Type

Vector Group

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

Polygon Group

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

Character Group

• SS	Select Standard Font
• SA	Select Alternate Font
• DI	Absolute Direction
• DR	Relative Direction
• SI	Absolute Character Size
• SR	Relative Character Size
• SL	Character Slant
• ES	Extra Space
• SD	Standard Font Definition
• AD	Alternate Font Definition
• CF	Character Fill Mode
• LO	Label Origin
• LB	Label
• DT	Define Label Terminator
• CP	Character Plot
• TD	Transparent Data
• DV	Define Variable Text Path

Configuration and Status Group

SC	Scale
IW	Input Window
IP	Input P1 and P2
IR	Input Relative P1 and P2
DF	Default Values
IN	Initialize
RO	Rotate Coordinate System
PG	Advance Full Page
RP	Replot

Color and Palette Group

CR	Set Relative Color Range
PC	Pen Color Assignment

NEW GROUP (to be added soon :-)

MC	Merge Control
PP	Pixel Placement

Unsupported PCL Commands

The following group consists of PCL commands recognized by the firmware PCL language parser, but are not acted upon. This implies that these commands may be present in the data stream, but will not cause any action to be taken place within the firmware parser and language environment. No mention of these commands will be made in the HP Color LaserJet Technical Reference Documentation. The following commands provide extensions for special engine features particular to EP (electrophotographic) print engines that support the realization of duplex printing as well as Thermal Ink Jet device particulars. None of these options are available in the HP Color LaserJet printer, and therefore they will not be supported.

• Esc&l#S	Simplex / Duplex Printing
• Esc&l#T	Job Separation (PENDING)
• Esc&a#G	Duplex Page Side
• Esc%-1B	Enter HP-GL/2 Plotter Mode
• Esc&a#N	Negative Motion
• Esc&v#S	Text Color
• Esc&o#Q	Mechanical Print Quality
• Esc(s#Q	Primary Font Quality
• Esc)s#Q	Secondary Font Quality
• Esc&k#W	Print Mode Selection
• Esc*b#X	Raster X Offset

The following group of PCL commands consist of either obsolete or not officially support PCL extensions. The language parser will recognize these commands, but WILL NOT act upon them in the same manner as is done by the IIISi controller. However, no mention of these commands will be made in the HP Color LaserJet Technical Reference Documentation.

• Esc&l#V	Print With Paper Motion Via VFC
• Esc&l#W[data]	Download 16 Channel VFC
• Esc&k#S	Set Primary/Secondary Font Pitch Mode
• Esc&a#W[data]	User Defined Logical Page

Unsupported HP-GL Commands

The following group consists of HP-GL commands recognized by the firmware HP-GL language parser, but are not acted upon. This implies that these commands may be present in the data stream, but will not cause any action to be taken place within the firmware parser and language environment. No mention of these commands will be made in the HP Color LaserJet Technical Reference Documentation.

• BP	Begin Plot
• CT	Chord Tolerance Mode
• DL	Download Character
• EC	Enable Cutter
• FR	Frame Advance
• MG	Message
• MT	Media Type
• NR	Not Ready
• OE	Output Error
• OH	Output Hardcopy Limits
• OI	Output Identification
• OP	Output P1 and P2
• OS	Output Status
• PS	Plot Size
• QL	Quality Level
• ST	Sort
• VS	Velocity Select

PCL commands List Alphabeticaly

Command	Description
BS	Backspace
CR	Carriage Return
Esc%#A	Enter PCL mode
Esc%#B	Enter HP-GL/2 mode
Esc%#X	Universal Exit Language/Start PJL
Esc&a#C	Move CAP Horizontal (Columns)
Esc&a#H	Move CAP Horizontal (decipoints)
Esc&a#L	Left Margin
Esc&a#M	Right Margin
Esc&a#P	Print Direction
Esc&a#R	Move CAP Vertical (Rows)
Esc&a#V	Move CAP Vertical (Decipoints)
Esc&b#F	Finish Mode
Esc&b#M	Monochrome Print Mode
Esc&b#W	Configure AppleTalk
Esc&d#D	Underline Mode
Esc&d@	Underline Mode Off
Esc&f#S	Push/Pop CAP

Esc&f#X	Macro Control
Esc&f#Y	Macro ID
Esc&k#G	Line Termination
Esc&k#H	Horizontal Motion Index (HMI)
Esc&l#A	Page Size
Esc&l#C	Vertical Motion Index (VMI)
Esc&l#D	Line Spacing
Esc&l#E	Top Margin
Esc&l#F	Text Length
Esc&l#H	Media Source
Esc&l#L	Perforation Skip Mode
Esc&l#O	Page Orientation
Esc&l#P	Page Length
Esc&l#U	Left Registration
Esc&l#X	Copies
Esc&l#Z	Top Registration
Esc&p#X	Transparent Data Mode
Esc&s#C	End-of-Line Wrap
Esc(#X	Font Selection by ID (Primary)
Esc(f#W	Download Symbol Set
Esc(ID	Primary Font Symbol Set
Esc(s#B	Primary Font Stroke Weight
Esc(s#H	Primary Font Pitch
Esc(s#P	Primary Font Spacing
Esc(s#S	Primary Font Style
Esc(s#T	Primary Font Typeface
Esc(s#V	Primary Font Height
Esc(s#W	Download Character
Esc)#X	Font Selection by ID (Secondary)
Esc)ID	Secondary Font Symbol Set
Esc)s#B	Secondary Font Stroke Weight
Esc)s#H	Secondary Font Pitch
Esc)s#P	Secondary Font Spacing
Esc)s#S	Secondary Font Style
Esc)s#T	Secondary Font Typeface
Esc)s#V	Secondary Font Height
Esc)s#W	Download Font
Esc*b#M	Compression Method
Esc*b#V	Transfer Raster by Plane
Esc*b#W	Transfer Raster by Row/Block
Esc*b#Y	Y Offset
Esc*c#A	Horizontal Rectangle Size (PCL Units)
Esc*c#B	Vertical Rectangle Size (PCL Units)
Esc*c#D	Font ID
Esc*c#E	Character Code
Esc*c#F	Font Control
Esc*c#G	Pattern ID
Esc*c#H	Horizontal Rectangle Size (Decipoints)
Esc*c#K	Plot Horizontal Size
Esc*c#L	Plot Vertical Size

Esc*c#P	Fill Rectangular Area
Esc*c#Q	Pattern Control
Esc*c#R	Symbol Set Code
Esc*c#S	Symbol Set Control
Esc*c#T	Picture Frame Anchor Point
Esc*c#V	Vertical Rectangle Size (decipoints)
Esc*c#W	Download Pattern
Esc*c#X	Picture Frame Horizontal Size (Decipoint)
Esc*c#Y	Picture Frame Vertical Size (Decipoints)
Esc*i#W[data]	Set Viewing Illuminant
Esc*l#O	Logical Operation
Esc*l#R	Pixel Placement
Esc*l#W[data]	Color Modification Lookup Tables
Esc*m#W	Download Dither Matrix
Esc*p#P	Push/Pop Palette
Esc*p#R	Pattern Reference Point
Esc*p#X	Move CAP Horizontal (PCL Units)
Esc*p#Y	Move CAP Vertical (PCL Units)
Esc*r#A	Start Raster Graphics
Esc*r#C	End Raster
Esc*r#F	Presentation Mode
Esc*r#S	Source Raster Width
Esc*r#S	Raster Width
Esc*r#S	Source Raster Width
Esc*r#T	Source Raster Height
Esc*r#U	Simple Color
Esc*rB	End Raster Graphics
Esc*rC	End Raster Graphics
Esc*t#H	Destination Raster Width
Esc*t#I	Gamma Correction
Esc*t#J	Render Algorithm
Esc*t#K	Scale Algorithm
Esc*t#R	Raster Resolution
Esc*t#V	Destination Raster Height
Esc*v#A	Color Component 1
Esc*v#B	Color Component 2
Esc*v#C	Color Component 3
Esc*v#I	Color Index
Esc*v#N	Source Transparency Mode
Esc*v#O	Pattern Transparency Mode
Esc*v#S	Foreground Color
Esc*v#T	Current Pattern
Esc*v#W	Configure Image Data
Esc9	Clear Margins
EscE	Reset
EscY	Display Functions Mode On
EscZ	Display Functions Mode Off
Escz	Self-test
FF	Form Feed
HT	Horizontal Tab

LF	Line Feed
SI	Primary Font (Shift In)
SO	Secondary Font (Shift Out)
SP	Space

HPGL Commands listed Alphabetically

Command	Description
AA	Arc Absolute
AC	Anchor Corner
AD	Alternate Font Definition
AR	Arc Relative
AT	Absolute Arc Three Point
BR	Bezier Relative
BZ	Bezier Absolute
CF	Character Fill Mode
CI	Circle
CP	Character Plot
CR	Set Relative Color Range
DF	Default Values
DI	Absolute Direction
DR	Relative Direction
DT	Define Label Terminator
DV	Define Variable Text Path•
EA	Edge Rectangle Absolute
EP	Edge Polygon
ER	Edge Rectangle Relative
ES	Extra Space
EW	Edge Wedge
FI	Primary Font Selection by ID
FN	Secondary Font Selection by ID
FP	Fill Polygon
FT	Fill Type
IN	Initialize
IP	Input P1 and P2
IR	Input Relative P1 and P2
IW	Input Window
LA	Line Attributes
LB	Label
LO	Label Origin
LT	Line Type
MC	Merge Control
NP	Number of Pens
PA	Plot Absolute
PC	Pen Color Assignment
PD	Pen Down
PE	Polyline Encoded
PG	Advance Full Page

PM	Polygon Mode
PP	Pixel Placement
PR	Plot Relative
PU	Pen Up
PW	Pen Width
RA	Fill Rectangle Absolute
RF	Raster Fill Definition
RO	Rotate Coordinate System
RP	Replot
RR	Fill Rectangle Relative
RT	Relative Arc Three Point
SA	Select Alternate Font
SB	Scalable or Bitmapped Raster Fonts
SC	Scale
SD	Standard Font Definition
SI	Absolute Character Size
SL	Character Slant
SM	Symbol Mode
SP	Select Pen
SR	Relative Character Size
SS	Select Standard Font
SV	Screened Vectors
TD	Transparent Data
TR	Transparency Mode
UL	User Defined Line Type
WG	Fill Wedge
WU	Pen Width Unit Selection

Chapter 2: HP Color LaserJet Features

HP Color LaserJet Features

There are several areas to examine when looking at new features of a printer.

- New PCL features.
- Features **Not** in HP Color LaserJet
- New hardware/IO differences.
- New engine difference.

PCL features.

Color features.

The HP Color LaserJet has extended the IIISi firmware base by adding color to every PCL and HP-GL/2 entity. In addition, The HP Color LaserJet has implemented the use of color palettes as well as palette control or management with commands such as push/pop palette and palette id within the PCL5 language.

Device Independent Color.

In order to provide *"what you want is what you get* ***(WYWIWYG)****"* color from the original specification of a page composition to the printed page, The HP Color LaserJet will support appearance matching via device independent color extensions. The following provisions are provided to enable this functionality:

1) Color lookup Tables
2) Viewing Illuminant Support
3) Multiple Color Space Support
4) Multiple Half toning Techniques
5) HP Color LaserJet Device Characterization
6) Driver Support to Achieve Screen to Printer Match

Each of these major features will be further illustrated and discussed in subsequent chapters concerning Color Extensions.

Raster Operations.

Traditional LaserJet printers have implemented the PCL print model via Logical Operations. Raster operations (ROP3) are extensions to the PCL Print Model, which parallel the ROP3 functionality of Windows GDI. These are RGB ROP3 operations, and are converted internally into CMY ROP3 equivalents.

Improved Rendering Capability.

The HP Color LaserJet will support various modes of halftone generation for increasing the printer's color gamut. The following forms of rendering are supported for all language page marking commands:

1) Cluster Dither for the more traditional halftone screen.
2) Bayer Dither for better resolution.
3) Error Diffusion for maintaining detail quality.
4) Probability based error diffusion.
5) Monochrome Page Printing For Color Compositions.
6) Improved scaling technique - provides smoother image quality

Also included in the new HP Color LaserJet firmware base is provision for printing a color page in its monochrome equivalent. This feature maintains page composition distinction with gray levels instead of color. This provides improved throughput for color page compositions that a user wants to print in a black and white mode without altering the original file. As an example, in Windows, the driver simply inserts the monochrome print mode command at the beginning of the original data stream when the user chooses to do so via the driver dialog box.

User Defined Color Patterns.

The HP Color LaserJet supports the creation of user defined color patterns used for solid area fills.

User Defined Halftones.

The HP Color LaserJet has added new extensions for supporting downloadable user defined halftones. This functionality supports the concept of alterable screening halftones.

Color Palette Control.

The HP Color LaserJet supports palette management (pushing and popping of the current color palette or selection of palette id) for saving color palette context during language switching between PCL and HP-GL.

Additional Internal Fonts.

The HP Color LaserJet controller will support the "HP 45" fonts (35 Intellifont and 10 True Type) in addition to the 8 bitmap fonts for a total of 53 internal fonts, compatible with Sawtooth and Payette. This includes the Latin 2 and Latin 5 line printer bitmap fonts.

Character Set.

The HP Color LaserJet will support unbound PCLEO's and user defined symbol sets.

New HP-GL/2.

The HP Color LaserJet will support the ***"Next Generation"*** HP-GL/2 firmware base. This new code base includes:

1) Bezier Curve support
2) Both Even-Odd and Zero-Winding Fill options.
3) PCL Coordinate System tracking capability.
4) PCL font position placement capability.
5) Raster Operation support.

Resident languages.

PCL5 with Color Extensions is the native language for the HP Color LaserJet, but on board Postscript will be offered as an option via SIMM support.

The HP Color LaserJet will support a subset of the PJL commands. PJL is the job control language supporting features relating to print jobs. The features allowable with the supported subset are detailed in the Chapter "PJL Commands". PJL support and consist of Context sensitive language switching, true end-of-job notification, job boundary recognition (MIO, PJL, timer compatible with 4Si and Payette), I/O switching, automatic installation, automatic configuration. The subset of PJL commands to be supported by The HP Color LaserJet are: COMMENT, ECHO, ENTER, EOJ, INFO CONFIG, INFO ID, INFO USTATUS, JOB, UEL, USTATUS JOB, USTATUSOFF and can be found in the Printer Job Language Technical Reference Manual.

The HP Color LaserJet will **NOT** be supporting environment saving between language switching. PJL features are described in detailed in Chapter "PJL Commands".

Self test page.

The HP Color LaserJet's self test page has been modified to include the MIO 5.2 information as well as additional changes for color functionality.

Extended diagnostics.

When in the extended diagnostics mode, The HP Color LaserJet provides improved diagnostic information for the electro-photographic color engine.

New hardware features.

Base RAM support.

The HP Color LaserJet supports 8 Mbytes for resident base memory. Under full color processing, The HP Color LaserJet guarantees no page punt printing.

Expansion RAM.

A HP Color LaserJet supports 4 SIMM sockets. These sockets may be used for language personality SIMMs or expansion RAM slots. As an example, if the SIMM sockets are used for expansion RAM, and each socket can contains 16 megabytes, giving a total of 64 megabytes via SIMM support, add this to the 8 Mbyte on board RAM for a total of 72 Mbyte system printer memory. Certain memory configuration are not allowed. Refer to *HP Color LaserJet User Manual* for more information.

Front Panel enhancements.

The HP Color LaserJet front panel has been modified to support 4Si's new menu system, and to increase simplicity.

New modular I/O communications.

The HP Color LaserJet supports recent MIO 5.1 interface specifications; this MIO interface is detailed in the MIO Specification document available through the MIO developers program.

Error reporting.

The HP Color LaserJet engine provides more detailed error and warning capability when consumables are near their useful life. The engine can detect seven distinct paper jam locations and several laser beam errors. Error reporting also includes low toner conditions and photo drum and developers reaching the end of their useful life.

Resolution Enhancement Technology.

The HP Color LaserJet supports Resolution Enhancement Technology (RET) hardware for black and white only. RET will not be applied to color entities.

Edge Recognition Technology

In order to provide better color rendering on the new engine, specialized hardware will be provided to detect the presence of certain types of color edges as well as isolated dots that need unique image rendering. This functionality will be completely transparent to the user of a HP Color LaserJet printer. Much like the RET technology, output processed by Edge Recognition Technology (ERT) will constitute much better color and print quality.

New engine features

New engine.

The HP Color LaserJet engine is based on a new electro-photographic engine technology. Page throughput is rated at 10 pages per minute for monochrome page printing, and 2.0 pages per minute for matte color page processing. Transparency output has a rated throughput of 1 page per minute.

Paper Handling.

The HP Color LaserJet controller will be the second laser printer with B-Size paper handling support (the LJ2000 was first). The HP Color LaserJet supports Letter, Executive and A4 in color and Letter, A4, Executive, Legal, Tabloid in black and White.

Dual source trays.

The HP Color LaserJet provides one 250 sheet (front) media source tray or cassette and an optional rear (upper) 250 sheet media tray. The HP Color LaserJet also provides manual feed entry for non-standard weight media or transparencies.

Dual output bins.

There are two output bins which can be selected via firmware extensions. The first mode places paper in the upper output tray in a face down, correct order fashion. The second selection provides for face up, non-correct order (reverse) output on the middle tray, this is primarily for transparencies to prevent curling. This middle tray is accessed by pulling out the smoky gray cover on the main media source tray.

Industrial design.

The HP Color LaserJet has a new industrial design.

Chapter 3:   Paper Handling

Paper Handling

While job control commands are usually sent at the beginning of the print job, page control commands are associated with a single page or with groups of pages. If a number of consecutive pages within a job have the same format, the associated page control commands need only be sent once for that group of pages. Page control commands determine such features as media source, size, orientation, margins, and text spacing.

Page Length and Page Size

The default length of a PCL logical page extends from the top edge of the physical page to the bottom edge. The width is device dependent and usually does not extend as far as the unprintable area on the sides.

The logical page defines the current addressable area for PCL mode. In the HP Color LaserJet, the default logical page is bounded on the left and right sides of the physical page by a 1/4 inch region for US paper sizes, and 6mm for A4 paper. The upper bound of the default logical page is specified by the page length command and is generally the bottom of the physical page. The user cannot move outside the logical page bounds while in PCL mode.

The Page Length command (*Esc&l#P*) sets logical page length for a given VMI. The Page Size command (*Esc&l#A*), which selects physical Page Size from one of several pre-set sizes, indirectly sets logical page length (which always defaults to the physical page length). Since both commands eject the current page if any printable data has been sent, they should be sent at the beginning of the page prior to any print data.

For comparison, some default logical page sizes for letter paper are given below in inches.

Registration is the distance from the physical page to the logical page.

Default Logical Pages for the LJ2000, LJIII, LJ4, PJXL300

Parameter	**Portrait**	**Landscape**
Width	8.0	10.6
Height	11.0	8.5
Left Registration	0.25*	0.2
Top Registration	0.0	0.0

* Left Registration for LJ, LJ+, LJ500, LJII is 0.1666.

Default Logical Pages for the HP Color LaserJet

Parameter	**Portrait**	**Landscape**
Width	8.1666	10.666
Height	11	8.5
Left Registration	0.1666*	0.1666*
Top Registration	0.0*	0.0*

* For US. Letter Paper. For European (A3 and A4) size paper, these values are all 6mm.

Page Length *Esc & l # p/P*

Designates the length of the logical page for a given VMI.

Value(#)	=	Number of lines at a given VMI
Default	=	Device dependent, control panel value
Range	=	0 to the maximum supported paper size (other values ignored and current size retained)*****

* A value of 0 should set page length to the default media size configured at the control panel in the current orientation. If the received value is greater than the maximum supported media size, or if the VMI is zero, the Set Page Length command is ignored, however, the other actions below still occur.

This command performs the following actions:

- Prints any unprinted pages.
- Ejects the current page if printable data has been received (FF-CR).
- Sets a new Page Size if the requested media is supported, otherwise the current Page Size is retained. If the Page Size has changed, the HP Color LaserJet requests that the new size be loaded.
- Sets text length, top margin, left margin, and right margin to user defaults.
- Moves CAP to the left edge of the logical page at the top margin.
- Disables the automatic macro overlay.
- Defaults:
  - the picture frame anchor point
  - the picture frame
  - the PCL print direction
  - the HP-GL/2 plot size
  - P1 and P2
  - the soft clip window
- Updates the HP-GL/2 CAP to the lower-left corner of the picture frame
- Clears the polygon buffer

- Perform an IN command

The printer may select a different page size for the same line count, since lines are defined by the current VMI. This command is ignored if VMI is 0.

For comparison purposes, the following table for the HP Color LaserJet lists the page length line values associated with some standard paper sizes. To calculate the number of lines per page, multiply lines per inch (lpi) times the length of the physical page. For example, US Letter size paper is 11 inches; therefore: 6 x 11 = 66.

HP Color LaserJet Portrait Orientation Page Length Settings

Paper Size	6 lpi	8 lpi
Letter	66	88
Legal	84	112
A4	70	93
Executive	63	84
Tabloid	102	136
A3	99	132

HP Color LaserJet Landscape Orientation Page Length Settings

Paper Size	6 lpi	8 lpi
Letter	51	68
Legal	**	**
A4	49	66
Executive	43	58
Tabloid	66	88
A3	70	93

NOTE The Page Length command must be used with the LJ, LJ+, and LJ500 to set page size. LJII, LJIID, LJIIP, LJIII, or LJ2000 can use either the Page Length or the Page Size command. Page Size is preferred.

Default page length may be switch selectable. Common lengths are 8½, 11 and 12 inches.

PJ page lengths are set by switch to English or metric. The default at 6 lpi is 66 lines English and 72 lines metric, with a perforation skip of 6 lines.

This command must be sent at the beginning of a page prior to any printable data; otherwise when the command is sent, the current page is closed and printed.

** Page lengths for legal land letter sizes are identical in landscape orientation. To set legal media in landscape orientation, the user should switch to portrait orientation, set the size for legal, and then return to landscape orientation.

Page Size *Esc & l # a/A*

Designates the size of the media as one of those listed below.

Value(#) = Page Size (listed below)
Default = Device dependent
Range = 1, 2, 3, 6, 26, and 27 are supported by the HP Color LaserJet. For a comprehensive list of selectable media sizes see the table below. (unsupported values are ignored)

This command performs the following actions:

- Prints any unprinted pages.
- Ejects the current page if printable data has been received (FF-CR).
- Sets text length, top margin, left margin, and right margin to user defaults.
- Moves CAP to the left edge of the logical page at the top margin.
- Disables the automatic macro overlay.

Since the logical page is defined to extend from the top and bottom edges of the physical page, this command also sets logical page length.

The following table lists the current paper sizes selectable by this command.

Media sizes Selectable by the Page Size Command
(The sizes the HP Color LaserJet supports are highlighted in light cyan)

Value	Page Description	Page Size
1	US-Executive	(7.25" x 10.5")
2	US-Letter	(8.5" x 11")
3	US-Legal	(8.5" x 14")
4	US-EDP	(11" x 14")
5	European-EDP	(12" x 14")
6	Tabloid	(11" x 17")
7	US Government Letter	(8" x 10")
8	US Government Legal	(8" x 13")
9	Folio	(8.3" x 13")
10	Foolscap	(8.5" x 13")
11	Tabloid	(11" x 17")
12	C Size	(17" x 22")
13	D Size	(22" x 34")
14	E Size	(34" x 44")
15	Mini	(5.5" x 8.5")
20	ISO and JIS A10	(26mm x 37mm)
21	ISO and JIS A9	(37mm x 52mm)
22	ISO and JIS A8	(52mm x 74mm)
23	ISO and JIS A7	(74mm x 105mm)
24	ISO and JIS A6	(105mm x 148mm)
25	ISO and JIS A5	(148mm x 210mm)
26	ISO and JIS A4	(210mm x 297mm)
27	ISO and JIS A3	(297mm x 420mm)
28	ISO and JIS A2	(420mm x 594mm)
29	ISO and JIS A1	(594mm x 841mm)
30	ISO and JIS A0	(841mm x 1189mm)
31	ISO and JIS 2A0	(1189mm x 1682mm)
32	ISO and JIS 4A0	(1682mm x 2378mm)

Value	Page Description	Page Size
40	JIS B10	(32mm x 45mm)
41	JIS B9	(45mm x 64mm)
42	JIS B8	(64mm x 91mm)
43	JIS B7	(91mm x 128mm)
44	JIS B6	(128mm x 182mm)
45	JIS B5	(182mm x 257mm)
46	JIS B4	(257mm x 364mm)
47	JIS B3	(364 x 515mm)
48	JIS B2	(515mm x 728mm)
49	JIS B1	(728mm x 1030mm)
50	JIS B0	(1030mm x 1456mm)
60	ISO B10	(31mm x 44mm)
61	ISO B9	(44mm x 62mm)
62	ISO B8	(62mm x 88mm)
63	ISO B7	(88mm x 125mm)
64	ISO B6	(125mm x 176mm)
65	ISO B5	(176mm x 250mm)
66	ISO B4	(250mm x 353mm)
67	ISO B3	(353mm x 500mm)
68	ISO B2	(500mm x 707mm)
69	ISO B1	(707mm x 1000mm)
70	ISO B0	(1000mm x 1414mm)

Value	Envelope Description	Envelope Size
-90	International DL (landscape)	(110mm x 220mm)
-81	Number 10 (landscape)	(4 1/8" x 9 1/2")
80	Monarch (Letter)	(3 7/8" x 7 1/2")
81	Commercial-10 (Business)	(4 1/8" x 9 1/2")
82	Catalog 1	(6" x 9")
90	International DL	(110mm x 220mm)
91	International C5	(162mm x 229mm)
92	International C6	(114mm x 162mm)
93	International C4	(229mm x 324mm)
100	International B5	(176mm x 256mm)

NOTE: LJ, LJ+, LJ500 ignore this command and use *Esc&l#P* (Page Length) instead. LaserJets set logical page width to (physical width - 1/2") for US portrait sizes, to (physical width - 12mm) for European portrait sizes, to (physical length - 2/5") for US landscape sizes, and to (physical length - 10mm) for European landscape sizes.

In DJ550, envelopes load in landscape and do not distinguish by sign. Pre-DJ550s load envelopes in portrait.

On DJ, 0 returns paper size to the configuration switch selection.

The HP Color LaserJet does not support envelopes. The HP Color LaserJet does not support full-color printing on the following three media sizes - tabloid, legal and A3, which only print in monochrome.

Media Source

Media Source *Esc & l # h/H*

Selects the media source.

Value(#)	=	-3	-	Reserved
	=	-2	-	Reserved
	=	-1	-	Reserved
	=	0	-	Print current page (source is unchanged)
	=	1	-	Feed from Front/Main tray #1*****
	=	2	-	Feed from Manual top-rear input tray #2
	=	3	-	Reserved
	=	4	-	Feed from optional cassette or Alternate Feed Unit (AFU), 250 sheet cassette at the top-rear of the printer. tray #3
	=	5	-	Large paper source
	=	6	-	Reserved
Default	=			Device dependent
Range	=	-3 to 6 (defaults if non-existent tray selected)		

***** In a 2-tray system, tray #1 is usually the upper tray and tray #2 is the lower tray, however in the HP Color LaserJet, a three tray printer, Tray #2 is the Manual feed and tray #3 the optional AFU (Alternate Feed Unit). When purchased and installed the AFU incorporates the manual feed unit, providing an integrated 250 sheet cassette with a manual feed slot and is located at the top back of the printer. Tray #1 is still the lower front cassette or tray.

This command prints the current page and moves CAP to top of form at the left margin on the next physical page. The command remains effective until another tray is selected.

NOTE: Rugged Writer implements the -1 value.

The HP Color LaserJet's auto cassette selection (TRAY=BOTH as set from the front panel) will override the media source or cassette selection if the requested cassette does not have the correct size media or if during a print job the currently accessed cassette runs out of paper and another unsecured cassette has the correct size media (i.e. the media size has priority over cassette selection). If (TRAY ≠ BOTH) and the requested tray does not have the requested media size loaded the printer will display "11.x Load requested media size", and printing will be held up until someone presses continue (On Line) and the proper size paper can be found in another cassette, the front panel is reconfigured for TRAY=BOTH, or the proper size media is loaded into the requested tray. For further discussion concerning front panel override priority or function refer to the *HP Color LaserJet Front Panel ERS***.**

Orientation

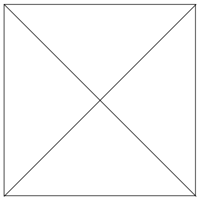
Orientation defines the position of the logical page on the physical page. *Portrait* orientation means (0,0) is toward the top left corner of the physical page: positive X direction is to the right and positive Y direction downward. In *reverse portrait* orientation, (0,0) is toward the lower right corner of the physical page: positive X direction is to the left and positive Y direction upward.

Landscape orientation means (0,0) is toward the lower left corner of the physical page with the X direction up and the Y direction to the right. In *reverse landscape* orientation, (0,0) is toward the upper right corner of the physical page with the X direction downward and the Y direction toward the left.

Logical page orientation changes text orientation, **not** raster graphics orientation; for example, raster graphics will continue to print in portrait if orientation is changed from portrait to landscape. Fonts are automatically rotated to the current orientation on devices having auto-rotation.

Although orientation may be changed by either the Orientation command or the Print Direction command, these two commands have different side-effects:

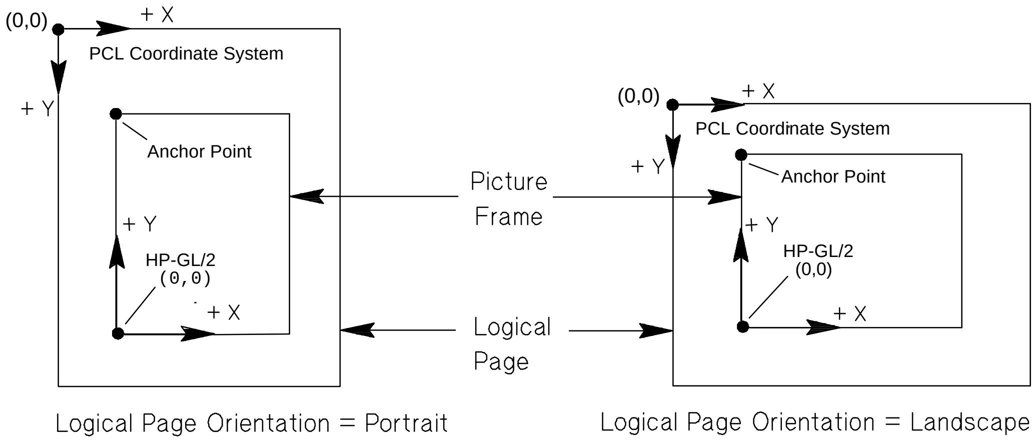
- Orientation ejects the page and Print Direction does not; therefore, Print Direction can be used to make orientation changes on the same page.
- Print Direction does not affect HP-GL/2 graphics unless HP-GL/2 is entered with *Esc%2B* or *Esc%3B.* HP-GL/2 graphics can always be rotated by the Orientation or "RO" commands.



HP-GL/2 Orientation

Default HP-GL/2 orientation tracks the Orientation command (*Esc&l#O*). HP-GL/2 orientation also tracks the Print Direction command (*Esc&a#P*) if and only if HP-GL/2 has been entered by *Esc%2B* or *Esc%2B*. Otherwise HP-GL/2 command RO must be used to rotate the coordinate system or orientation.

In both coordinate systems, X-axes are parallel and increasing in the same direction, and Y-axes are parallel and increasing in opposite directions. This remains true even if logical page orientation changes.



How HP-GL/2 Orientation Tracks Logical Page Orientation*

* HP-GL/2 orientation always tracks the Orientation command (*Esc&l#O*). However HP-GL/2 orientation tracks the Print Direction command (*Esc&a#P*) only if HP-GL/2 has been entered by *Esc%2B* or *Esc%3B*.

Orientation *Esc & l # o/O*

Defines the position of the logical page and the default direction of print with respect to the physical page.

Value(#) = 0 - Portrait, physical page has one of the short sides up and printing is across the width of the page.
= 1 - Landscape, the page is rotated counterclockwise by 90 degrees and printing is across the "length' of the physical page.
= 2 - Reverse portrait, page is rotated counterclockwise 180 degrees from portrait orientation.
= 3 - Reverse landscape, page is rotated counterclockwise 180 degrees from landscape orientation.
Default = 0
Range = 0 to 3 (undefined values are ignored)

If the printer has received printable data, an orientation change ejects the current page and opens another page in the new orientation. The command is ignored if the new orientation and the current orientation are the same.

IMPLEMENTOR'S NOTE Even if this command is not supported, it should still close and print the current page.

This command has the following effects:

- Prints all data received before the command
- Executes a formfeed and carriage return.
- Sets the following to their user defaults:
  - logical page
  - print direction
  - page length
  - text length
  - top margin, left margin, right margin
  - HMI and VMI
- Disables the auto macro overlay

NOTE Since this command ejects the page, it cannot be used to change text orientation within a page. The Print Direction command (*Esc&a#P*) may be used to print multiple directions per page.

This command should be sent at the beginning of a page because it ejects a page containing printable data. Since this command defaults the above features, it should be followed by commands that set any desired non-default values.

This command affects only text orientation — **not** raster graphics orientation. For example, if orientation is changed from portrait to landscape, graphics will continue to print in portrait. This can cause clipping if graphics margins are insufficient. See Chapter 13 for information on raster graphics margins.

Fonts automatically rotate to the current orientation on devices having auto-rotation.

NOTE: Pre-LJIII printers did not rotate fonts to the current orientation. Orientation was a selection criterion. Pre-DJ550s lock out graphics data in landscape orientation.

Print Direction *Esc & a # p/P*

Rotates the logical page coordinate system with respect to its current orientation, without ejecting the page.

Value(#) = Degrees of rotation (ccw 90 degree increments only)
Default = 0
Range = 0, 90, 180, 270 (other values are ignored)

By not ejecting the page, this command allows orientation changes on the same page.

This command has the following effects:

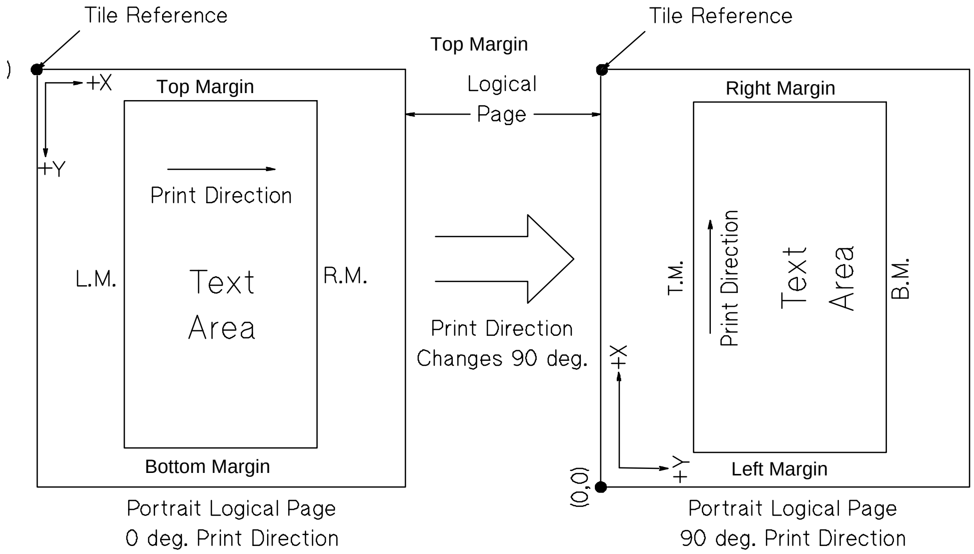
1. The logical page coordinate system origin (0,0) rotates with the logical page. For example, rotating a default page (portrait orientation, 0° print direction) by 90° causes data to print in the landscape direction across the "portrait" page. That is, orientation is changed from portrait to landscape. Page width, length, top offset, and left offset are set appropriately.

NOTE: When the print direction is rotated by 90 or 270, it must be noted that upon entering HP-GL mode the origin (0,0) falls outside the picture frame, ie outside of the printable area.

2. If CAP is fixed (i.e., following printable data or commands affecting CAP), it remains at the same location on the physical page. If CAP is floating (i.e., before printable data or commands affecting CAP), it remains floating.
3. The margins are translated (as shown on the next page). For example, a print direction change from 0 to 90 degrees makes the left margin the new top margin, the top margin the new right margin, etc.
4. The positions stored in the CAP stack are translated to reference the same location on the physical page. For example, if print direction changes from 0 to 90 degrees, stored X coordinates become Y coordinates, and vice versa. This allows the Push/Pop CAP command to be used for storing exact physical page positions, regardless of print direction.
5. The picture frame anchor point coordinates are transformed so the anchor point and picture frame remain at the same location on the physical page.
6. Any current raster graphics terminates when the print direction changes.
7. All subsequent print entities — raster, area fills, and characters — are rotated.
8. The HP-GL/2 coordinate system is also rotated if the Enter HP-GL/2 command (*Esc%#B*) has been sent with a value field of 2 or 3. Unless *Esc%#B* is sent with a value of 2 or 3, print direction has no effect on HP-GL/2 graphics, which can then only be rotated by RO or the Orientation command (*Esc&l#O*).
9. The tiling reference point is not affected: it is fixed at the upper-left corner of the logical page in the specified viewing orientation. However, the tiles are rotated to coincide with the current print direction.

10. The auto macro overlay environment is not disabled, since an overlay may invoke this command.

11. Print Direction does not default HMI.



Changing print direction

HMI and VMI

The Horizontal Motion Index (HMI) defines the width of columns used for setting margins, horizontal CAP positioning, and the spacing between all characters in a fixed-pitch font. For proportional fonts, HMI affects only the SPACE character (unless SPACE is downloaded as a glyph). HMI is specified in units of 1/120 inch.

The Vertical Motion Index (VMI) defines the distance between lines of print. It is the distance CAP moves for a linefeed operation. VMI is specified in 1/48 inch increments.

Horizontal Motion Index (HMI) ***Esc & k # h/H***

Designates the width of columns used for horizontal movement calculations.

Value(#)	=	Number of 1/120 inch increments.
Default	=	Determined by the pitch value in the default font descriptor.
Range	=	0 - 32767 (command is ignored for unsupported values)

HMI defaults to the invoked font's space character when any of the font's characteristics (orientation, character set, pitch, etc.) are changed, or when switching between primary and secondary fonts with SI and SO.

For fixed pitch fonts, HMI affects all printable characters, including the space and backspace characters. For proportionally spaced fonts, HMI may affect only the space character: if the space character glyph exists, CAP moves the width of the space character; otherwise CAP moves according to HMI.

Devices which do not have an integral number of HMI units-to-dots should implement fractional HMI units for dot addressing.

Vertical Motion Index (VMI) *Esc & l # c/C*

Sets the vertical spacing between lines of print (the vertical distance CAP will move for a linefeed.

Value(#)	=	Number of 1/48 inch increments between rows.
Default	=	8
Range	=	0 - current logical page length up to 32767 (command is ignored for unsupported values)

This command performs the same functions as Line Spacing (*Esc&l#D*), except the measurement interval is in 1/48 inch increments instead of lines per inch (lpi). Both commands set linefeed and half linefeed spacing. To convert lpi to VMI:

VMI = 48.0 / lpi

If the Page Length command (*Esc&l#P*) follows a VMI change, physical page size is recalculated. Depending on the VMI modification, the printer may request a different page size.

Devices which do not have an integral number of VMI units-to-dots should implement fractional VMI units for dot addressing.

NOTE: LJII and LJ2000 allow VMI to be set from the front panel.

Line Spacing *Esc & l # d/D*

Sets the number of lines printed per inch.

Value(#)	=	Number of lines per inch (lpi)
Default	=	6
Range	=	0 to the current logical page length up to a maximum of 32767 (command is ignored for unsupported values)

A value of 0 sets line spacing to its default of 6.

This command performs the same functions as VMI (*Esc&l#C*), except it identifies the VMI in lines per inch (lpi). Both commands set linefeed and half linefeed spacing. To convert VMI to lpi:

lpi = 48.0 / VMI

NOTE: LJII and LJIII provide line spacings of 1-4, 6(default), 8, 12, 16, 24, and 48 lpi.

PJ provides line spacings of 6(default), 8, and 9 lpi.

HP-GL/2 Mode:

Extra Space ES

ES [width [,height]];

ES; Default turns extra space transforms off.

width, height = rc

This command changes the amount of horizontal or vertical space between label characters or lines. The spacing of both fixed-spaced and proportionally-spaced fonts may be changed using this command.

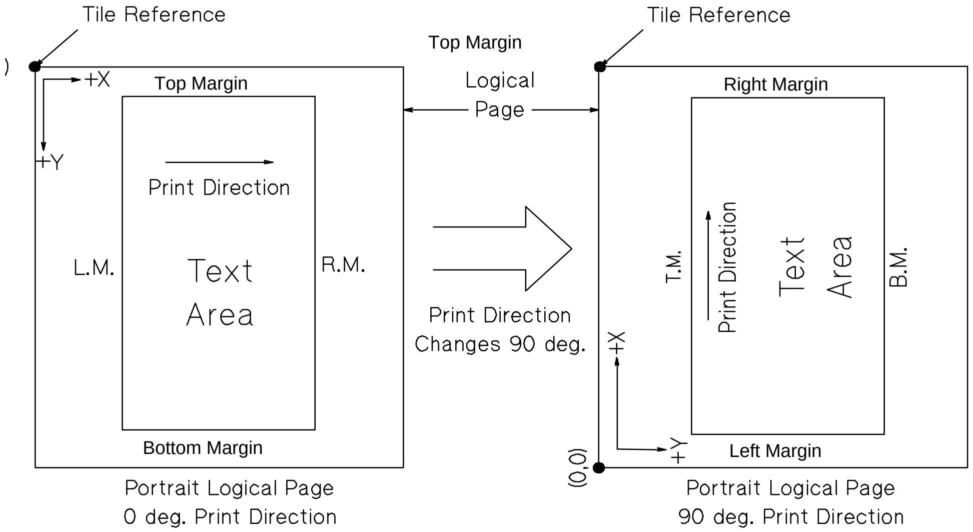
The units of *width* and *height* are fractions of the character cell delta X width and linefeed height. Character images are not distorted by this command.

For example, sending "ES 0.15;" under default conditions causes the normal amount of space which would occur between the left edge of any character cell and the left edge of the next cell (the normal spacing between any two characters) to increase by 15 percent, that is, to 1.15 times its normal value. The label length will also increase by 15 percent.

For proportionally-spaced fonts, the ES percentage will apply to the character cell size; thus cell size will vary from character to character.

If the parameter is positive, the characters will be drawn further apart; if the parameter is negative, the characters will be drawn closer together (possibly overlapping).

Margins and Text Area



Text Area and Margins

Margins are related to the logical page, not the physical page. Since the printer can only address the area within the logical page, the actual distance from the text area to the edge of the physical page is the margin plus the distance between the edge of the physical page and the edge of the logical page.

Margins represent a physical position and do not change with subsequent HMI changes. However, margins do correspond to different character positions for different horizontal print pitches.

NOTE Text margins and graphics margins may differ. See Chapter 13 for information on raster graphics margins.

Printing outside the Margins: The only way to move CAP outside the margins is with the CAP Move commands (e.g., *Esc&a#C* and *Esc&a#H*). BS is ignored when CAP is at the left margin.

Once outside the right or left margins, all features except CR function normally to the edge of the logical page. For example, if CAP is to the right of the right margin, data is printed and CAP updated accordingly.

Text Area: Area defined by the left margin, right margin, top margin, and text length. The text area is entirely contained by the logical page: it may be the same size, or be restricted by margins within the logical page. Characters can be printed outside the text area under the following conditions:

- If perforation skip is disabled, characters may be printed between the bottom of the text area and the top of the text area on the next page.
- CAP Move commands can position CAP outside the text area and allow printing anywhere within the logical page.

Left Margin: Distance between the left edge of the logical page and the left edge of the text area. The term "left margin" may also refer to left edge of the text area.

Right Margin: Distance between the left edge of the logical page and the right edge of the text area. The term "right margin" may also refer to right edge of the text area.

Top Margin: Distance between the top of the logical page and the top of the text area. The term "top margin" may also refer to the top edge of the text area.

Bottom Margin: Distance from the bottom of the text area to the bottom of the logical page. There is no bottom margin command. The bottom margin can only be set indirectly by setting text length when perforation skip mode is enabled. The bottom margin may be calculated by subtracting the text length (at the current VMI) from the logical page length.

Text Length: Distance from the top margin to the bottom of the text area. Text length has meaning only if perforation skip mode is enabled. When perforation skip is enabled, text length defines the bottom margin.

Perforation Region: Distance from the bottom of the text area on the physical page to the top of the text area on the next physical page. Disabling perforation skip mode causes text to be printed to the end of the page, into the unprintable region, and onto the top of the next page; text length and top margin are ignored. Historically, this term originated with continuous feed devices.

Top of Form: After power on or reset, CAP moves to the baseline of the first row of characters. This position, called *top of form,* is 3/4 of a line below the top margin:

```
top of form = top margin + (3/4 * line spacing)
```

The phrase "move to top of form" means if the CAP is not at top of form, move it to the top of form of the next logical page.

In printers that have no means of sensing the edges of a sheet there is no guarantee that the top of form will be a specified distance from the physical edge of the sheet.

Printable Area: This is area of the physical page in which the printer is able to place a dot; this is usually determined by the technology of the printing device. Although the text area may be larger than the printable area, text outside the printable area is lost.

Clear Horizontal Margins *Esc 9*

Resets the left and right margins to their default positions. CAP is unchanged.

Left Margin	=	left bound of the logical page (column 0)
Right Margin	=	right bound of the logical page.

Left Margin *Esc & a # l/L*

Sets the left margin to the left edge of the specified column.

Value(#)	=	Column number
Default	=	Column 0 (the left bound of the logical page)
Range	=	0 to the right margin

If CAP is to the left of the new left margin, it moves to the new left margin; otherwise, this command does not affect CAP. Attempts to set the left margin to the right of the right margin should be ignored; however, left and right margins can be set at the same location.

NOTE Column 0 is the first column on the left edge of the logical page, not the physical page.

Right Margin *Esc & a # m/M*

Sets the right margin to the right edge of the specified column.

Value(#)	=	Column number
Default	=	Logical page right bound
Range	=	Current left margin - logical page right bound

This command is ignored for columns preceding the left margin. Specifying a column beyond the right logical page limit sets the right margin to the right logical page limit. If CAP is to the right of the new right margin, CAP is moved to the new right margin; otherwise this command does not affect CAP.

Characters whose cells overlap with the right margin are not printed unless end-of-line wrap is enabled: then the character prints at the left margin on the next line. Attempts to set the right margin to the left of the left margin should be ignored; however, left and right margins can be set to the same location.

NOTE: LJIII and LJ2000 print cells overlapping with the right margin. LJ, LJ+, LJ500, and LJII clip overlapping characters.

Top Margin *Esc & l # e/E*

Specifies the distance between the top of the logical page and the top of the text area.

Value(#)	=	Number of lines to the top of the logical page
Default	=	½"
Range	=	0 to the length of logical page (command is ignored for other values)

The top margin is specified in lines whose spacing is determined by the current line spacing. This command is ignored if the current line spacing is 0, or if a value beyond the current logical page length is received.

CAP is not moved if it is below the new top margin. If CAP is above the new right margin, CAP is placed at the baseline of the first row of characters. This position, called **top of form**, is calculated as follows:

top margin in inches + (.75 x VMI in inches)

The first line of the logical page is always line 0. The vertical position of the first line of print on the logical page is determined by the current top margin and VMI, using the following:

first line in inches to the baseline = top margin in inches + (.75 x VMI) / 48.0

The top margin represents a physical position: once the top margin is set, it is unaffected by subsequent changes in line spacing.

NOTE: Since this command defaults text length, it should precede the Text Length command (*Esc&l#F*).

NOTE: DJ defaults to ½" with perforation skip mode on, and 0" with perforation skip mode off.

Text Length *Esc & l # f/F*

Sets length of the text area in lines from the top margin.

Value(#)	=	Number of lines
Default	=	Logical page length - top margin - 1/2 inch *
Range	=	0 to the top margin

* If the calculation is negative, text length defaults to (logical page length - top margin).

This command is ignored if current VMI is 0 or a text length greater than (logical page length - top margin) is requested. A value field of 0 defaults text length.

Text length determines the bottom margin when perforation skip is enabled. Text length is ignored when perforation skip is disabled.

Perforation Skip Mode *Esc & l # l/L*

Controls perforation skip mode.

Value(#)	=	0	-	Disables perforation skip mode
	=	1	-	Enables perforation skip mode
Default	=	1 *		
Range	=	0,1 (command is ignored for other values)		

* Historically, the default has been device dependent; "1" is recommended.

The perforation region extends from the bottom of the text area to the top margin of the next page. Text Length (*Esc&l#F*) determines the size of the perforation skip region (and, indirectly, the bottom margin).

If perforation skip is enabled, text length sets the bottom margin. Printing ends at the specified text length, and then starts again at the top margin of the next page.

If perforation skip is disabled, text is printed to the end of the page and onto the next page. Text length and top margin are ignored. Text in the unprintable region may be lost.

NOTE Disabling perforation skip may cause text outside of the printable area to be lost. To prevent this, perforation skip should be enabled. Changing perforation skip mode defaults top margin and text length.

Chapter 4: Raster Graphics

Raster Graphics

Introduction to Raster Graphics

The PCL language defines raster graphics in the following terms:

Dot

A dot is the smallest mark a printer can make. Its size and spacing are device specific.

Pixel

A pixel is the smallest definable picture element in the source image. At maximum machine resolution, each pixel consists of one dot. When scaling up, or at lower resolutions, a pixel may consist of more than one dot.

Bit

Bits refer to the binary 1's and 0's in the source data. In black and white printers, a single bit defines a pixel; a "1" bit prints a black pixel and a "0" bit prints a white pixel.

Bits	**Pixel Rows**
11111111	• • • • • • • •
10101010	• • • •

Generally, a color pixel requires more bits: e.g., two bits per pixel for four colors, three bits per pixel for eight, etc.

Raster Image

A raster image is composed of pixel rows, each of which describes a strip of the image one pixel high. A pixel row is transferred to the printer as a string of bytes. Rows are organized from top to bottom: the first data row becomes the top pixel row of the image.

Raster Area

The PCL raster graphics system uses the concept of a raster graphics picture, or *raster area*. The raster area is a bounded region. Within the boundary the printer will zero-fill missing rows and short rows, and clip any raster line outside that boundary.

Raster height extends vertically from CAP to one of the following:

- the distance specified by the Source Raster Height command (*Esc*r#T*).

- the row preceding an implicit or explicit End Raster Graphics (*Esc*rC*).
- the lower edge of the printable area.

Raster width extends from the *left graphics margin* (CAP or the left edge of the logical page), and is limited to one of the following:

- the distance specified by the the Source Raster Width command (*Esc*r#S*).
- the right edge of the printable area if width is not specified.
- the right edge of the logical page if width is specified.

The raster area is independent of text margins and perforation skip mode. Raster graphics never causes a page eject; at the end of the physical page, CAP is set to the logical page boundary and the rest of the graphic discarded.

Raster Resolution

Machine resolution is the dots per inch (dpi) a device is capable of printing. Some devices allow different *requested resolutions* up to the machine resolution. At lower requested resolutions, each pixel consists of more than one dot; but dot-spacing remains the same and the same number of dots-per-inch are printed. When 150 dpi is requested in a 300 dpi machine, 300 dpi is still printed; but the pixel is now composed of four dots, and is twice as wide and twice as tall as a pixel printed at 300 dpi. Since the spacing between dots is always the same, changing the resolution changes the size of the printed raster image.

bit	300 dpi	150 dpi
1	•	•• ••

Requesting a lower resolution does not by itself cause less detail to be printed. The printer still prints at its highest resolution; it just prints the image larger. Using the same data, a 150 dpi image is printed four times as large as a 300 dpi image. To keep the destination image the same size, it is necessary to send the printer one-fourth the information.

Data Compression

Once a rectangular raster area is defined, compression methods can minimize data transmission. The PCL language provides several data compression methods (*Esc*b#M*). Data transfer can also be reduced by sending empty rows (*Esc*b0W*), or by offsetting the left graphic margin (*Esc*r1A)* or the vertical starting point (*Esc*b#Y*).

Raster Graphics Command Sequence

The normal execution sequence for raster commands is shown below.

Configure Image data	*Esc*v#W*
Palette create	*Esc*v#A,Esc*v#B,Esc*v#C*
Set Forground color	*Esc*v#S (set forgound color to black)*
Logical Operations	*Esc*l#O*
Raster Presentation	*Esc*r#F*
Raster Resolution	*Esc*t#R*
Source Raster Height	*Esc*r#T*
Source Raster Width	*Esc*r#S*
Start Raster Graphics	*Esc*r#A*
Raster Compression	*Esc*b#M*
Transfer Raster Data	*Esc*b#W[raster data]* or *Esc*b#V[raster data]*
...	

Y Offset	*Esc*b#Y*
Raster Compression	*Esc*b#M*
Transfer Raster Data	*Esc*b#W[raster data]* or *Esc*b#V[raster data]*
...	
Raster Compression	*Esc*b#M*
Transfer Raster Data	*Esc*b#W[raster data]*
...	
End Raster Graphics	*Esc*rC*

Note that color space, palette, foreground, presentation, resolution, height, and width are all set outside the *start...data...end* sequence — that is, before Start Raster Graphics; they should not be sent again until after End Raster Graphics. The entire picture is sent during the *start...data...end* sequence, choosing the most effective compression method for each row.

Raster presentation, resolution, and source raster width are all true modes, since they remain in their specified state until explicitly changed by issuing the command again or reset to their default values by a soft reset or power cycle. Raster compression, source raster height, and left graphics margin are reset by End Raster Graphics (*Esc*rC*).

Raster Mode

After a Start Raster Graphics (*Esc*r#A*) command, the printer enters a restricted state called *raster mode*. Raster mode locks out commands that would affect rendering of the graphics image. These commands remain locked out until raster mode is terminated.

Commands Locked Out of Raster Mode

The following raster commands are ignored in raster mode:

Assign Color Index	*Esc*v#I*
Color Component (first)	*Esc*v#A*
Color Component (second)	*Esc*v#B*
Color Component (third)	*Esc*v#C*
Configure Image Data	*Esc*v#W*
Destination Raster Height	*Esc*t#H*
Destination Raster Width	*Esc*t#V*
Foreground Color	*Esc*v#S*
Gamma Correction	*Esc*t#I*
Logical Operation	*Esc*l#O*
Push/Pop Palette	*Esc*p#P*
Raster Presentation	*Esc*r#F*
Raster Resolution	*Esc*t#R*
Render Algorithm	*Esc*t#J*
Scale Algorithm	*Esc*t#K*
Simple Color	*Esc*r#U*
Source Raster Height	*Esc*r#T*
Source Raster Width	*Esc*r#S*
Start Raster	*Esc*r#A*
User-defined Dither Matrix	*Esc*m#W*

Commands Allowed in Raster Mode

The following commands may be used in raster mode:

Compression Method	*Esc*b#M*
Transfer Raster Data by Row/Block	*Esc*b#W*
Transfer Raster Data by Plane	*Esc*b#V*
Y Offset	*Esc*b#Y*

All other escape sequences end raster mode.

Note: Forground color interacts with the raster graphics (more detail at the end of this chapter under *raster graphics interactions*) we suggest setting foreground color to black prior to sending raster graphics. Y offset uses index 0. if foreground color is index 0 there is a problem if the defualt render algorythm is used. The combination of the render algorythm and the pattern mora patterns to apeer in the fill area or the raster area.

Raster Data Transfer

The Transfer Raster Data commands (*Esc*b#V*, *Esc*b#W*) define how many bytes are to be interpreted as binary raster data rather than ASCII data.

Transfer Raster Data by Plane (*Esc*b#V*) is used when the data is encoded by plane, as specified by the Simple Color command (*Esc*r#U*) or the Configure Image Data command (*Esc*v#W*). Usually, *Esc*b#V* sends each plane in the row except the last, and *Esc*b#W* sends the last plane and advances the row.

The Transfer Raster by Row command (*Esc*b#W*) moves CAP to the next pixel row after execution. It is used in single-color mode, for the last plane in a multi-plane row, or for color raster transfer when the data is encoded by pixel.

Transfer Raster Data by Plane *Esc * b # V[Data]*

Sends a plane of data to the printer and advances to the next plane, but not to the next row.

Value(#)	=	Number of bytes in the data field
Default	=	NA
Range	=	0 - 32767

The number of planes per row is specified by the Simple Color command (*Esc*r#U*) or the Configure Image Data command (*Esc*v#W*). The first plane sent represents the least significant bit in the pixel.

*Esc*b#V* does not advance the row. Only *Esc*b#W* can advance the row.

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

NOTE: The color of zero is defined by the current palette if pixel encoding mode is "index by plane".

If fewer planes are sent than specified and the row ended early by *Esc*b0W*, the undefined planes are zero-filled (except in compression method 3).

If more planes are sent than specified, the extra planes are ignored and the binary data discarded. If *Esc*b#W* is one of the extra planes, its data is ignored and the row incremented.

If an End Raster Graphics command (*Esc*rC*) is received before the row is completed, the output is not rendered and the row is incremented.

The data field is interpreted according to the current Compression Method (*Esc*b#M*).

Transfer Raster Data by Row/Block *Esc * b # W[data]*

Transfers the number of bytes specified in the value field to the printer in a row by row or block format, depending on the current compression method *(Esc*b#M)*, then moves CAP to the next row.

Value(#)	=	Number of bytes in the raster row/block
Default	=	NA
Range	=	$0 - 2^{31}-1$ (Command is ignored for out of range values)

This is the only data transfer command used in single-color devices or multi-color devices where the data is encoded by pixel rather than by plane. Because this command advances the row, it is also used for the last plane in a multi-plane row, or for single-plane rows.

CAP is updated for each row in both row or block formats. For row formats, CAP is reset to the left graphics margin at the next raster row. For block formats, at the end of each row CAP is reset to the initial X coordinate of the raster block, and the Y coordinate is incremented. For color data, the plane counter in a multi-plane row is reset.

The data sent can vary from row to row and is independent of source raster width. If the picture width can accommodate more data than is specified by *Esc*b#W*, the undefined area is zero-filled. If more than one plane per row is specified, undefined data planes are filled with zeros. Zeros are defined by the current palette. Black and white devices interpret zeros as white.

Data is limited to the current logical page. Data that would be placed beyond the logical page or within the unprintable region is truncated but does affect the seed row (see Compression method *Esc*b#M*)

The data field is interpreted according to the current Compression Method (*Esc*b#M*).

Color Raster Examples

This example shows how to use 24-bit RGB color direct by pixel. In addition, it shows how to get CYM color by reversing the black/white reference. The raster image is composed of blocks of different colors: red, green, blue, and several blocks of nonprimary colors

We first send the image as 24-bit RGB raster. Then we send the same image again using the error diffusion and monochrome error diffusion digital halftoning algorithms. We send it for the fourth time after reversing the black/white reference. Thus instead of getting red, green, and blue blocks, we get cyan, magenta, and yellow blocks respectively. Reversing the black/white reference is useful if the source raster data is in CYM.

This example comes from the 24bitcol.c file include with this manual.

```
/***** Universal Exit Language/Start of PJL Command *****/
Esc%-12345X@PJL COMMENT *** Beginning of PCL Job ***
@PJL ENTER LANGUAGE = PCL

/***** Initialize Printer *****/
EscE                                    /* send an esc E to reset printer */
```

```
Esc&l0O                              /* set orientation to portrait */
Esc&l0E                              /* set top margin to 0 */
Esc*t300R                            /* set graphics resolution to 300 dpi */
Esc*r200T                            /* set raster height to 200 pixels */
Esc*r1400S                        /* set raster width to 1400 pixels */
Esc*v6W00h 03h 00h 08h 08h 08h       /* Configure Image Data */
                                     COLOR_MODEL, ENCODING_MODE,
                                     BITS_PER_INDEX, BITS_PER_RED,
                                     BITS_PER_GREEN, BITS_PER_BLUE
                                     /* Configure image data for relative color
                                     model, set pixel encoding mode to direct
                                     by pixel, set bits per index to be 0
                                     (ignored parameter), and set 8 bits for
                                     each primary color */

/**** Draw 1st raster image ****/
Esc*p300x300Y                        /* position the cursor 1 inch from left margin
                                     and 1 inch below top margin */
     Esc*r1A                         /* Start raster graphics with the current
                                     x position as the left graphics margin. */
     Esc*b0M                         /* Mode 0 compression (uncompressed) */
     Esc*b4200W                      /* Transfer raster data by row with 4200
                                     bytes per row.
                                     3 bytes/pix * 1400 pix/row = 4200
                                     bytes/row */

     *** This section is shown in C code to show how Raster data is formated****
     for (i=0;i<200;i++)          /* red block */
       fprintf(prn, "%c%c%c", 255,0,0)
     for (i=0;i<200;i++)          /* Green block */
       fprintf(prn, "%c%c%c", 0,255,0)
     for (i=0;i<200;i++)          /* Blue block */
       fprintf(prn, "%c%c%c", 0,0,255)
     for (i=0;i<200;i++)          /* send 4 other blocks of color */
       fprintf(prn, "%c%c%c", 216,218,50);
     for (i=0;i<200;i++)
       fprintf(prn, "%c%c%c", 249,120,15);
     for (i=0;i<200;i++)
       fprintf(prn, "%c%c%c", 158,81,200);
     for(i=0;i<200;i++)
       fprintf(prn, "%c%c%c", 45,120,249);

     Esc*b3M                         /* Switch to mode 3 (delta row encoded)
                                     compression and send the rest of the
                                     rows (which are identical). */

     **** Repeat command 200 times  *****
     Esc*b0W                      /* Repeat previous row */

     Esc*rC                          /* Signify the end of raster transfer */

/**** Draw 2nd raster image Same raster with Error Diffusion****/
Esc*p300x+100Y                       /* Position the cursor 1" from left margin
                                     (absolute movement) and 1/3" below current
```

```
                                                position (relative movement). */
Esc*t4J                                         /* Select Error Diffusion algorithm */

    Esc*r1A                                     /* Start raster graphics with the current
                                                x position as the left graphics margin. */
    Esc*b0M                                     /* Mode 0 compression (uncompressed) */
    Esc*b4200W                                  /* Transfer raster data by row with
                                                4200 bytes per row.
                                                3 bytes/pix * 1400 pix/row = 4200
                                                bytes/row */

    *** This section is shown in C code to show how Raster data is formated****
    for (i=0;i<200;i++)                     /* red block */
      fprintf(prn, "%c%c%c", 255,0,0)
    for (i=0;i<200;i++)                     /* Green block */
      fprintf(prn, "%c%c%c", 0,255,0)
    for (i=0;i<200;i++)                     /* Blue block */
      fprintf(prn, "%c%c%c", 0,0,255)
    for (i=0;i<200;i++)                     /* send 4 other blocks of color */
      fprintf(prn, "%c%c%c", 216,218,50);
    for (i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 249,120,15);
    for (i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 158,81,200);
    for(i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 45,120,249);

    Esc*b3M                                     /* Switch to mode 3 (delta row encoded)
                                                compression and send the rest of the
                                                rows (which are identical). */

    **** Repeat command 200 times  *****
    Esc*b0W                                 /* Repeat previous row */

    Esc*rC                                      /* Signify the end of raster transfer */

/**** Draw 3rd raster image Same raster with Monochrome Error Diffusion****/
Esc*p300x+100Y                                  /* Position the cursor 1" from left margin and
                                                1/3" below current position. */
Esc*t5J                                         /* Select Mono Error Diffusion algorithm */

    Esc*r1A                                     /* Start raster graphics with the current
                                                x position as the left graphics margin. */
    Esc*b0M                                     /* Mode 0 compression (uncompressed) */
    Esc*b4200W                                  /* Transfer raster data by row with 4200
                                                bytes per row.
                                                3 bytes/pix * 1400 pix/row = 4200
                                                bytes/row */

    *** This section is shown in C code to show how Raster data is formated****
    for (i=0;i<200;i++)                     /* red block */
      fprintf(prn, "%c%c%c", 255,0,0)
    for (i=0;i<200;i++)                     /* Green block */
      fprintf(prn, "%c%c%c", 0,255,0)
```

```
        for (i=0;i<200;i++)                 /* Blue block */
          fprintf(prn, "%c%c%c", 0,0,255)
        for (i=0;i<200;i++)                 /* send 4 other blocks of color */
          fprintf(prn, "%c%c%c", 216,218,50);
        for (i=0;i<200;i++)
          fprintf(prn, "%c%c%c", 249,120,15);
        for (i=0;i<200;i++)
          fprintf(prn, "%c%c%c", 158,81,200);
        for(i=0;i<200;i++)
          fprintf(prn, "%c%c%c", 45,120,249);

        Esc*b3M                                 /* Switch to mode 3 (delta row encoded)
                                                compression and send the rest of the
                                                rows (which are identical). */

        **** Repeat command 200 times  *****
        Esc*b0W                             /* Repeat previous row */

        Esc*rC                                  /* Signify the end of raster transfer */

/**** Draw 4th raster image Same raster with the black/white references reversed.****/
Esc*t3J                                         /* go back to ordered dithering */

/**** Configure Image Data for reversing black/white reference ****/
ESC*v18W                                        /* Configure Image Data */
                                                COLOR_MODEL, ENCODING_MODE,
                                                BITS_PER_INDEX, BITS_PER_RED,
                                                BITS_PER_GREEN, BITS_PER_BLUE,
                                                0,0,0,0,0,0,0,255,0,255,0,255);
                                                /* Configure image data and reverse the
                                                black/white reference for each of R,G,B.
                                                The references are written as 2 byte values
                                                thus we must write 12 bytes to write the
                                                6 reference values.  The white reference
                                                is 0 = 00000000 00000000 (binary) and the
                                                black reference is 255 = 00000000 11111111
                                                (binary) */

Esc*p300x+100Y                                  /* Position the cursor 1" from left margin and
                                                1/3" below current position. */

        Esc*r1A                                 /* Start raster graphics with the current
                                                x position as the left graphics margin. */
        Esc*b0M                                 /* Mode 0 compression (uncompressed) */
        Esc*b4200W                              /* Transfer raster data by row with 4200
                                                bytes per row.
                                                3 bytes/pix * 1400 pix/row = 4200
                                                bytes/row */

        *** This section is shown in C code to show how Raster data is formated****
        for (i=0;i<200;i++)                 /* red block */
          fprintf(prn, "%c%c%c", 255,0,0)
        for (i=0;i<200;i++)                 /* Green block */
          fprintf(prn, "%c%c%c", 0,255,0)
```

```
    for (i=0;i<200;i++)                /* Blue block */
      fprintf(prn, "%c%c%c", 0,0,255)
    for (i=0;i<200;i++)                /* send 4 other blocks of color */
      fprintf(prn, "%c%c%c", 216,218,50);
    for (i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 249,120,15);
    for (i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 158,81,200);
    for(i=0;i<200;i++)
      fprintf(prn, "%c%c%c", 45,120,249);

    Esc*b3M                               /* Switch to mode 3 (delta row encoded)
                                          compression and send the rest of the
                                          rows (which are identical). */

    **** Repeat command 200 times  *****
    Esc*b0W                            /* Repeat previous row */

    Esc*rC                                /* Signify the end of raster transfer */

EscE                                      /* Reset printer and eject page */
/***** Universal Exit Language/Start of PJL Command *****/
Esc%-12345X
```

Raster Compression

Raster images take a large amount of memory and processing time. Generally, raster data should be compressed. The Compression Method command (*Esc*b#M*) provides five different ways of compressing data. White-space transfer can also be reduced by sending empty rows and planes (*Esc*b0W*, *Esc*b0V*), or by offsetting the left graphic margin (*Esc*r1A)* or the vertical starting point (*Esc*b#Y*).

Compression Method   *Esc * b # m/M*

Determines how raster data is interpreted in the *Transfer Raster Data by Row* or *Transfer Raster Data by Plane* commands. The compression method stays in effect until explicitly changed to another method or reset by *EscE* or *Esc*rC*.

Value(#)	=	0	-	Unencoded
	=	1	-	Run-length encoding
	=	2	-	Tagged Image File Format (TIFF) rev 4.0 "Packbits" encoding
	=	3	-	Delta row encoding
	=	4	-	(reserved)
	=	5	-	Adaptive encoding
Default	=	0		
Range	=	0 to 5		

Method 0 - Unencoded

This is a simple binary transfer: no compression.

Method 1 - Run-Length Encoding

Method 1 interprets raster data as byte pairs. The first byte tells how many times the second byte is repeated, i.e., a repetition count. The second byte is the data to be printed. The repetition count can be 0 through 255. A count of 0 means the byte pattern occurs once and is not repeated; a count of 1 means the pattern occurs twice; and a count of 255 means the pattern occurs 256 times.

NOTE: Method 1 requires byte pairs which means value fields should always have even number of bytes. A Transfer Raster Data command (*Esc*b#V* or *Esc*b#W*) with an odd value field is ignored and the data discarded.

Method 2 - Tagged Image File Format Encoding (TIFF revision 4)

Method 2 ("packbit" encoding) combines methods 0 and 1, with blocks of repeated bytes and blocks of *literal* (unencoded) bytes. A control byte precedes the raster data; it indicates the number of succeeding bytes, and whether they are repeated or literal.

The control byte is in two's complement format. If bit 7 is set (byte = -1 to -127), the following bytes are repeated. The control byte's absolute value is the number of repetitions (occurrences = repetitions + 1). For example, a control byte of -5 means the subsequent byte is repeated 5 times (6 occurrences).

If bit 7 of the control byte is zero (byte = 0 to 127), the following bytes are literal. The number of succeeding data bytes is 1 + the value of the control byte. A control byte of 0 means 1 literal byte follows; a control byte of 6 means 7 literal bytes follow.

A control byte of -128 is ignored; the next byte is then treated as a control byte.

It is more efficient to code two consecutive identical bytes as a repeated byte, unless these bytes are preceded and followed by literal bytes. Three-byte repeats should always be encoded as replicate runs.

NOTE: The row length indicated by the Transfer commands (*Esc*b#V* or *Esc*b#W*) has precedence if it is met before the literal run count.

EXAMPLES OF METHODS 0-2

The following examples show how methods 0-2 can be used to send the raster row below:

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

Compression method 0 - Unencoded

```
Esc*r1A
Esc*b0m7WUUUUATT
Esc*rC
```

Compression method 1 - Run-length encoding

```
Esc*r1A
Esc*b1m6W(3)U(0)A(1)T
Esc*rC
```

Compression method 2 - TIFF encoding

```
Esc*r1A
Esc*b2m6W(-3)U(0)A(-1)T     -or-     Esc*b2m6W(-3)U(2)ATT
Esc*rC
```

Method 3 - Delta Row Encoding

This method replaces only bytes in the current row that are different from the preceding row (**seed row**). Unreplaced bytes are replicated from the seed row. The current row then becomes the seed row.

A delta compression string consists a command byte and the replacement (**delta**) bytes:

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

The command byte itself has two parts: the number of consecutive replacement bytes that follow, and the left offset from the current byte position.

7 5	4 0
Number of replacement bytes (1-8)	Left offset from the current byte (0-30)

The upper three bits of the command byte contain the *replacement count*: the number of consecutive replacement bytes that follow (000 = 1, 111 = 8).

The lower five bits contain the offset from the current byte to the next replacement byte. The *current byte* is the byte following the last replacement byte or, at the beginning of a row, the left graphics margin. An offset of 0 indicates the current byte; an offset of 1 indicates the byte following the current byte.

For example, assume the current byte is the first byte in the row. An offset of 7 skips bytes 0 through 6, and a replacement count of 5 replaces bytes 7 through 11. The new current byte is 12. A second offset of 3 skips bytes 12, 13, and 14; byte 15 is the next byte to be replaced.

The values of the offset have the following definitions:

0 to 30: The offset is 0 to 30.

31: The offset is 31 or greater. If the offset is 31, an additional **offset byte** follows the command byte. The offset in the command byte is added to the offset byte. If the offset byte is 0, the offset is 31; if the offset byte is 255, additional offset bytes follow. The last offset byte will have a value less than 255. All the offset bytes are added to the offset in the command byte to get the offset value. For example, if there are two offset bytes, and the last byte contains 175, the total offset would be: 31 + 255 + 175 = 461.

If more than eight delta bytes are needed, additional command byte/delta bytes are added:

[(Command Byte)(1-8 Delta Bytes)**][**(Command Byte)(1-8 Delta Bytes)**]. . .**

The seed row is zeroed when raster mode is entered. Every raster transfer affects the seed row, regardless of the compression method. For example, an *Esc*b0W* while in compression method 0 zeros the seed row. This allows method 3 to be combined with other methods in order to achieve better compression performance.

A vertical offset also affects the seed row: the Y Offset command (*Esc*b#Y*) zero-fills skipped rows, and zeros the seed row.

Method 3 operates on each plane independently, and a separate seed row is maintained for each graphic plane. However, a Y Offset affects all planes and seed rows simultaneously.

The examples below show the effects of the following escape sequences in method 3 compression:

*Esc*b0W*	Repeat the previous row. The seed row is unchanged.
*Esc*b1Y*	Move down one raster row. Zero the seed row.
*Esc*b0Y*	Zero raster rows. Zero the seed row.

The seed row width is set to the source raster width at graphics entry.

The byte count of the value field of the Transfer Raster Data (*Esc*b#V*, *Esc*b#W*) command has precedence over the byte count in the command byte. And if the Transfer Raster Data byte count is less than the number of bytes that can be replaced, the byte count again has precedence; e.g., if the byte count is 10, but the data field has 3 bytes, only 3 bytes are replaced.

If the last byte indicated by the value field of the Transfer Raster Data is a control byte, that byte is ignored. Therefore, *Esc*b1W* does not affect the seed row, but causes the previous row to be repeated.

DELTA ROW EXAMPLE:

The following data is compressed using method 3. Italicized bytes are those needing replacement — bytes different from the seed row. Graphics data is binary.

Byte #:	1	2	3	4	5
Row 1	00000000	***11111111***	00000000	00000000	00000000
Row 2	00000000	11111111	***11110000***	00000000	00000000
Row 3	***00001111***	11111111	11110000	***10101010***	***10101010***

*Esc*r1A*

Start Raster Graphics initializes the seed row to all zeros.

Row 1 - *Esc*b3m2W*(00000001)(11111111)

The **3m** selects method 3, and **2W** indicates that two bytes of data will follow. The upper three bits of the command byte are zero, indicating one byte will be replaced. The lower five bits contain a relative offset of 1, indicating that the replacement occurs 1 byte in from the current position. The replacement byte follows and contains 11111111.

Row 2 - *Esc*b2W*(00000010)(11110000)

The first three bits of the command byte are 0 indicating one byte will be replaced. The lower five bits contain a relative offset of 2; so the replacement will occur 2 bytes from the current position. The replacement byte 11110000 follows.

Row 3 - *Esc*b5W*(00000000)(00001111)(00100010)(10101010)(10101010)

Three bytes are replaced using two commands. The first three bits of the first command byte are zero, indicating a single byte replacement, and the next five bits are zero, indicating a relative offset of zero. The replacement byte 00001111 follows.

The first three bits of the second command byte are 001, indicating the replacement of two bytes; and the lower five bits contain a relative offset of two. The two replacement bytes (10101010)(10101010) follow the command byte.

Method 5 - Adaptive Compression

NOTE: Method 5 is included for compatibility with the LaserJet IIIP.

Adaptive compression uses compression methods 0 - 3 to optimally compress an entire **block** of data. When the row data within a block is no longer optimally compressed by one method, the compression method may be changed to "adapt" to the data. Adaptive compression also allows the specification of empty or duplicate rows to skip white space or replicate identical rows within a block.

In adaptive compression, a raster image is interpreted as a block of data, rather than as individual rows. The Transfer Raster Data command (*Esc*b#W*) is sent only once, at the beginning of a raster transfer; its value field specifies the number of bytes in the block. The block size of the compressed data is limited to 32767 bytes. To transfer more bytes, more blocks can be sent.

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 that row. The following two bytes specify either the number of bytes within the row or the number of duplicate or empty rows. The format of an adaptive compression raster row is shown below:

<command byte> <# of bytes/rows><# of bytes/rows> <raster data>

The command byte designates the compression method, an empty row, or a duplicate row. Command byte values are shown below:

Value	=	0	-	Unencoded
		1	-	Run-Length Encoding
		2	-	Tagged Image File Format (TIFF) revision 4.0
		3	-	Delta Row
		4	-	Empty row
		5	-	Duplicate row
Range	=	0 to 5 (out-of-range values & data are ignored, and CAP is not advanced)		

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

For cases 0 to 3, the two-byte field, *<# of bytes/rows>*, specifies the **row length** (i.e., the total number of bytes to be transferred for that row within the raster block. For cases 4 and 5, this field defines the number of empty or duplicate rows to be encountered after the current row, including the current row. The most significant byte (high byte) of this field is sent first, followed by the least significant byte (low byte).

The maximum value for the two-byte field, *<# of bytes/rows>*, is 65535; however, the image will be clipped to the logical page.

Values 0 to 3 indicate the identical compression methods described above. Values 4 and 5 are explained below:

Empty Row: A command byte of 4 causes a row of zeros to be printed. The number of rows printed is contained in the two-byte field, *<# of bytes/rows>*, following the command byte. An empty-row operation resets the seed row to zero and updates CAP.

Duplicate Row: A command byte of 5 causes the previous row to be printed again the number of times contained in the two-byte field, *<# of bytes/rows>*, following the command byte. A duplicate-row operation updates CAP, but does not change the seed row.

Usage Guidelines For Adaptive Compression

- Compression methods cannot be mixed within one row.

- Since method 3 requires that the seed row be available when entering raster mode, the seed row is zeroed upon raster mode entry *(Esc*r#A)*. The seed row is also zeroed upon receipt and completion of each raster block.

- Within a block, every compression method or type of row updates the seed row. For example, a row compressed by method 2 updates the seed row; and an empty row zeros the seed row. Maintaining the seed row allows method 3 to be mixed with other methods to achieve optimal compression performance.

- CAP is updated with each row of the raster block. CAP is also incremented for block counts less than 3.

- The Y Offset command moves the entire block of raster data and zeros the seed row — 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 size is truncated to the block size.

- For method 1, if the row length is odd, CAP is incremented and the row data is skipped (discarded), but the seed row is unchanged.

- For method 1, a row length of zero increments CAP and zeros the seed row.

- For method 2, CAP is always incremented and a new row started when row length is satisfied, regardless of the control byte value. "Extra" bytes following the control byte are used to start the next row.

- For method 2, a row length of 1 inputs one bye from I/O and increments CAP. The data is ignored and the seed row is zeroed.

- For method 3, CAP is always incremented and a new row started when row length is satisfied, regardless of the control byte value. "Extra" bytes following the control byte are used to start the next row.

- For method 3, a row length of zero duplicates the current row and increments CAP.

- For method 3, a row length of 1 inputs one byte from I/O, duplicates the current row, and increments CAP. The data is ignored.

- For duplicate and empty rows, a row length of zero does not update CAP; but the seed row is zeroed.

- An unsupported command byte for a row causes the remaining bytes in the block to be skipped and the seed row zeroed; CAP is not incremented.

Example of Adaptive Compression

The following example demonstrates adaptive compression.

*Esc*r3F* Set presentation method to 3; and print along the width of the physical page, regardless of the logical page orientation.

*Esc*t300R* Set graphics resolution to 300 dpi.

*Esc*r1A* Start raster graphics at CAP. The seed row is initialized to all zeros.

*Esc*b5M* Set compression method to 5, Adaptive Compression.

*Esc*b84W* Transfer a raster block of data containing 84 bytes.

Five rows of raster data are sent within the block. The first three rows are compressed using method 3, Delta Row, compression. The next row is compressed by compression method 1, Run Length Encoding. The next three rows of data are specified as duplicate rows of the previous Run Length Encode row. Finally, the last three rows are compressed using method 3. CAP is updated after each raster row within the block is processed.

The raster block has the following format and HEX data:

Row	Control byte	# bytes per row	Raster data
1	03 hex	00 09 hex	E0 FF F0 00 FF FF 00 0F FF
2	03 hex	00 09 hex	E0 00 00 FF F0 0F FF 00 00
3	03 hex	00 09 hex	E0 FF F0 00 FF FF 00 0F FF
4	01 hex	00 06 hex	00 FF 05 00 00 FF
5	05 hex	00 03 hex	
6	03 hex	00 09 hex	E0 FF F0 00 FF FF 00 0F FF
7	03 hex	00 09 hex	E0 00 00 FF F0 0F FF 00 00
8	03 hex	00 09 hex	E0 FF F0 00 FF FF 00 0F FF

*Esc*rC* End raster graphics session.

Raster Scaling

NOTE: To use raster scaling, Configure Image Data (*Esc*r#W*) must be sent prior to Start Raster Graphics (*Esc*r#A*) with a value field of 2 or 3.

Start Raster Graphics (*Esc*r#A*) with a value field of 2 or 3 turns on scale mode. The raster image is rendered in the desired size independently of device resolution.

Source Raster Width (*Esc*r#S*) and Source Raster Height (*Esc*r#T*) define source size; Destination Raster Width (*Esc*t#H*) and Destination Raster Height (*Esc*t#V*) define destination size. The "scale factor" is implicitly determined from destination size, source size, and device resolution.

Specification of destination raster width and height is unnecessary for scaling, since these dimensions default to graphics margin and printable area boundaries. If these dimensions are not specified, isotropic scaling is maintained such that the entire image is rendered on the page without clipping. If only one destination dimension is specified, that dimension prevails and the other dimension is implicitly determined to maintain isotropic scaling; in this case, clipping may occur.

Destination Raster Width ***Esc * t # h/H***

Defines the width in decipoints of the destination raster picture denoted by the next Start Raster Graphics command (*Esc*r#A*) with a value field of 2 or 3.

Value(#)	=	Width in decipoints
Default	=	Right logical page boundary minus left graphics margin
Range	=	0 - 32767

Zero or absent values default 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 ***Esc * t # v/V***

Defines the height in decipoints of the destination raster picture denoted by the next Start Raster Graphics command (*Esc*r#A*) with a value field of 2 or 3.

Value(#)	=	Height in decipoints
Default	=	Bottom logical page boundary minus vertical CAP
Range	=	0 - 32767

Zero or absent values default 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 Scaling Example

This example illustrates raster image scaling. The image is printed once normally, then it is scaled down to 1/2 its original size.

The PCL commands to do scaling are:

```
Esc*r # S     This sets source raster width to # pixels
Esc*t # H     This sets destination raster width to
              # decipoints.
Esc*r 3 A     This starts raster graphics with scale mode
              turned on with the current x position as the
              left graphics margin.
```

This scales an image isotropically (by only specifying either the destination raster width or destination raster height). To scale anisotropically, both destination width and height must be specified.

The PCL commands to send landscape raster are:

1) Set logical page orientation to landscape with
   Esc&l1O
2) Set presentation mode to print in the orientation of the logical page with
   Esc*r0F

This example also illustrates how to use 3-bit color indexed by plane. This example comes from the 24bitcol.c file include with this manual.

```
Esc%-12345X@PJL COMMENT *** Beginning of PCL Job ***
@PJL ENTER LANGUAGE = PCL

/***** Initialize Printer *****/
EscE                                  /* send an esc E to reset printer */
Esc&l0O                               /* set orientation to portrait */
Esc&l0E                               /* set top margin to 0 */

   /***** Raster Setup Commands *****/
Esc*r#S     #  SOURCEWIDTH            /* set raster width in pixels */
Esc*r#T     #  SOURCEHEIGHT           /* Set raster height. */
Esc*t#R     #  RESOLUTION             /* set graphics resolution */
Esc*v6W 00h 00h 03h 01h 01h 01h /**** Configure Image Data ****/
                                      COLOR_MODEL, ENCODING_MODE,
                                      BITS_PER_INDEX,
                                      BITS_PER_RED, BITS_PER_GREEN,
                                      BITS_PER_BLUE);
/* Configure image data for relative color model, set pixel encoding mode to indexed by
plane, set bits per index to be 3, and set # bits for R,G,and B. */

/**** PRINT 1ST IMAGE -- ORIGINAL SIZE ****/
Esc*r1A                               /* Start raster graphics with the current
                                      x position as the left graphics margin. */

Esc*b#V****   #  DATA LENGTH
              *  RASTER DATA          /* This is the RED data for this raster row */
Esc*b#V****   #  DATA LENGTH
              *  RASTER DATA          /* This is the GREEN data for this raster row */
```

```
Esc*b#W****  #  DATA LENGTH
             *  RASTER DATA     /* This is the BLUE data for this raster row
                                The W terminator advances the row */

Esc*rC                          /* End of raster graphics. */
ff                              /* Send a formfeed to eject the page. */

  /**** PRINT 2ND IMAGE -- SCALED 1/2 ORIGINAL SIZE ****/
Esc*t#H    #  DESTWIDTH_SCALEDOWN
                                /* Set destination raster width.
                                We don't need to specify destination
                                raster height for isotropic scaling */
*Escr3A                         /* Start raster graphics with the current
                                x position as the left graphics margin.
                                Turn on scale mode. */

Esc*b#V****  #  DATA LENGTH
             *  RASTER DATA     /* This is the RED data for this raster row */
Esc*b#V****  #  DATA LENGTH
             *  RASTER DATA     /* This is the GREEN data for this raster row */
Esc*b#W****  #  DATA LENGTH
             *  RASTER DATA     /* This is the BLUE data for this raster row
                                The W terminator advances the row */

Esc*rC                          /* End of raster graphics. */
ff                              /* Send a formfeed to eject the page. */

EscE                            /* Send an esc E to reset printer
                                (this ejects the page too). */

/***** Universal Exit Language/Start of PJL Command *****/
Esc%-12345X
```

Driver Configuration Command

Specifies Lightness, Saturation, Under Color Removal, Scaling Algorithm, Selection of Color Mode, and Download Color Map specification to be applied to the ensuing document.

Esc * o#W[device_id function_index Arguments]

#			Specifies the number of bytes to follow (device id + function index + arguments)	
device_id	=	6	Specifies device as HP Color LaserJet	
function_index	=	0	Lightness	Argument Range [-100...100]
		1	Saturation	Argument Range [-100...100]
		2	Under Color Removal	Argument [Shadow,Highlight, Gamma]
			0-255, unsigned bytes (Under investigation for future release, not currently supported)	
		3	Scaling Algorithm	Argument [0 (Pixel Replication) 1 (Bilinear Interpolation) 2 (Modified Bilinear Interp)]
		4	Select Color Mode	Argument [0 No Adjustment 1 Process Blue 2 Vivid Graphics 3 Transparency 4 Out of Gamut(DIC) 5 CIE Lab Match(DIC)
		5	Download Color Map	ColorSpace [1 CMY 3 CIELab] MapId [See Valid MapId List] Data [14739 bytes]

Lightness: Negative values will darken (unlighten) the image, text or graphics color, but will not have any effect on black or white data. Positive values will lighten the image. Zero will turn lightness adjustment off. This replaces the old Esc*u#L command. The number of bytes of data required for this function index is three.

Saturation: Negative values will desaturate (gray) the image, text or graphics color, but will not have any effect on Black or White data. Positive values will increase the amount of saturation making the image more vivid. Zero will turn saturation adjustment off. This replaces the old Esc*u#S command. The number of bytes of data required for this function index is three.

Under Color Removal: (Under investigation for future release, not currently supported) Three values are specified; Shadow, Hilight, and Gamma. The number of bytes of data required for this function index is five:

- Shadow: unsigned byte, range [0..255]
- Highlight: unsigned byte, range [0..255]
- Gamma:in ubytes unsigned byte, range [0..255], internally scaled by 10 such that internal representation is [0..25.5]

Scaling: The backward compatible scaling algorithm is pixel replication. Bilinear Interpolation is a high quality scaling algorithm for smooth edge interpolated scaling. Modified Bilinear only interpolates when it is best to do so. The number of bytes of data required for this function index is three.

Select Color Mode: Specifies which color treatment mode to use for rendering the ensuing job.

- No Adjustment:
    - Linearization only: ie the user sees the device as a linear device.
        + Non-halftone dependent - we have linearized for each halftone we support.
        + 3-space linear response to numerically linear input. (ie Smooth gradients independent of hue)
        – HP has calibrated.
- Process Blue: Produce a more saturated rendition of the input image. (Linearization plus user preferred enhancements). Same as Vivid Graphics, without the blue mapping, Therefore blue is printed as process blue, which to most users would be referred to as purple.

- Vivid Graphics; Produces a more saturated rendition of theinput image. (Linearization plus user preferred enhancements) with the addition of mapping process blue (visually printed as purple) to a blue closer to that of a monitor.

- Transparency: Uses a map to render the best color output on transmissive media.

- Out of Gamut: Prints colors in an image that are out of gamut - all colors that are in gamut snap to white. (Supported for Device Independent Color Mode only)

- CIE Lab Match: This map performs a true color match to requested CIE Lab input. IE there are no appearance match adjustments.(Supported for Device Independent Color Mode only)

NOTE: For Screen matching, the long form of the Configure Image Data command is used and the color maps are generated internally dependent upon the monitor calibration data. IE the Driver Configuration Command is not needed.

Download Color Map: The printer supports the downloading of color adjustment maps dependent upon the halftone requested, the type of color treatment desired (including device dependent or independent), and the type of media. This is to ensure the ability of providing the best output in the future as more is known about the stability/operating point of our engines and our customer satisfaction and preference.

Valid MapId List Description MapId

Device Dependent (LinearCMY)

Description	MapId
No Adjustment:	
Cluster-No Adjust-LinearCMY	1
Disperse-No Adjust-LinearCMY	2
Scatter-No Adjust-LinearCMY	3
ErrorDiffusion-No Adjust-LinearCMY	4
Vivid Graphics:	
Cluster-VividGraphics-LinearCMY	5
Disperse-VividGraphics-LinearCMY	6
Scatter-VividGraphics-LinearCMY	7
ErrorDiffusion-VividGraphics-LinearCMY	8

Transparency:	
Cluster-Transparency-LinearCMY	9
Disperse-Transparency-LinearCMY	10
Scatter-Transparency-LinearCMY	11
ErrorDiffusion-Transparency-LinearCMY	12
Process Blue:	
Cluster-VividGraphics-LinearCMY	13
Disperse-VividGraphics-LinearCMY	14
Scatter-VividGraphics-LinearCMY	15
ErrorDiffusion-VividGraphics-LinearCMY	16
Out of Gamut:	
Cluster-OutOfGamut-LinearCMY	17
Disperse-OutOfGamut-LinearCMY	18
Scatter-OutOfGamut-LinearCMY	19
ErrorDiffusion-OutOfGamut-LinearCMY	20
CIELab Match:	
Cluster-TrueMatch-LinearCMY	21
Disperse-TrueMatch-LinearCMY	22
Scatter-TrueMatch-LinearCMY	23
ErrorDiffusion-TrueMatch-LinearCMY	24
Device Independent (LabtoCMY):	
No Adjustment:	
Cluster-No Adjust-LabToCMY	51
Disperse-No Adjust-LabToCMY	52
Scatter-No Adjust-LabToCMY	53
ErrorDiffusion-No Adjust-LabToCMY	54
Business Blue:	
Cluster-Business Blue-LabToCMY	55
Disperse-Business Blue-LabToCMY	56
Scatter-Business Blue-LabToCMY	57
ErrorDiffusion-Business Blue-LabToCMY	58
Transparency:	
Cluster-Transparency-LabToCMY	59
Disperse-Transparency-LabToCMY	60
Scatter-Transparency-LabToCMY	61
ErrorDiffusion-Transparency-LabToCMY	62
Vivid Graphics:	
Cluster-VividGraphics-LabToCMY	63
Disperse-VividGraphics-LabToCMY	64
Scatter-VividGraphics-LabToCMY	65
ErrorDiffusion-VividGraphics-LabToCMY	66
Out of Gamut:	
Cluster-OutOfGamut-LabToCMY	67
Disperse-OutOfGamut-LabToCMY	68
Scatter-OutOfGamut-LabToCMY	69
ErrorDiffusion-OutOfGamut-LabToCMY	70

CIELab Match:

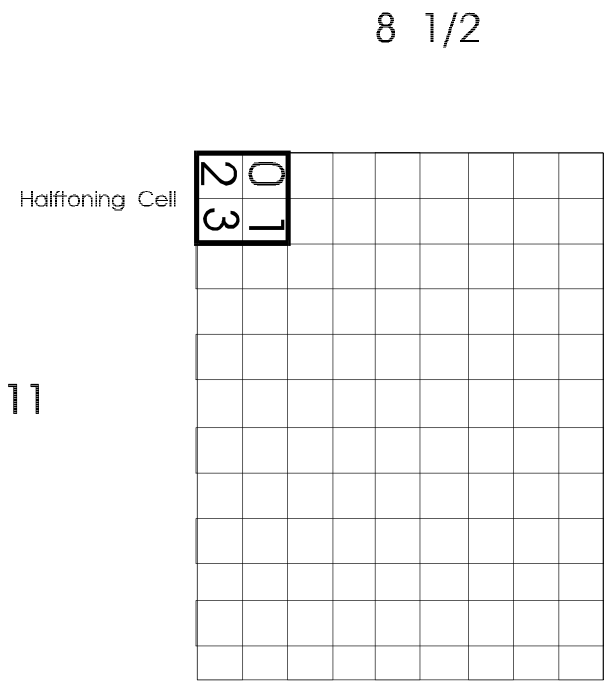
Cluster-TrueMatch-LabToCMY	71
Disperse-TrueMatch-LabToCMY	72
Scatter-TrueMatch-LabToCMY	73
ErrorDiffusion-TrueMatch-LabToCMY	74

Halftone Interactions

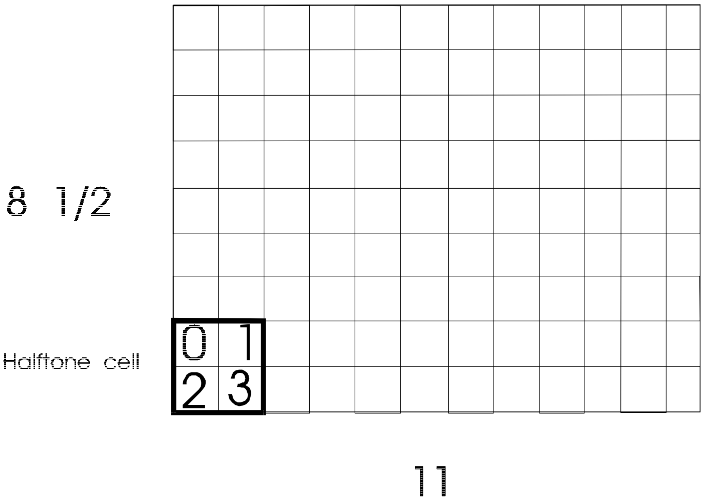
Interactions between two graphics objects or graphics states in a complicated printers graphics processor will happen. This section describes some interactions that can cause unwanted results.

Halftoning Grid

The halftoning grid in the HP Color LaserJet printer is anchored at one point. This grid does not change with the orientation. Any change in orientation does not change the orientation of the halftone. This can cause some interesting differences between landscape printing and portrait printing. The halftoning cells aligned as shown below.



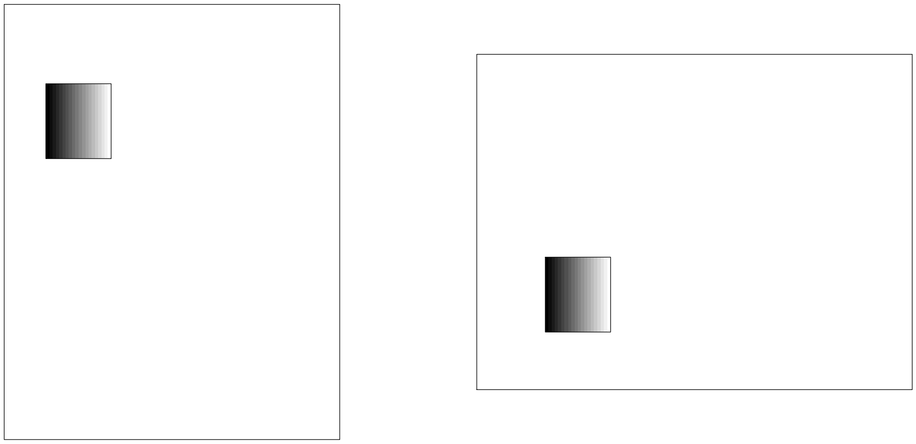
This sample halftone is number 0 -3. The 0 cell is the first cell of the halftone. The cell number 3 is the last halftone cell. The orientation of the halftone is so that the bottom half of the cell is along the long edge of a portrait letter size page. If the page was rotated to landscape orientation the cell would not rotate on that page.



As the page rotated the halftone stayed in the same orientation relative to the physical page. With the halftoning cell not changing orientation with the logical page a few errors can occur to PCL programs if they are written incorrectly. The DeskJet 1200C did rotate the halftone grid with the page rotation. If a DeskJet 1200C driver relied on the halftone grid rotating there could have some trouble.

Use patterns for patterns

The HP Color LaserJet printer allows the programmer to define their own halftoning algorithm (*see PCL5 Color Extension Technical Reference*). These halftone algorithm can be used to create patterns for fills and gradient fills, **DON'T**. This will work for documents printed in portrait. If that page is rotated to landscape the gradient will not change directions.



If the gradient was built with user defined pattern instead of user defined halftone the direction of the gradient would have rotated.

Patterns crossing halftones

Patterns can rotate with the page, but as the page rotates the halftone grid does not. With patterns that have small elements in them a color shift could occur. The reason for the shift in color is the same reason color changes when the halftone pattern changes. Elements with in the pattern will fall on different parts of the halftone cell. This changes which dots are turned on of off. This occurs mostly with ordered halftones. The solution is to use a more random halftone, such as HP Color LaserJets device best.

Chapter 5: PJL

PJL

Print Job Formation

At the beginning of every print job, basic setup commands must be issued before sending print data. Job control commands are usually grouped together and sent at job boundaries. Most print jobs consist of three segments:

- Job Control Commands
- Page Control Commands
- Print Data

Job control commands control the output of the entire print job.

Page control commands involve either a single page or groups of pages.

Print data is the data to be printed, as well as the commands that select fonts, call macros, and position data. Also included are any downloaded fonts or macros, which are usually placed in the print file immediately following the page control commands.

Structure of a Typical Job

Esc%-12345X	Universal Exit Language (UEL)[1]
@PJL Enter Language = PCL	Enter PCL[1]
EscE	Printer Reset
	Job Control Commands
PAGE 1	Page Control Commands
	Data
PAGE 2	Page Control Commands[2]
	DATA
•	•
•	•
PAGE n	Page Control Commands[2]
	DATA
EscE	Reset Printer
Esc%-12345X	UEL[1]

1 The Universal Exit Language (UEL) command and the PJL ENTER command are used only with printers that support Printer Job Language (PJL).

2 Page Control commands need only be sent once for consecutive pages within a job that have the same format.

What Are Defaults?

Default settings are programmed at the factory or selected at the control panel. Feature settings may be changed from their defaults when a job is printed; but then these job-specific settings may need to be changed by the next job. Each new job must be able to set the device to a known state. For example, if the number of copies is set to 65 copies by the previous application, the new application would also print 65 copies, which may not be desirable. Each job should clear all previous settings by performing a PCL reset (*EscE*). Starting with the user default environment at the beginning of each job eliminates the need to set each feature individually.

The printer's current feature settings are collectively called the ***Print Environment.*** Color LaserJet maintains all four of the print environments described below, with only one active at a time. Not all PCL devices have all four environments, and the feature set in each environment may vary between devices.

Factory Default Environment

This is the set of default features programmed at the factory. Factory defaults become active when the printer is first powered on — before control panel settings are changed or any escape sequences are sent from an application.

User Default Environment

This is the set of PCL features for which user defaults may be selected from the control panel. User defaults are stored in the User Default Environment and are retained even if the printer is turned off. User-defaults become active after a reset, or when the control panel settings have been altered.

Modified Print Environment

The current feature settings constitute the Modified Print Environment. Settings modified by escape sequences become part of the Modified Print Environment. The Modified Print Environment is saved during a macro call or overlay and restored upon completion.

Overlay Print Environment

The Overlay Print Environment becomes active whenever an automatic macro overlay is enabled (the modified print environment is saved). Upon completion of the macro overlay, the saved modified print environment again becomes active.

Print Environment Hierarchy

Control panel settings override the factory default environment, creating the user default environment. Printer commands from an application override the user default environment, creating the modified print environment. And finally, the overlay print environment can override the modified print environment. Any feature not set by the active environment defaults to the value set in the next lower environment in the hierarchy.

PCL5 Mode

Reset *Esc E*

Performs a reset on the printer (this is NOT a hardware reset). The reset:

- Prints all data received before the reset.
- Moves CAP to the default left margin at the top of form on the next page.
- Resets all programmable features to their user default values.
- Deletes temporary fonts.
- Does not delete downloaded permanent features.
- Defaults the PCL5 palette
- Defaults the PCL5 logical page
- Defaults the logical page orientation
- Defaults the PCL5 print direction
- Defaults the picture frame
- Erases temporary downloaded patterns, symbol sets, and macros.
- Clears the color palette stack.
- Defaults the color modification tables
- Defaults the render algorithm
- Defaults the viewing illuminants
- Defaults the gamma correction number
- Defaults the media finish mode
- Defaults the logical operators
- Disables the auto macro overlay
- Returns control to PCL5 parsing mode if executed in HP-GL/2 mode
- Executes an HP-GL/2 **IN** command

EscE is a valid HP-GL/2 terminator (see Chapter 16) and has all the functionality of IN, as well as defaulting the:

- Unified Color palette
- Logical page orientation
- Picture frame
- Anchor point
- defaults CAP
- HP-GL/2 plot size

EscE has no effect on I/O and causes no disruption in host-to-peripheral communication. The printer remains on-line and no data is lost.

EscE should be the first command received at the beginning of a job (e.g. before font download) to establish default conditions; and it should be the last command at the end of a job to leave the machine in the user default state and clear any partially composed pages.

EscE moves CAP to the intersection of the top and left margins on the page if no printable data has been sent; otherwise the page is ejected and CAP goes to the intersection of the top and left margins on the next page.

Within an HP-GL/2 label in normal mode ("TD 0;"), *Esc E* is a special case: It causes a device to reset and transition to the PCL5 environment. In transparent data mode ("TD 1;"), *Esc E* does not reset the device.

CAP is ***floating*** prior to printable characters, or commands affecting CAP. (Commands affecting CAP are ASCII data, LF, CAP moves, space, etc.). Commands affecting top and left margins, line spacing, page length, and orientation can change CAP while it is floating.

CAP is ***fixed*** following printable characters or commands affecting CAP. Commands changing the top and left margins or line spacing cannot affect a fixed CAP; but page length or orientation commands eject the page. If no printable data is received, page length and orientation commands eject the page only if the device cannot move the paper back to the top of form. The intent is to not eject unnecessary blank pages.

NOTE: Since control-panel settings can override the factory defaults, *EscE* uses the control panel settings as the default values.

NOTE: The SPACE character is not considered a printable character with respect to CAP unless it is a printable SPACE (i.e. a downloaded SPACE that has a glyph).

HP-GL/2 Mode

Default Values DF

DF; Returns the device to HP-GL/2 default conditions. Factory defaults are restored.

The DF command is equivalent to issuing the following commands:

Command	**Default Value Condition**
AC;	Anchor corner set to lower-left corner of the picture frame relative to the current coordinate system,
AD;	Restores default values to the ALTERNATE font characteristics,
CF;	Solidly fill characters with current pen,
DI;	Absolute character direction parallel to x-axis,
DT;	Label terminator defined to ETX (ASCII decimal value 3) and the mode to 1 (non-printing),
DV;	Right text path with normal line feed,
ES;	Extra space transforms turned off,
FT;	Solid bi-directional area-fill,
IW;	Window defined as printable area (picture frame limits),
LA;	Line Cap = butt, Line Join = mitered, Miter Limit = 5,
LO;	Label origin = 1,
LT;	Solid lines and relative mode,
PA;	Drawing mode absolute,
PM0;PM2;	Polygon buffer cleared,
RF;	Solid black,
SC;	Turn user scaling off,
SD;	Restores default values to the STANDARD font characteristics,
SI;	Turns off size transformation,
SL;	Turns off slant transformation of characters,
SM;	Symbol mode off,
SS;	Select STANDARD font,
TD;	Normal printing mode,
UL;	Defaults all 8 line types.
SB;	Restricts font selection to scalable fonts only.

NOTES:

- The carriage-return point is always updated to the current position when "DF;" is invoked.
- The PD and PU forms of the PA and PR commands are defaulted to be forms of the PA command.
- DF does not affect the position of P1 and P2.
- In addition to returning to solid lines, the pattern length is set to 4% of the diagonal distance from P1 to P2.
- DF does not affect the HP-GL/2 or PCL current active position.
- DF does not affect the currently selected pen.
- DF does not affect the current pen state (up/down).
- DF does not affect the current HP-GL/2 or PCL5 coordinate system orientation.
- DF does not affect the current pen turret (pen widths and pen width unit selection).
- DF clears the current pattern residue and terminates any sequence of continuous vectors (see LA and LT).
- Extension block commands which are defaulted by "DF;" are documented explicitly in the command definitions.

Initialize IN

IN [n];

IN;

n = ic

The "IN n;" command restores factory default conditions to all programmable HP-GL/2 features. The only allowable value for *n* is 1; any other value will be treated like "IN;".

Specifically, the following actions are performed:

"DF;" The default command is executed. See DF for the list of actions.

"PA0,0;" The pen is returned to the lower-left corner of the hard-clip limits.

"PU;" The pen is "raised."

"RO0;" Rotation is canceled.

"IP;" The positions of P1 and P2 are defaulted.

"WU;" Pen width mode is defaulted to metric; units are millimeters.

"PW;" The width of all pens is defaulted.

The "IN;" command defaults to the feature setting(s) selected from the device's control panel, with the remainder of the programmable HP-GL/2 features set to their factory defaults..

NOTE: For The HP Color LaserJet, no HP-GL/2 features can be set from the front panel. Extension blocks may define commands which, due to their special nature, are not defaulted by "IN;". (These exceptions are documented explicitly in the command definitions.)

Printer Job Language (PJL)

Devices may implement a control level language called PJL that can remotely control the front panel, switch between PCL and other printer languages, and perform page accounting and page recovery. PJL that resides logically above the PCL and PostScript language parsers.

PJL Kernel

For a complete description of PJL, see *The Printer Job Language External Reference Specification*. Most devices do not support all PJL functionality; however, all devices using PJL support at least the following commands: *Universal Exit Language/Start of PJL*, *Enter Language*, and *Comment*. These commands are inserted in the data stream between print jobs in the sequence shown below:

Universal Exit Language/Start of PJL

Comment

Enter Language

(Begin PCL, HP-GL/2 or Postscript Print Job)

.

.

(End of Print Job)

Universal Exit Language/Start of PJL

PJL Command Syntax

The following syntax conventions are used in PJL commands.

{}	Items in braces indicate a required parameter.
[]	Items in brackets indicate optional parameters.
<>	Identifies a control code character or special defined identifier.

PJL command syntax uses the special identifiers defined below:

`<HT>`	Horizontal tab character (ASCII 9).
`<LF>`	Linefeed character (ASCII 10).
`<CR>`	Carriage return character (ASCII 13).
`<SP>`	Space character (ASCII 32).
`<WS>`	White space, a result of one or more of either <SP> or <HT>.
`<Words>`	Printable characters (ASCII 33-126) and <WS> starting with a printable character.

Spaces in PJL commands depend on their location in the command: some spacing is required and some is optional. Spacing requirements are shown below:

```
@PJL<WS>Command<WS>Option[<WS>]=[<WS>]Value[<CR>]<LF>
```

OR

```
@PJL<WS>Command<WS><words>[<CR>]<LF>
```

The PJL prefix "@PJL" must always be upper case. The portion of the PJL command following the prefix may be either upper or lower case.

The option field ("Option=Value") identifies the command options (for example, "LANGUAGE=PCL").

The "<words>" field is used only in the *Comment* command.

NOTE: A <LF> is required to terminate a PJL command (except for the *Universal Exit Language/Start of PJL* command). A <CR> may be included with a <LF>. For example, "<CR><LF>" may be used to terminate a command; however, the carriage return is optional and will be ignored.

NOTE: The *Universal Exit Language/Start of PJL* command does not follow the syntax of the other PJL commands.

Background

As originally defined, the HP Color LaserJet printer supports only the kernel PJL commands, ENTER, UEL and COMMENT as did the LaserJet IIIsi. These commands enable a customer's printer driver and/or spooler to switch a HP Color LaserJet from one personality system (e.g. PCL) to another (e.g. PostScript). We refer to this capability as "language switching" or, more properly in the case of HP Color LaserJet, as "system switching."

Since the inception of HP Color LaserJet, certain parallel efforts (Payette and Sawtooth) have been underway to implement the complete PJL feature set, and future efforts (White Clouds, WIN 3.11[1], Compadre and Chariot) will take advantage of many PJL features to administer and manage printers.

HP Color LaserJet did not implement the complete PJL feature set. However, customer advantages are available from a subset of the PJL features that include the following commands:

COMMENT	Ignored.
ECHO	Echoes parameters to the host.
ENTER	Enter language (eg. PostScript, PCL).
EOJ	Delineates the end of a job.
INFO CONFIG	Identifies installed options (eg. memory, trays, etc) in the printer.

[1] Previously known as WIN 3.1A.

INFO ID	Identifies printer model number.
INFO PAGECOUNT	Returns number of pages printed by engine (not by the job!).
INFO USTATUS	Returns the current state, ON/OFF, of USTATUS JOB.
JOB	Delineates the beginning of a job.
RESET	Ignored.
UEL	Universal exit language.
USTATUS DEVICE	Enables/disables unsolicited device status
USTATUS JOB	Enables/disables unsolicited job status.
USTATUSOFF	Disables unsolicited job status.
SET RET	Changes the current (per job) RET setting
INQUIRE RET	Returns the current (per job) RET setting

Avoiding interaction with the control panel is a key feature of this proposed subset which has simplified the testing matrix.

Customer Advantages

Ideally, we would like to present the various dialog boxes that the printer management tools provide, and identify exactly which features of these applications will work and not work for The HP Color LaserJet. However, many of these tools are still being defined and such detailed analysis is not yet possible. Lacking this, we present below a table that identifies important system capabilities that are possible with The HP Color LaserJet (including the proposed PJL enhancements) and related printers. The discerning eye will note that not all these features are wholly implemented using PJL commands.

Feature	**LaserJet IIIsi**	**PaintJet XL300**	**Payette**	**Sawtooth**	**Color LaserJet**
Environment saving	no	no	no	yes	no
Context sensitive lang switching	no	no	yes	yes	yes
True end-of-job notification	no	no	yes	yes	yes
Job boundary recognition	mio	mio,pjl, timer	mio,pjl, timer	mio, pjl, timer	mio,pjl, timer
Automatic I/O switching	no	no	yes	yes	yes
Automatic installation	no	unorthodox	yes	yes	yes
Automatic configuration	no	no	yes	yes	yes
Remote control panel	no	no	yes	yes	no
Remote printer status popups	basic	basic	enhanced	enhanced	basic

Context Sensitive Language Switching

Language switching permits a user or spooler to select a page description language (eg. PCL or PostScript). Language switching may be accomplished using PJL commands (eg. ENTER and UEL), or automatically inferred from the received data in the case of context sensitive switching. The HP Color LaserJet supports both PJL directed and context sensitive language switching.

True End-Of-Job Notification

True end-of-job notification tells a standalone or network user when the printer has actually finished printing their job. Traditional end-of-job notification merely tells the user when the spooling system has transferred the print data into the peripheral which will then continue to work on the job for some time afterwards. The HP Color LaserJet supports true end-of-job notification.

Spoolers implement true end-of-job notification using the PJL JOB and EOJ commands and the PJL USTATUS JOB responses (there are two, one for the START of the job and another for its END).

When enabled with the USTATUS JOB command, The HP Color LaserJet will transmit the USTATUS JOB START and END responses to the host for the JOB and EOJ commands respectively. The HP Color LaserJet supports (and transmits in USTATUS JOB status) the NAME parameter of both JOB and EOJ. However, The HP Color LaserJet does not support nor transmit the optional PAGES parameter of the unsolicited end-of-job status.

The HP Color LaserJet transmits the unsolicited end-of-job status after the last page of the job has been dropped in the output bin.

Job Boundary Recognition

Printers that support the concept of a "job" must recognize where one job ends and another begins. The job concept is the foundation for a number of other features including I/O switching, true end-of-job notification, and network error recovery (the ability to abandon the current PCL job or PostScript program when a network connection fails).

Job boundaries may be recognized using MIO packets (these are important for the network case), the PJL JOB and EOJ commands, or inferred from an in-activity timer.

The HP Color LaserJet employs MIO packets, timers and the most primitive form of the JOB and EOJ PJL commands to implement the same job boundary algorithm as Payette and Sawtooth.

I/O Switching

I/O switching permits the printer to accept data from a different I/O port after it has recognized a job boundary. The HP Color LaserJet supports I/O switching.

Automatic Installation

An application program may select its required printer driver using the Bi-tronics model id command. A variation of this feature will also be implemented for MIO 5.1 network cards.

The HP Color LaserJet includes the Bi-tronics interface and MIO 5.1 to support automatic installation in standalone and network environments.

Automatic Configuration

An application program may query a peripheral to determine what options (e.g. memory, paper trays, etc.) have been installed.

Automatic configuration of the application's driver requires the PJL INFO CONFIG, INFO ID and ECHO commands. The HP Color LaserJet supports these commands.

Remote Control Panel

A remote control panel (similar to Compadre) permits remote administration of a network printer or local administration of a low-end device lacking a local control panel.

Remote control panel applications require the PJL DEFAULT, DINQUIRE, INFO ID, INFO VARS, and ECHO commands. The HP Color LaserJet does not implement the DEFAULT, DINQUIRE nor INFO VARS commands, and hence does not support a remote control panel.

Remote Printer Status Popups

A printer status popup is a message box that appears on your screen to notify you of an important change of your printer's state.

The basic functionality provides only the status messages that have historically been available with the Centronics printer interface, off-line, paper-out and printer-error. MIO peripherals implement the basic functionality using the ASYNC PERIPHERAL STATUS message. The HP Color LaserJet supports the basic functionality.

Enhanced functionality offers some refined messages such as out-of-toner, punt, paper-mount, page-jam, etc. Peripherals implement enhanced remote printer status with the PJL USTATUS DEVICE responses. The HP Color LaserJet does not support the enhanced status popups available with PJL commands.

Supported PJL Commands

The following identifies the PJL commands implemented by The HP Color LaserJet. For more information about these and other PJL commands, see the PJL ERS.

@PJL ENTER

Enter Language

This command enables the specified printer language.

```
@PJL ENTER LANGUAGE = {PCL/HPGL2/POSTSCRIPT}[<CR>]<LF>
```

If the printer does not receive this command, the default language is enabled. This ensures that applications not supporting PJL switching have a higher probability of correctly printing.

NOTE: Entering HP-GL/2 with this command puts the device in "stand-alone" HP-GL/2. For dual context operations, *Esc%#A* and *Esc%#B* should be used to switch between PCL and HP-GL/2 contexts.

The HP Color LaserJet is a system switching peripheral. Switching from one language to another will discard the print environment; switching from one language to the same language will not discard the print environment.

UEL

This command — also referred to as the **UEL** command — terminates the current printer language and returns control to PJL.

Value(#)	=	-12345	Exit to PJL
Default	=	NA	
Range	=	-12345	

This command performs the following actions:

- Prints all data received before this command.
- Performs a reset: *EscE* in PCL, (*Cntrl-D*) in PostScript.
- Turns control over to PJL.

In the PCL and HP-GL/2 contexts, this command is always recognized except when in HP-GL/2 label mode with TD = 1, or during a PCL binary transfer.

This command is also a valid HP-GL/2 terminator.

NOTE: The UEL command must be immediately followed by the "`@PJL`" command prefix. Characters or control codes (e.g., `CR` or `LF`) other than `@PJL` enable the default language and process the job in that language.

NOTE: All jobs should start and end with the UEL command. Besides entering PJL, the UEL command has the same effect as the *EscE* command. However, the *EscE* command should always be included to ensure backward compatibility.

DEVICE NOTE: Current DJs do not support PJL. They ignore the UEL command and cannot interpret the subsequent @PJL ENTER LANGUAGE command.

@PJL Comment

This command designates the current line as a comment, which is ignored.

```
@PJL COMMENT <Words>[<CR>]<LF>
```

@PJL INFO CONFIG

The HP Color LaserJet's solicited response will describe the configuration of the installed options in the printer. The purpose of including this command in The HP Color LaserJet is to allow application programs to determine the installed options in a printer during the course of automatic configuration. The HP Color LaserJet's response will be similar to:

```
@PJL INFO CONFIG
IN TRAYS [3 ENUMERATED]
    INTRAY1 UT
    INTRAY2 MF
    INTRAY3 LT
OUT TRAYS [2 ENUMERATED]
    NORMAL FACEDOWN
    LOWER FACEUP
PAPERS [6 ENUMERATED]
    LETTER
    LEGAL
    A4
    EXECUTIVE
    TABLOID
    A3
LANGUAGES [2 ENUMERATED]
    PCL
    POSTSCRIPT
USTATUS [1 ENUMERATED]
    JOB
    DEVICE
FONT CARTRIDGE SLOTS [2 ENUMERATED]
    UPPER=INSTALLED/EMPTY
    LOWER=INSTALLED/EMPTY
MEMORY=8388608
DISPLAY LINES=2
DISPLAY CHARACTER SIZE=16
SERIAL NUMBER="496604618"
```

@PJL INFO ID

The purpose of including this command in The HP Color LaserJet is to support application programs that use it instead of or in addition to the Bi-tronics model id to determine the type of printer when selecting a printer driver during automatic installation. The HP Color LaserJet responds similar to (the final message will be chosen later):

```
@PJL INFO ID <CR><LF>
"Hewlett-Packard Color LaserJet" <CR><LF><FF>
```

@PJL INFO USTATUS

This command queries the state of USTATUS. **The HP Color LaserJet supports only unsolicited JOB status; PAGE, DEVICE and TIMED are not supported** nor included in the returned message. The HP Color LaserJet responds with:

```
@PJL INFO USTATUS <CR><LF>
JOB=OFF [2 ENUMERATED] <CR><LF>
    <HT>ON <CR><LF>
    <HT>OFF <CR><LF><FF>
```

@PJL INFO PAGECOUNT

This command returns the number of pages printed by the engine.

```
@PJL INFO PAGECOUNT<CR><LF>
PAGECOUNT=30556<CR><LF>
<FF>
```

@PJL INFO *unimplemented*

The HP Color LaserJet will respond to an INFO command for an unimplemented variable per the PJL ERS. In particular, **The HP Color LaserJet will not implement INFO IOCONFIG,, INFO STATUS, INFO ERRORLOG and INFO VARIABLES.**

```
@PJL INFO <unimplemented variable> <CR><LF>
"?" <CR><LF><FF>
```

@PJL JOB

The purpose of including this command in the HP Color LaserJet is to bring our I/O switching algorithm in-line with Payette and Sawtooth. The HP Color LaserJet counts nested jobs, incrementing the counter for @PJL JOB and decrementing it for @PJL EOJ.

The HP Color LaserJet ignores the optional START=, END=, OFFSET= and PASSWORD= parameters. They may be included, but they will have no effect.

The HP Color LaserJet accepts the optional NAME= parameter and returns the name string in the unsolicited JOB start status message (if enabled). The HP Color LaserJet transmits the unsolicited JOB status message to every I/O channel which has enabled this function. This message's format is:

```
@PJL USTATUS JOB <CR><LF>
START <CR><LF>
{NAME=<Job Name> <CR><LF>} <FF>
```

The HP Color LaserJet resets the nested job counter whenever it switches the source to a different I/O channel. Thus a corrupt job on one channel will not disrupt the activities of another channel.

@PJL EOJ

This command delineates the end of a job and is a hint for the I/O switching algorithm. The HP Color LaserJet counts nested jobs, incrementing the counter for each JOB, decrementing it for each EOJ, and recognizing a job boundary when the counter decrements to zero. The HP Color LaserJet will ignore an isolated EOJ command that has not been preceded by a JOB command.

The HP Color LaserJet accepts the optional NAME= parameter and returns the name string in the USTATUS JOB END status message (if enabled) as follows. The HP Color LaserJet transmits the unsolicited JOB status message to every I/O channel which has enabled this function.

```
@PJL USTATUS JOB <CR><LF>
END <CR><LF>
{NAME=<Job Name> <CR><LF>} <FF>
```

The EOJ command will push an open (incomplete) page out of the printer.

@PJL ECHO

The HP Color LaserJet supports the ECHO command which transmits its parameters to the I/O channel (to the host) which delivers the command.

@PJL USTATUS JOB=ON/OFF

This command enables or disables unsolicited JOB status (eg. USTATUS JOB messages sent to the host) for the I/O channel that delivers the command.

Note: The HP Color LaserJet supports only unsolicited JOB status; no other unsolicited status messages are supported.

@PJL USTATUS DEVICE=ON/OFF

This command enables or disables unsolicited DEVICE status (eg. USTATUS DEVICE message sent to host) for the I/O channel that delivers the command.

@PJL USTATUSOFF

This command disables unsolicited JOB status for the I/O channel that delivers the command. For The HP Color LaserJet, it duplicates the functionality of @PJL USTATUS JOB=OFF.

@PJL RESET

The HP Color LaserJet accepts this command, but does not actually reset anything since The HP Color LaserJet does not support any of the variables which are supposed to be reset.

Unimplemented PJL Commands

The HP Color LaserJet ignores unsupported PJL commands (eg. INITIALIZE, SET, DEFAULT, INQUIRE, DINQUIRE, OPMSG, STMSG and RDYMSG) and resumes parsing with the next line unless the FORMAT:BINARY command modifier is present in which case the entire job upto the UEL is discarded.

Interactions

This section documents how certain PJL functions interact with other features of the HP Color LaserJet printer. The complete set of interactions will be published in the PJL Feature Analysis documents at a later time.

Front Panel Cold Reset

A front panel cold reset will disable unsolicited status on all I/O channels when the reset takes effect.

Front Panel Reset

A front panel reset will disable unsolicited status on all I/O channels when the reset takes effect.

ESC-E

An Esc-E will neither enable nor disable unsolicited status.

ESC-Z

An Esc-Z will neither enable nor disable unsolicited status.

UEL

The universal exit language command will neither enable nor disable unsolicited status.

A UEL will never cause the printer to transmit unsolicited JOB status.

A UEL will not discard the PCL print environment (eg. downloaded soft fonts, palettes and so on).

Lost Network Connection

When an MIO card indicates that the network connection on an I/O channel has failed, the peripheral will disable unsolicited status on that I/O channel.

Deadlock Timer

The HP Color LaserJet will neither enable nor disable unsolicited status when the PJL Deadlock Timer expires. However, a job boundary may be recognized and a new job begun on a different channel. The HP Color LaserJet will reset the nested job counter when the Deadlock Timer expires.

I/O Timers

The HP Color LaserJet will neither enable nor disable unsolicited status when the I/O Timer expires. However, a job boundary may be recognized and a new job begun on a different channel. The HP Color LaserJet will reset the nested job counter when the I/O Timer expires.

Print Environment

PJL commands will not cause The HP Color LaserJet to discard the modified PCL print environment unless a language switch occurs. For example, you may download soft fonts, palettes and patterns in one PCL job and refer to them during subsequent PCL jobs where each job is separated from another with a block of PJL commands that does not include an ENTER command for another language.

Printer Diagnostics

Self-test *Esc z*

Initiates printer self-test, which instructs the printer to perform the following actions:

- Stop processing page and print all data preceding the self-test.
- Perform reset (ejecting the page if printable data has been sent.) Downloaded fonts are not deleted.
- Move CAP to the top of form, if not already there.
- Perform the self-test.
- After self-test, move to the top of form, if not already there.
- Resume execution of the user data without data loss (programmable features need not be saved).
- Print results and currently installed fonts.

This is as complete a self-test as The HP Color LaserJet is capable of performing in the allotted time. If no error is detected, the printer will remain on-line. If an error is detected, the printer will go to the off-line state.

Chapter 6:   PostScript Operators

PostScript Operators

This chapter describes the implementation specific details of the external interfaces to the Level 2 PostScript interpreter in the HP Color LaserJet PostScipt simm. A copy of the PostScript Language Reference Manual Second Edition (PLRM) is an essential complement to this document. Although some parts of the PLRM have necessarily been incorporated here, this is a description of what is specific to our implementation and thus is not in the PLRM.

The first sections describe our implementations of features unique to Level 2. The remaining sections describe how Level 1 compatibility "operators" will behave. In our previous Level 1 PostScript cartridge product, these were operators in the statusdict dictionary which were unique to our implementation. In the current Level 2 implementation, these are procedures in statusdict which invoke Level 2 mechanisms to emulate the behavior of the operators in the Level 1 environment.

Level 2 PostScript introduced several operators that take dictionaries as arguments and return dictionaries as results. The key-value pairs in these dictionaries are referred to as parameters because their values typically select optional features or control the operation of some part of the PostScript implementation. The use of dictionaries as containers for parameters provides an extensible method of adding support for new features by adding a new parameter key to the appropriate dictionary. This approach avoids adding new operators to the language on a per-feature basis, thereby maintaining the device independence of the PostScript language.

Many Level 2 key/value pairs have names that are similar to the Level 1 operator that they interact with. For example, **DoAutoContinue** is a key/value pair in the Level 2 System Parameters dictionary; **doautocontinue** is a Level 1 compatibility operator. In general, key/value pairs use this spelling convention; similar but not identical to the Level 1 compatibility operator that is being emulated.

In this section we will describe the format for presentation of the Level 2 PostScript Language Parameters and their operators as well as the Level 1 compatibility "operators." A tree diagram illustrating the structure of the Level 2 Language Parameter categories is included.

Operator Descriptions

The following documents HP's implementations of PostScript Level 2 Parameter Sets/Operators as well as the behavior of HP supported Level 1 PostScript operators in the Level 2 environment. Each operator subsection will be documented in the following manner:

Operator Name

- Operator Usage -

Description (EXAMPLES are included in the Level 2 chapters)

DEFAULT:

ERRORS:

PARAMETER TABLES (Level 2 chapters only)

The operators are grouped by functionality.

Here is an example of an Level 1 Operator Subsection:

margins

- margin top left

returns the two integers in the array in the Margins value in the current page device.

DEFAULT: top = 0, left = 0

ERRORS: stackoverflow

Here are descriptions of each type of entry in an Operator Subsection:

Operater Name

If a single asterisk (*) appears following the operator name, then the operator modifies a persistent parameter and changes NVRAM. In order to successfully execute the operator, the PostScript interpreter must be executing an unencapsulated job (unless otherwise noted). These operators remain in effect through power cycles, language switching, and control panel resets. If a double asterisk (**) appears following the operator name, then the operator applies in a semi-persistent manner. If the operator is executed as part of an encapsulated job, the operator will apply to the current job only. If the operator is executed as part of an unencapsulated job, then the operator will apply in a semi-persistent manner. Semi-persistent operators remain in effect as long as the printer remains in PostScript and powered up. System default values will be reinstated when the printer is powered back up. There are very few semi-persistent operators in this implementation.

Operators with no asterisks will make changes that remain in effect ONLY through the current print job, if in fact the operator causes settings changes at all.

Operator Usage

The second line in an operator subsection describes the usage of the operator. The general format for operator usage is:

operand$_1$.. *operand*$_n$ **operator** *result*$_1$.. *result*$_m$

where *operand*$_1$ through *operand*$_n$ are the operands that the **operator** requires, with *operand*$_n$ being the topmost element on the operand stack, **operator** is the name of the operator, and *result*$_1$ through *result*$_m$ are the results left on the operand stack after the successful execution of the **operator** with *result*$_m$ being the topmost element. The notation '-' in the operand position indicates that the operator does not consume any objects off of the operand stack. The notation '-' in the result position indicates the operator does not leave any results on the operand stack.

Description

The third part of an operator sub-section is the operator description. A paragraph describes function and limitations of the operator. Examples illustrating usage will be included for the chapters dealing with Level 2. The examples are drawn from an interactive session with the PostScript executive. See the PostScript Language Reference Manual pp. 22, 79 and 412 for more information.

DEFAULT:

Some parameters that can be read by the operator have a default value that is used if no operator has been executed that changed the default value. If applicable, the default value is listed in the default part of the operator subsection. This may also appear in the form of a table.

ERRORS:

Executing an operator can result in a PostScript error. There are 28 different PostScript errors in this implementation. The PostScript Errors chapter of this document describe each error in detail. Refer to the PostScript Language Reference Manual pp.64, 99-101, and 408 for more information.

PARAMETER TABLES

Parameter Tables are appropriate for the chapters dealing with Level 2 (Chapters 2, 3, 4, and 5) Each parameter or key is presented with its value type, its default value and a word about its purpose. Supporting information is included where appropriate.

For each table, there is a value type field. Here is some additional information about the different value types possible for the parameters described:

volatile - New value is retained only for the duration of the job.

semi-persistent - New value is retained across job boundaries if set in an unencapsulated job; otherwise it is volatile.

persistent - New value is retained across job boundaries and power cycles if set in an unencapsulated job; otherwise, it is volatile.

read only - Value is not changeable via PostScript but may be changed via the control panel.

write only - Value may be written but not read (passwords only).

constant - Value never changes.

persistent/read only - Value changes by internal mechanisms only and is saved

PostScript Paper Handling

The HP Color LaserJet supports the following three models for media selection during the printing process:

1. HP-Specific Level 2 Paper Handling Model
2. Adobe-Standard Level 2 Paper Handling Model
3. Level 1 Paper Handling Model

Models 2 and 3 are discussed in the Red Book. The HP Color LaserJet HP-Specific Paper Handling algorithm is as follows:

```
IF   requested PageSize is supported by the printer AND requested tray contains
     media of requested PageSize
THEN  print from the requested tray
Otherwise...
     IF   tray=both AND  requested PageSize is available in the alternate tray
     THEN  print from matching tray selected by the printer's firmware
     Otherwise...
          prompt the user to mount the proper size in the requested tray
          IF   online key is pressed AND other tray contains the requested
               PageSize AND tray=other at the Front Panel
          THEN  print from the other tray
```

Level 2 Page Device Parameters

One category of PostScript Language Parameters for Level 2 is known as the Page Device Parameters. The Page Device Parameter sets are modified by the setpagedevice operator and returned by the currentpagedevice operator. The Page Device Parameters are concerned with printing capabilities in general. For example, how input media should be selected, what choices for image sizes are available, how pages should be processed/output and how exceptions/errors should be handled. First, the operators will be discussed; then the Parameter Sets will be described in tables. These tables have been organized into three broad categories (Input Media, Processing and Output, and Policy and Special Action) although there is only ONE dictionary containing all of the Page Device Parameters

setpagedevice*

dict setpagedevice -

where dict is one or more of the key/value pairs described in the tables in this chapter.

A new raster output device in the graphics state is installed based on a combination of the information in the current device and the information found in the dictionary operand. The dictionary is a container for communicating requests as key/value pairs, which are logically passed by value and copied by the setpagedevice machinery into internal storage.

Calls to **setpagedevice** are cumulative: information established in a previous call to **setpagedevice** will persist unless overridden explicitly in a subsequent call. Therefore, there are no required keys in any particular call. This behavior applies not only to the top-level dictionary, but also to the sub-dictionary Policies. An exception is the **InputAttributes** dictionary. It is necessary to specify all of the entries for a given tray in order to change any one entry.

It is required that the PageSize entry for the tray be present or a rangecheck error will occur when writing the **InputAttributes** dictionary. .

See the example immediately following on changing the **InputAttributes** dictionary.

See the PostScript Language Reference Manual pp. 226-255, 390, and 512 for additional information.

EXAMPLES:

1.) A page description might request an executive image PageSize in this way which would initiate a search for a tray matching the requested PageSize. If no tray matched the request, a policy would be invoked according to the PageSize value in the Policies Dictionary:

```
<</PageSize [522 756] >> setpagedevice
```

2.) A program might semi-persistently change the contents of the **InputAttributes** dictionary in the following manner. This example presupposes that a currentpagedevice call has already returned the PageSize and MediaType values for each tray (or the printer has been inspected to determine which trays are fitted). The PageSize and MediaType values are then used to generate this sequence. Note that this requires a call to startjob to alter initial Virtual Memory (VM) although no password is active:

```
true 0 startjob not                             % exit encapsulated job state
<</InputAttributes                              % access InputAttributes
<< 0 <</PageSize [612 792] /MediaColor          % access Tray 0 and modify Color,
(blue)                                          Weight, Type of media
/MediaWeight 20 /MediaType                      % completes statement to access
(transparency) >>                               InputAttributes
/Priority [1 0]                                 % also modify the Priority array to
                                                make Tray 1 default.
>>>> setpagedevice
false 0 startjob not                             % return to encapsulated job state;
                                                changes take effect
```

ERRORS: configurationerror, typecheck, rangecheck, stackunderflow, limitcheck, invalidaccess

currentpagedevice

- currentpagedevice *dict*

would return a read only dictionary containing the key/value pairs of the pagedevice dictionary described in the following tables. This read only dictionary describes the page-oriented output device in the current graphics state. currentpagedevice creates a new dictionary if necessary. If the device in the current graphics state is not a page device, currentpagedevice returns an empty dictionary. See the PostScript Language Reference Manual pp. 229, 390 for additional information.

EXAMPLE:

To see the contents of the InputAttributes dictionary:

```
currentpagedevice /InputAttributes get          % puts a copy of the InputAttributes
                                                dictionary on the stack
{} forall pstack                                % pops all of the keys off of the
                                                stack preceded by their values and
                                                prints them out
```

ERRORS: stackoverflow, VMerror

NOTE: that the dictionary returned by **currentpagedevice** has the following subdictionaries in addition to its top level entries:

InputAttributes dictionary

Policies dictionary

PostRenderingEnhanceDetails dictionary

NOTE: also that the **InputAttributes** dictionary has the following subdictionaries in addition to its entries. These dictionaries correspond to the trays fitted to the printer:

0 dictionary

1 dictionary

Dictionary 0 is always present. Dictionary 1 is present if the optional lower tray is installed.

The following Page Device Parameters are implemented in this product. A brief description is included but for additional information the reader is referred to the PostScript Language Reference Manual Section 4.11, the PostScript Language Supplement (Chapter 2) and the PostScript Language Addendum for Hewlett-Packard LaserJet III Family (Chapter 1). Any parameters described in the PostScript Language Reference Manual or Supplement that are not included here are not supported in this product:

MatchAll - If this key is present in a **InputAttributes** entry and it's value is true, a **setpagedevice** request will match that entry only if it specifies matching (non-null) values for all the attributes present in the entry. If **MatchAll** is false or absent, a **setpagedevice** request will match the entry if it specifies any subset of the entry's attributes and leaves the others null. See PostScript Language Reference Manual page 243. **NOTE:** If the **TraySwitch** key is true, the printer will switch trays in mid-job to a tray that doesn't meet the **MatchAll** requirement (even if **MatchAll** is true). While this is not correct, it matches what all previous HP products have done.

The "default" for MatchAll is that it is NOT present (that is undefined).

PAGE DEVICE - INPUT MEDIA SELECTION

KEY	VALUE TYPE	DEFAULT VALUE	SEMANTICS
PageSize	array [x y] (semi-persistent)	Installed tray size in highest priority tray.	currentpagedevice entry specifying size of page requested. Default PageSize of [x = width y = height] installed at start of job based on the first element in the Priority array.
MediaColor	string or null (semi-persistent)	null	currentpagedevice entry specifying media color requested.
MediaWeight	number or null null (semi-persistent)	null	currentpagedevice entry specifying media weight requested.
MediaType	string or null (semi-persistent)	Based on control panel setting for highest priority tray	currentpagedevice entry specifying media type requested.
ManualFeed	boolean (persistent)	false	If true, media is drawn from the manual feed position. Also see manualfeed in Level 1 Tray selection operators section.

TRAY 0

InputAttributes dictionary - This table describes the Parameters for Tray 0 within the **InputAttributes** subdictionary

0	**dictionary** (read only)		**Upper Tray**
PageSize	array [x y] (read only)	Installed tray size.	Specifies size of media in this tray. Determined by tray sensor hardware and updated at the beginning of each job.
MediaColor	string or null (semi-persistent)	null	Specifies color of the media in this tray.
Media Weight	number or null (semi-persistent)	null	Specifies weight of the media in this tray.
MediaType	string or null (read only)	Based on control panel setting	String to specify the type of the media.null = plane paper (transparency) = transparency

Tray 1

InputAttributes dictionary - This table describes the Parameters for Tray 1 within the **InputAttributes** subdictionary

1	dictionary (read only)		Lower Tray
PageSize	array [x y] (read only)	Installed tray size.	Specifies size of media in this tray. Determined by tray sensor hardware and updated at the beginning of each job.
MediaColor	string or null (semi-persistent)	null	Specifies color of the media in this tray.
Media Weight	number or null (semi-persistent)	null	Specifies weight of the media in this tray.
MediaType	string or null (semi-persistentread only)	Based on control panel setting	String to specify the type of the media. null = plane paper (transparency) = transparency

InputAttributes dictionary - The Priority array is also contained within the **InputAttributes** subdictionary

Priority	array[x y] (persistent)	[0 1]	Priority can be set persistently for [0 1] or [1 0].

Page Device - Processing and Output

KEY	VALUE TYPE	DEFAULT VALUE	SEMANTICS
ImagingBBox	null (constant)	null	Optional page bounding box which is described as a four element array of integers.
NumCopies	integer or null (persistent)	null	Default number of copies.
OutputFaceUp	boolean (semi-persistent)	false	Selects output bin for paper delivery.
HWResolution	array [x y] (constant)	[300 300]	Always = [300 300] for this printer.
OutputPage	boolean (semi-persistent)	true	Indicates whether a page should be output.

Page Device - Policy.

Key	Value Type	Default Value	Semantics
Policies	**dictionary** (read only)		Contains feature-policy key-value pairs that specify what setpagedevice should do when a feature request cannot be satisfied.
PolicyNotFound	integer (semi-persistent)	1	0 = Generate a **configurationerror** 1 = Ignore the feature request 2 = Interact with a human (Not implemented in this product)
PageSize	integer (semi-persistent)	0	0 = Generate a **configurationerror** 1 = Ignore the requested PageSize 2 = Interact with a human (Not implemented in this product) 3 = Select the nearest available medium and adjust the page to fit 4 = Select the next larger medium and adjust the page to fit 5 = Select the nearest sized available medium but do not adjust the page to fit 6 = Select the next larger medium but do not adjust the page to fit 7 = Behave as in a Level 1 device. IE: print the image starting in the upper left hand corner of the page. Do not change from the specified paper tray no matter what. This is not implemented in this device! NOTE: THIS IS AN UNDOCUMENTED POLICY AND IS FOR USE ONLY THROUGH THE COMPATIBILITY OPERATORS.
PolicyReport	procedure (semi-persistent)	{pop}	Called upon successful completion of **setpagedevice** if it is needed to consult policies in order to handle one or more unsatisfied feature requests.

any feature name	integer (semi-persistent)	1	OPTIONAL: See note following table. 0 = Generate a **configurationerror** 1 = Ignore the feature request 2 = Interact with a human (Not implemented in this product)

Page Device - Special Action

Key	Value Type	Default Value	Semantics
Install	procedure (semi-persistent)	See note following table,	Executed to install values in the graphics state during each invocation of setpagedevice.
BeginPage	procedure (semi-persistent)	{pop}	This procedure is executed at the beginning of each page. This can be any PostScript procedure.
EndPage	procedure (semi-persistent),	{exch pop 2 ne}	This procedure is executed at the end of each page. This can be any PostScript procedure.
PostRendering Enhance	boolean (persistent)	true	Presence of this key indicates the product supports any type of Post Rendering Enhancement feature. The value indicates whether it is on or off. This key is implementation independent.
ExitJamRecovery	boolean (persistent),	false	Indicates whether Jam Recovery on the Exit path is active. A performance degradation may result when printing single paged jobs if this feature is set true. Entry Jam Recovery is always active.
ManualFeedTime out	integer (persistent),	60	Indicates the number of seconds the printer waits for a job to be manually fed before canceling the job. Only active if the ManualFeed boolean is true.
TraySwitch	boolean (persistent)	false	Specifies whether automatic tray switching is active when paper out conditions occur.

DeferredMediaSelection	boolean (semi-persistent)	false	Specifies whether or not to "bypass" the PostScript selection model and allow pagesize, etc. to be "satisfied" at print time.

Page Device - PostRendering Enhance Details.

Key	Value Type	Default Value	Semantics
PostRendering Enhance Details	dictionary (read only)		Contains details about HP's specific resolution enhancement features.
Type	integer (constant)	8	Unique ID to describe the class of Post Rendering Enhancement available in this product.
REValue	integer (persistent)	2	Specifies the current value of Resolution Enhancement: 0 - Off, 1 - Light, 2 - Medium, 3 - Dark

MISCELLANEOUS NOTES

Any unimplemented Page Device entries (examples: **MirrorPrint**, **Jog**, **Collate**, etc.) will be handled based on the value of the **PolicyNotFound** key.

any feature name specifies the policy to use when a specific named feature (other than PageSize) cannot be satisfied. Any key that can appear in a dictionary supplied to **setpagedevice** may also be used as a key in the **Policies** dictionary. This is NOT limited to keys recognized by this implementation but may include any key. That is, **setpagedevice** consults **Policies** the same way for an unknown feature as it does for a known feature whose requested value cannot be satisfied.

Policies can be set on an individual feature basis such as in the following example:

```
<</Duplex true /Collate true /Policies << /Duplex 0 /Collate 1 >> >> setpagedevice
```

Multiple Copies - NumCopies

Additional information is included here regarding the **NumCopies** key found in the preceding PROCESSING AND OUTPUT table. Since there are three separate mechanisms involved with determining the number of copies to be output, each will be referred to in the following manner:

NumCopies	= Level 2 Page Device key
#copies	= Level 1 entry now in Systemdict
COPIES	= Control panel setting

If the **NumCopies** entry in the page device is not null, it specifies the number of copies of each page of the document to produce. A *null* value indicates that **showpage** and **copypage** should consult the value of #copies in the current dictionary stack each time they are executed. The number of copies can also be specified from the control panel with COPIES. At the beginning of each job, if the current value of **NumCopies** is not null, the control panel value is copied into **NumCopies**. If the value of **NumCopies** is null, the control panel value of COPIES is copied into #copies. If the **NumCopies** entry is set persistently in an unencapsulated job, the value is copied back to the control panel value of COPIES at the end of the job. Setting #copies in an unencapsulated job is semi-persistent and will not affect the **NumCopies** entry or the control panel setting of COPIES. In an initialized printer, the default control panel value of COPIES is 1, the value of **NumCopies** is *null* and the value of #copies is 1.

Resolution Enhancement - PostRenderingEnhanceDetails

Additional information is included here regarding the **PostRenderingEnhanceDetails** dictionary found in the preceding POLICY AND SPECIAL ACTION table. A general mechanism for supporting resolution enhancement features is provided in Level 2 PostScript using the **PostRenderingEnhance** entry in the **pagedevice**. When this Boolean entry is set to true, the **PostRenderingEnhanceDetails** entry then specifies the type of feature provided. The **PostRenderingEnhanceDetails** dictionary contains a **Type** entry, which is a read-only id assigned specifically for this type of Resolution Enhancement, and (**REValue**), a product specific entry specifying the current value of RET. ,   Legal values for **REValue** are: 0, 1, 2, or 3. At the beginning of a job, if the desired value of Resolution Enhancement is nonzero, the **PostRenderingEnhance** key is set to true and the **PostRenderingEnhanceDetails** dictionary is set to <</Type 8 /REValue x >>, where x is the Resolution Enhancement value. If the desired Resolution Enhancement value is zero, the **PostRenderingEnhance** key is set to false and the **PostRenderingEnhanceDetails** dictionary is set to <</Type 8 /REtValue 0 >>.

Level 2 I/O Device Parameters

I/O Device parameters are one class of Level 2 parameters to control the operation and behavior of the PostScript Interpreter. These are referred to in the PostScript Language Reference Manual as device parameters but are entirely concerned with I/O devices and I/O channel parameters. The I/O Device Parameters sets are modified by the **setdevparams** operator and returned by the currentdevparams operator. First the operators will be discussed; then the I/O Device Parameter dictionaries containing the Parameter Sets for each device will be described. Typically, I/O Device parameters are similar to system parameters in that they require a password, are global to the PostScript environment, and have similar persistence characteristics. As with system parameters, some of these parameters are stored in NVRAM. One way in which I/O parameters differ from system or user parameters is that they are interdependent; that is, the setting of a given value for a parameter may restrict the ability to set or change another parameter. In this implementation with the factory installed I/O card, the RS-232 serial channel and the parallel channel are always active and cannot be altered through the data stream. No dictionaries exist for these channels. The third channel can be selected at the control panel to be either LocalTalk or RS-422 serial. In this implementation, all I/O parameters are read only except for **LocalTalkType** in the LocalTalk device which can be written via the PostScript interpreter. Any attempt to write read-only values is ignored. The value of **LocalTalkType** is semi-persistent if set in an unencapsulated job. It cannot be set persistently and does not require a password. Also see the **appletalktype** Level 1 compatibility operator for more information.

When using the factory installed I/O card, selection of I/O channel parameters can only be made from the control panel. For most of the optional I/O cards all I/O Device Parameters are determined by hardware or firmware design and cannot be changed. It has been determined that few PostScript applications, drivers, or utilities make use of operators that change I/O configurations so the impact of this incompatibility with other PostScript products should be minimal. See the setsccbatch Level 1 compatibility operator for more information. The reader is also referred to the PostScript Language Reference Manual pp. 384, 501, 571, and 579 for more general information on this section.

setdevparams**

string dict setdevparams

attempts to set one or more parameters for the device identified by string according to keys and new values contained in the dict operand. The string identified is a storage or I/O device (See PostScript Language Reference Manual pp. 75-77 on "Named Files"). The dictionary is a container for key-value pairs; **setdevparams** reads the information from the dictionary but does not retain the dictionary itself. Device parameters whose keys are not mentioned in dict are left unchanged. If a parameter name is not known to the implementation, an undefined error occurs. In general, *string* refers to some input/output or storage device (%Device%) and dict contains one or more key/value pairs constituting the parameter set for the device.

EXAMPLE:

One may wish to hide the presence of a printer on a LocalTalk network and so might modify the value of **LocalTalkType**. This example is appropriate for the case when the printer has been set up with an active password:

```
(%LocalTalk%)                  % indicate string specifying device type
<</LocalTalkType               % specify new value for LocalTalkType
(my_laser_type)
/Password (system_password)    % include password
>>setdevparams
```

ERRORS: configurationerror, invalidaccess, rangecheck, stackunderflow, typecheck, undefined

currentdevparams

string currentdevparams *dict*

returns a dictionary containing the keys and current values of all parameters for the device identified by string. The returned dictionary is merely a container for key-value pairs. Each execution of **currentdevparams** allocates and returns a new dictionary. (%Device%) **currentdevparams** dict where (%Device%) could be (%LocalTalk%), for example, would return a newly allocated special dictionary containing the key/value pairs for the LocalTalk parameter set described below.

EXAMPLE:

```
(%LocalTalk%)               % puts a copy of the dictionary containing the
currentdevparams            LocalTalk Parameter Sets on the stack
{} forall pstack            % pops all of the keys of off the stack preceded by
                            their values
                            % and returns them to the host for display.
```

ERRORS: stackoverflow, VMerror

PARAMETER TABLES:

The following I/O Device Parameters are implemented in this product. A brief description is included but for additional information the reader is referred to the PostScript Language Supplement (Section 3.3) and the PostScript Language Addendum for Hewlett-Packard LaserJet III Family (Section 2.3). Any Parameters described in those documents that are not included here are not supported in this product:

I/O DEVICE - LOCALTALK

There will be one set of I/O parameters, called (%LocalTalk%), containing the following key-value pairs:

Key	Value Type	Default Value	Semantics
LocalTalkType**	string (read/write)	LaserWriter	Defines the TYPE field of the printer's. AppleTalk name. Maximum size of 31 characters plus null termination. This is the ONLY writeable I/O Device Parameter.
HasNames	boolean (read only)	false	Always false. In other implementations this is used to indicate whether this is a named device (such as a disk) capable of containing "Named Files"
Type	name (read only)	Communications	Always = Communications

I/O DEVICE - ETHERTALK

There will be a second set of I/O parameters, called (%EtherTalk%), containing the following key-value pairs:

Key	Value Type	Default Value	Semantics
EtherTalkType**	string (read/write)	LaserWriter	Defines the TYPE field of the printer's. AppleTalk name. Maximum size of 31 characters plus null termination. This is the ONLY writeable I/O Device Parameter.
HasNames	boolean (read only)	false	Always false. In other implementations this is used to indicate whether this is a named device (such as a disk) capable of containing "Named Files"
Type	name (read only)	Communications	Always = Communications
EtherTalkZone**	string (read/write)	(*)	

The EtherTalkType and the LocalTalkType should "map to the same variable". In other words, if EtherTalkType changes then LocalTalkType will change to the same value and vice-versa.

Level 2 System Parameters

One category of PostScript Language Parameters for Level 2 to control the operation and behavior of the PostScript Interpreter is known as the System Parameters. System Parameters are concerned with the overall default configuration of the printer. Many of the Parameters are concerned with the allocation of resources such as memory. Determination of the global behavior of the printer with respects to such areas as Passwords, error handling and timeouts are found in these Parameter Sets.

The **setsystemparams** and **currentsystemparams** operators manipulate the System Parameter Sets described in the tables to be found in this chapter. The operators will be described first and then the tables will be presented.

It is important to realize that alterations to these parameters have a permanent, system wide effect which may persist through restarts of the PostScript interpreter as well as power down and restart of the printer. Persistent parameters that are stored in non-volatile memory (NVRAM) are indicated in the Value field of the following tables. System Parameters can be altered only by a program that presents a valid SystemParamsPassword string. The **SystemParamsPassword** is itself a System Parameter (write only). A user application should never attempt to alter system parameters. Only a system management program should be used to alter these parameters. The parameters **DoPrintErrors**, **DoStartPage** and **DoAutocontinue** are the only System Parameters that can also be changed via the control panel.

setsystemparams*

dict setsystemparams -

attempts to set one or more system parameters whose keys and new values are contained in the dict operand. The dictionary is merely a container for key-value pairs; setsystemparams reads the information from the dictionary, but does not retain the dictionary itself. System parameters whose keys are not referenced are left unchanged. The dictionary must contain an entry named Password whose value is the system parameter password (a string or integer). If the password is incorrect, **setsystemparams** executes an invalidaccess error and does not alter any parameters. The Password entry is not required if the **SystemParamsPassword** is the default empty string (). See the PostScript Language Reference Manual pp. 516, 571, 574 for more information.

EXAMPLES:

1.) Set up an active password for the System Parameters when the current password is the empty string ()as follows:

```
<< /Password () /SystemParamsPassword (system_password) >> setsystemparams
```

2.) A system management program might wish to persistently turn off the Startup Page, turn on Autocontinue, turn off the printing of error messages and set the host I/O wait timeout to 600 seconds. The arguments to accomplish this are as follows:

```
<< /DoStartPage false /DoAutoContinue true      % any number of Parameters can be changed
/DoPrintErrors false /WaitTimeout 600           % with no affect on other System Parameters
```

```
  /Password (system_password)>>                    % note password string is now required.
  setsystemparams
```

3.) Set up a StartJob password to protect the settings in the Page Device Parameter Set (discussed in chapter 2)

```
<</Password (system_password) /StartJobPassword (job_password) >> setsystemparams
```

ERRORS: invalidaccess, stackunderflow, typecheck,rangecheck (timeouts)

currentsystemparams

- currentsystemparams *dict*

returns a dictionary containing the keys and current values of all readable system parameters that are defined in the implementation. The returned dictionary is merely a container for key-value pairs. Each execution of currentsystemparams allocates and returns a new dictionary. Note that passwords (**SystemParamsPassword** and **StartJobPassword**) are write only and so are NOT returned in the dictionary accessible by this operator. See the PostScript Language Reference Manual pp. 392, 571, 575 for more information.

EXAMPLE:

```
  <<currentsystemparams /PrinterName get        % puts a copy of the PrinterName key/value
                                                pair on the stack
  ==                                            % pop the value off of the stack and display it
  (HP LaserJet myname)                          % current value of PrinterName
```

ERRORS: stackoverflow, VMerror

PARAMETER TABLES

The following System Parameters from Appendix C of the PostScript Language Reference Manual are implemented in this product. A brief description is included but for additional information the reader is referred to the PostScript Language Reference Manual pp. 571-579, the PostScript Language Supplement (Section 3.2) and the PostScript Language Addendum for Hewlett-Packard LaserJet III Family (Section 2.2). Any Parameters described in the PostScript Language Reference Manual or Supplement that are not included here are not supported in this product:

SYSTEM PARAMETERS

Key	Value Type	Default Value	Semantics
SystemParamsPassword	string (persistent/write only)	empty string ()	controls the ability of the setsystemparams operator to change the values of System Parameters.
StartJobPassword	string (persistent/write only)	empty string ()	controls the ability of the startjob operator to alter initial VM.
BuildTime	integer (read only)	xxxxxxxxx (for final build of code)	Time stamp identifying specific build of the PostScript interpreter.
ByteOrder	boolean (read only)	false	Native (preferred) order of multiple-byte numbers in binary encoded tokens; false indicates high-order first; true indicates low-order first
RealFormat	string (read only)	IEEE	Native (preferred) representation for real numbers in binary encoded tokens.
MaxFontCache	integer (persistent)	TBD (with maximum memory configuration),	Maximum bytes occupied by the font cache. This value is dependent on the amount of Optional memory fitted to the printer.
CurFontCache	integer (read only)	TBD (with maximum memory configuration)	Bytes currently occupied by the font cache. This value is dependent on the amount of Optional memory fitted to the printer.
MaxOutlineCache	integer (semi-persistent)	TBD	Maximum bytes occupied by cached character outlines (CharStrings) for fonts whose definitions are kept on disk instead of Virtual Memory.
CurOutlineCache	integer (read only)	0	Bytes currently occupied by CharStrings.
MaxUPathCache	integer (semi-persistent)	TBD	Maximum bytes occupied by the user path cache.
CurUPathCache	integer (read only)	0	Bytes currently occupied by the user path cache.
MaxFormCache	integer (semi-persistent)	TBD	Maximum bytes occupied by the form cache.
CurFormCache	integer (read only)	0	Bytes currently occupied by the form cache.

MaxPatternCache	integer (semi-persistent)	TBD	Maximum bytes occupied by the pattern cache.
CurPatternCache	integer (read only)	0	Bytes currently occupied by the pattern cache.
MaxScreenStorage	integer (semi-persistent)	TBD	Maximum bytes occupied by all active halftone screens,, including ones created by setscreen and saved by gsave.
CurScreenStorage	integer (read only)	TBD	Bytes currently occupied by all active halftone screens.
MaxDisplayList	integer (semi-persistent),	TBD	Maximum bytes occupied by display lists, excluding those held in caches.
CurDisplayList	integer (read only)	0	Bytes currently occupied by display lists.
GenericResourceDir	string (semi-persistent)	(%null)	Specifies the location in the file system for resources files.
Revision	integer (constant)	none	Indicates the current revision level of the machine dependent portion of the PostScript interpreter.
MaxSourceList	integer (semi-persistent)	TBD	Indicates the maximum number of bytes that can be utilized for source lists.
CurSourceList	integer (read only)	0	Indicates the number of bytes currently occupied by source lists.
JobTimeout	integer (persistent)	0	Indicates the number of seconds a job is allowed to execute before it is aborted and a timeout error is generated. This is the default value which may be overridden only within a job by a call to setuserparams which also contains a key by this name. Any value less than 0 will set the JobTimeout to 0. any value between 1 and 15 will set JobTimeout to 15. A value of 0 implies an infinite timeout. This is a product specific parameter.

WaitTimeout	integer (persistent)	40	Indicates the number of seconds the printer waits to receive additional characters from the host before it aborts the current job by executing a timeout error. This is the default value which may be overridden only within a job by a call to setuserparams which also contains a key by this name. Any value less than 0 will set the WaitTimeout to 0. Any value between 1 and 15 will set WaitTimeout to 15. A value of 0 implies an infinite timeout. This is a product specific parameter.
DoStartPage	boolean (persistent)	true	Specifies whether a startup page is to be printed. This value can be set persistently from the control panel. This is a product specific parameter.
DoAutoContinue	boolean (persistent)	true	Specifies whether to continue processing when an error occurs. This value can be set persistently from the control panel. If true, System Errors are displayed for 10 seconds and then the printer continues processing. PostScript Errors are not displayed. If false, System and PostScript Errors are displayed until the user presses the continue key. This is a product specific parameter.
DoPrintErrors	boolean (persistent)	false	Specifies whether to print error pages using a built in error handler when a PostScript error occurs. This value can be set persistently from the control panel. Also see PLRM p. 99. This is a product specific parameter.
RamSize	integer (read only)	memory configuration dependent	TBD with maximum memory configuration. This is the maximum amount of memory that is available to the PostScript Interpreter. This is a product specific parameter.

PageCount	integer (persistent/read only)	0 initially	This is a indication of the number of pages that have been printed by the marking engine. PageCount is incremented (by 10) every 10 pages. PageCount is not affected by a cold reset which returns values in NVRAM to factory default values. This is a product specific parameter.
PrinterName	string (persistent)	TBD	This is the default (non-LocalTalk) name of the printer. If an LocalTalk interface is installed, there are potentially two different default printernames depending on which communication mode is active (The LocalTalk name is stored on the LocalTalk board's NVRAM, not the resident printer NVRAM where all of the other Parameters are stored). Any 31 character string will be accepted for either name. This is a product specific parameter.
MaxImageBuffer	integer (semi-persistent)	TBD	Indicates the maximum number of bytes that can be utilized for a single image buffer.
GenericResourcePathSep	string (semi-persistent)	(/)	Path separator for the Generic resource directory
FontResourceDir	string (semi-persistent)	(fonts/)	Location of external fonts
FatalErrorAddress	integer (semi-persistent)	0	When a fatal software error occurs, the address at which the error occurs is stored in this parameter and is also sent to the host over the communications channel. A non-zero value indicates that a fatal system error has occurred earlier.
LicenseID	string (constant)	TBD	This parameter contains the Adobe-assigned license identifier. It's value is unique to each product.

Level 2 User Parameters

One category of PostScript Language Parameters for Level 2 to control the operation and behavior of the PostScript Interpreter is known as the User Parameters. User parameters, like System Parameters, are concerned with printer behavior in the areas of printer resource management, timeouts, and memory management (including garbage collection). However, unlike System Parameters, User Parameters only allow the programmer to modify his execution environment within the bounds of his own job and do not affect the global default configuration of the printer. Since User Parameters do not have any permanent affect on the behavior of the printer, they may be altered at will by the programmer (possibly to his/her own detriment) and do not require any special authorization such as a password to modify them.

All User Parameters, with the notable exception of **JobTimeOut**, are subject to the PostScript operators save and restore. This means that if an unencapsulated job changes User Parameters, these new values will be the initial values for subsequent encapsulated jobs. There are exceptions to this generalization. For a System Parameter whose name is the same as a User Parameter (such as **JobTimeOut** and **WaitTimeOut**), the value of the System Parameter is used to initialize the corresponding User Parameter at the beginning of each job.

The **setuserparams** and **currentuserparams** operators manipulate the User Parameter Sets described in the tables to be found in this chapter. The operators will be described first and then the tables will be presented.

setuserparams

dict setuserparams -

attempts to set one or more User Parameters whose keys and new values are contained in the dict operand. The dictionary is merely a container for key-value pairs; setuserparams reads the information from the dictionary, but does not retain the dictionary itself. User Parameters whose keys are not mentioned in dict are left unchanged.

Each parameter is identified by a key, which is always a name object. The value is usually an integer. If the named User Parameter does not exist in the implementation, it is ignored. If the specified value is the correct type, but is not achievable by the implementation, the nearest achievable value is substituted without error indication. See PostScript Language Reference Manual pp. 518-519, 571-579.

EXAMPLES:

1.) A user might wish to change the value of **JobTimeOut** to limit the processing time for his PostScript job and change the value of **WaitTimeOut** to limit the amount of time the interpreter should wait for host I/O data to be sent. These values would take affect for the duration of the users job only. The arguments to accomplish are as follows:

```
<< /JobTimeOut 500 /WaitTimeOut 250 >> setuserparams
```

2.) A user might to assign a name to his job for status readback, disable garbage collection, and increase the maximum allowable size of a character in the font cache:

```
<< /JobName (my_job) /VMReclaim -2 /MaxFontItem 20000 >> setuserparams
```

ERRORS: stackunderflow, typecheck, invalidaccess

currentuserparams

- currentuserparams *dict*

returns a dictionary containing the keys and current values of all user parameters that are defined in the implementation. The returned dictionary is a container for key-value pairs. Each execution of **currentuserparams** allocates and returns a new dictionary. See PostScript Language Reference Manual pp. 392, 571-579.

EXAMPLE:

Retrieve the names in the User Parameters dictionary:

```
currentuserparams      % put a copy of the User Parameters dictionary on the stack
{pop =} forall         % pop off all of the keys (or names) in the dictionary
MaxExecStack
VMThreshold
.
.
MaxDictStack           % all keys present will be displayed
```

ERRORS: stackoverflow, VMerror

User Parameters

The following User Parameters from Appendix C of the PostScript Language Reference Manual are implemented in this product. a brief description is included but for additional information the reader is referred to the PostScript Language Reference Manual pp 571-579, the PostScript Language Supplement (Section 3.1) and the PostScript Language Addendum for Hewlett-Packard LaserJet III Family (Section 2.1). Any Parameters described in the PostScript Language Reference Manual or the Supplement that are not included here are not supported in this product:

Key	Value Type	Default Value	Semantics
MaxFontItem	integer	12500	Maximum number of bytes occupied by the pixel array of a single character in the font cache.
MinFontCompress	integer	1250	Threshold at which a cached character is stored in compressed form instead of as a full pixel array.
MaxUPathItem	integer	5000	Maximum bytes occupied by a single cached user path.
MaxFormItem	integer	100000	Maximum bytes occupied by a single cached form.
MaxPatternItem	integer	20000	Maximum bytes occupied by a single cached pattern.
MaxScreenItem	integer	TBD	Maximum bytes occupied by a single halftone screen. This is hardware configuration dependent.
MaxOpStack	integer	100000	Maximum elements in operand stack.
MaxDictStack	integer	530	Maximum elements in dictionary stack.
MaxExecStack	integer	10015	Maximum elements in execution stack.
MaxLocalVM	integer (read only)	TBD	Maximum bytes occupied by values in local Virtual Memory.
VMReclaim	integer	0	0 = automatic garbage collection is enabled -1 = automatic garbage collection is disabled for local Virtual Memory -2 = automatic garbage collection is disabled for both local and global Virtual Memory.

VMThreshold	integer	40000	Frequency of automatic garbage collection,, which is triggered whenever this many bytes have be allocated since the previous collection.
JobTimeout	integer	0	Current value of JobTimeOut. This value is the only User Parameter that is NOT subject to save and restore. This Parameter is initialized to the value of the JobTimeOut System Parameter at the beginning of each job. This is a Product Specific Parameter.
WaitTimeout	integer	40	Current value of WaitTimeOut. This Parameter is initialized to the value of the WaitTimeOut System Parameter at the beginning of each job. This is a Product Specific Parameter.
JobName	string	null ()	Specifies the name of the current job for status responses. This is a Product Specific Parameter.
AccurateScreens	boolean	false	Controls whether or not the accurate screen algorithm will be used with the setsereen and setcolorscreen operators. See section 6.4.4 of the red book for a description of accurate screening for the sethalftone operator.

NOTE: that both the JobTimeout and WaitTimeout values are always 0 when the printer is in PostScript Interactive mode regardless of the system default value.

Level 1 System Setup Operators

The operator list in this and following chapters is included for compatibility with Level 1 PostScript. Most of these operators have preferred Level 2 mechanisms. See the individual operator description for more information on which Level 2 mechanism the operator is mapped into. In the Level 1 cartridge, all these Level 1 operators resided in the statusdict dictionary. The system setup operators are normally used by the system administrator when the printer is first installed.

buildtime

buildtime *Integer*

buildtime returns the value of the BuildTime key in currentsystemparams.

ERRORS: stackoverflow

byteorder

byteorder *Boolean*

byteorder returns the value of the ByteOrder key in currentsystemparams.

ERRORS: stackoverflow

realformat

realformat *String*

realformat returns the value of the RealFormat key in currentsystemparams.

ERRORS: stackoverflow

ramsize

ramsize *Integer*

ramsize returns the value of the RamSize key in currentsystemparams.

ERRORS: stackoverflow

checkpassword

password checkpassword *Boolean*

checkpassword checks whether string or int. (int. is converted to a string) is a valid password for either the **SystemParamsPassword** entry or the **StartJobPassword**. If valid, it returns true; otherwise it returns false (after delaying for one second). A returned value of true indicates that string or int. is a valid argument to **startjob** and **exitserver**.

ERRORS: stackunderflow, typecheck

appletalktype**

- appletalktype *string*

appletalktype is a string object that defines the TYPE field of the printer's AppleTalk name. The maximum number of characters supported for the TYPE field is 32. The last character in string must be a printing character. A printing character is any character in the ASCII range 33 through 126. The value can be changed using PostScript operators put, store, or def. The string will take effect at the end of the current job. If the appletalktype string is defined in an unencapsulated job, the string remains in effect until a language switch or a power cycle occurs. If the appletalktype string is defined in an encapsulated job, the TYPE field will not be affected. The default value is 'LaserWriter' because the Macintosh Print Manager assumes ALL PostScript printers are of type 'LaserWriter'. A redefinition of this string in the unencapsulated job environment will cause the LocalTalkType key of the (%LocalTalk%) device in the I/O Device Parameters to be updated as well.

This operator behaves exactly the same way in Level 2 as it did in Level 1. There is no new Level 2 mechanism to emulate for appletalktype.

DEFAULT: (LaserWriter)

ERRORS: stackoverflow

setdoidlefonts

Boolean setdoidlefonts -

pops the argument off the stack, but does not have any other function. This operator must be executed as part of an unencapsulated job. Idlefont caching is the process of rendering fonts into bit map images while the PostScript interpreter is not actively processing a print job. Control of the font cache is an internal mechanism in the Level 2 implementation. See pages 274-275 and 576-577 of the PostScript Language Reference Manual (Second Edition).

ERRORS: stackunderflow

doidlefonts

- doidlefonts *Boolean*

Always returns false. Idlefont caching is the process of rendering fonts into bit map images while the PostScript interpreter is not actively processing a print job. Control of the font cache is an internal mechanism in the Level 2 implementation. See pages 274-275 and 576-577 of the PostScript Language Reference Manual (Second Edition).

DEFAULT: false.

ERRORS: stackoverflow

idlefonts

- idlefonts *mark*

Pushes a mark onto the stack. Idlefont caching is the process of rendering fonts into bit map images while the PostScript interpreter is not actively processing a print job. Control of the font cache is an internal mechanism in the Level 2 implementation. See pages 274-275 and 576-577 of the PostScript Language Reference Manual (Second Edition).

DEFAULT: mark

ERRORS: stackoverflow

setdojamrecovery*

Boolean setdojamrecovery -

sets the value of the **ExitJamRecovery** entry in the current page device. If true, pages that jam in the exit path of the printer are reprinted. If false, pages that jam in the exit path are not reprinted. Overall performance is improved if ExitJamRecovery is false.

ERRORS: invalidaccess, stackunderflow, typecheck

dojamrecovery

- dojamrecovery *Boolean*

returns the value of the DoJamRecovery entry in the current page device.

DEFAULT: false (off)

ERRORS: stackoverflow

setdoprinterrors*

Boolean setdoprinterrors -

Sets the value of **DoPrintErrors** in the current system parameters which indicates whether or not PostScript error messages are printed after a PostScript error occurs. If Boolean is true, the printing of PostScript error message is enabled. If Boolean is false, PostScript error message are not printed. Error messages can be useful for debugging PostScript programs. When **DoPrintErrors** is true, errors are printed automatically by a built in error handler similar to the procedure "ehandler.ps" which is documented in Appendix A of the PostScript Level 1 Green Book. Errors 23, 24 and 25 will not produce an error page.

ERRORS: invalidaccess, stackunderflow, typecheck

doprinterrors

- doprinterrors *Boolean*

Returns the value of **DoPrintErrors** in the current system parameters which indicates whether or not PostScript error messages are printed. If Boolean is true, the printing of PostScript error message is enabled. If Boolean is false, PostScript error message are not printed.

DEFAULT: false (off)

ERRORS: stackoverflow

setdostartpage*

Boolean setdostartpage -

sets the value of the **DoStartPage** entry in the current system parameters which specifies whether a PostScript start page will be printed when the printer is subsequently powered on. If Boolean is true, then printing a PostScript start page is enabled. If Boolean is false, then printing a PostScript start page is disabled. This operator must be executed as part of an unencapsulated job. This value can also be set persistently from the control panel.

ERRORS: invalidaccess, stackunderflow, typecheck

dostartpage

- dostartpage *Boolean*

returns the value of the **DoStartPage** entry in the current system parameters. This is a boolean which specifies whether a start page is printed at the time the printer is powered on. If Boolean is true, then printing a PostScript start page is enabled. If Boolean is false, then printing a PostScript start page is disabled.

DEFAULT: true (on).

ERRORS: stackoverflow

setmargins

top left setmargins -

pops the 2 arguments off the stack (performing typechecking), but does not have any other function.

ERRORS: stackunderflow, typecheck

margins

-margin *0 0*

Always returns 0 0 as setting margins is not an option in the HP Color LaserJet.t

ERRORS: stackoverflow

pagecount

- pagecount *integer*

returns the value of **PageCount** in the current system parameters. This is the number of page images produced by showpage and copypage commands that have been executed since the machine was built. Note that on any given printer, the value of pagecount reflects the number of pages printed by the marking engine and includes pages printed by PCL or other printer languages. The PageCount value is updated (by 10) every 10 pages and is the value that is printed on the test page. If the printer is powered off before 10 pages have been printed, the PageCount is not updated.

ERRORS: stackoverflow

setprintername*

string setprintername -

stores string into the **PrinterName** entry in the current system params. This establishes string to be the printer's name. The string should be 31 or fewer characters long. It is advisable but not required to use characters within the ASCII printable range.

ERRORS: invalidaccess, limitcheck, stackunderflow, typecheck

printername

string printername *sub-string*

stores the value of **PrinterName** in the current system parameters into the supplied string (overwriting some initial portion of its value) and returns a string object designating the sub-string actually used. **PrinterName** can be up to 31 characters long. If string is not long enough to hold the printer name, string will be filled with as much of the printer's name as string will hold.

INITIAL DEFAULT: TBD

ERRORS: stackunderflow, typecheck, invalidaccess

product

- product *string*

returns a string object which is the printer's product name. The string resides in statusdict and is initialized to the value of the string product in systemdict. Any program that needs to know what type of printer it is running on should check this string.

DEFAULT: TBD

ERRORS: stackoverflow

jobname**

- jobname *string*

This operator returns a string with the same value as the user parameter **JobName** that specifies the name of the current job. This string should not contain the characters ':' or ']', since they would disrupt the syntax of the AppleTalk status message. The value can be changed using PostScript operators put, store, or def. The new value will take effect during the current job. If the jobname string is defined as part of an unencapsulated job, the string remains in effect until a power cycle occurs.

DEFAULT: ()

ERRORS: stackoverflow

revision

- revision *integer*

an integer that is the current revision level of the machine-dependent portion of the PostScript interpreter software. This is the current value of the Revision key in the System Parameters. (Note: The version operator in systemdict returns the version number of the machine independent portion of the PostScript interpreter.)

DEFAULT: None

ERRORS: stackoverflow

setsccbatch

channel baud parity setsccbatch -

In the current implementation, no values will be settable with this operator. Attempts to set values in the I/O Device Parameter sets with this operator using legitimate values will be accepted but will not have any other effect.

ERRORS: rangecheck, stackunderflow, typecheck

sccbatch

channel sccbatch *baud parity*

In this implementation, if the channel number is correct, the constants 9600 and 3 are returned.

ERRORS: rangecheck, stackoverflow, stackunderflow, typecheck

setdoautocontinue*

Boolean setdoautocontinue -

sets the **DoAutoContinue** entry in the current system parameters to the specified Boolean value. This operator must be executed as part of an unencapsulated job. If **DoAutoContinue** is true, system error messages will appear in the control panel display window for ten seconds before the printer returns on-line and resumes operation. PostScript error messages are not displayed. If **DoAutoContinue** is false, system and PostScript error messages remain in the display window until the user intervenes by pressing the ONLINE key on the control panel.

ERRORS: invalidaccess, stackunderflow, typecheck

doautocontinue

- doautocontinue *Boolean*

returns the value of **DoAutoContinue** in the current system parameters.

DEFAULT: true (on)

ERRORS: stackoverflow

setdoret*

integer setdoret -

sets the Resolution Enhancement technology value. This operator sets the value of REValue in the PostRenderingEnhanceDetails dictionary in the current page device to integer. Valid values for integer are 0 = OFF; 1 = LIGHT, 2 = MEDIUM, 3 = DARK.

ERRORS: rangecheck, stackunderflow, typecheck

doret

- doret *integer*

returns the value of **REValue** in the **PostRenderingEnhanceDetails** dictionary in the current page device. Valid values for integer are 0 = OFF; 1 = LIGHT, 2 = MEDIUM, 3 = DARK.

DEFAULT: 2 (Medium)

ERRORS: stackoverflow

firstside

- firstside *boolean*

needed for compatability with some drivers. In duplex printers, it indicates if we are currently on the front side or the back side of a page. In the HP Color LaserJet, firstside always returns true.

DEFAULT: true

ERRORS: stackoverflow

NOTE: The setpassword* operator is not supported in this implementation.

Level 1 Timeout Operators

There is a timeout facility for limiting the amount of time the server will remain in various states. The following Level 1 compatibility operators control the timeout facility.

setdefaulttimeouts*

job manualfeed wait setdefaulttimeouts -

sets the values of the **WaitTimeout** and **JobTimeout** entries in the current system parameters and the value of the **ManualFeedTimeout** entry in the current page device. All timeout values are integers denoting a time interval in seconds. The value 0 indicates that the corresponding timeout will never occur. This operator must be executed as part of an unencapsulated job.

ERRORS: invalidaccess, rangecheck, stackunderflow, typecheck.

defaulttimeouts

- defaulttimeouts job manualfeed wait

returns the value of the **JobTimeout** and **WaitTimeout** entries in the current system parameters and the value of the **ManualFeedTimeout** entry in the current page device.

DEFAULT: job = 0, manualfeed = 60, wait = 40

ERRORS: stackoverflow

setjobtimeout

integer setjobtimeout -

sets the value of **JobTimeout** in the current user parameters. This is a non-negative integer specifying a time interval in seconds. The job timeout value may be any integer but if negative, the timeout will be set to 0. If the current job continues for integer seconds, the PostScript interpreter executes a timeout error. The value 0 disables the job timeout altogether. In interactive mode, the job timeout is always 0.

ERRORS: stackunderflow, typecheck.

jobtimeout

- jobtimeout *integer*

returns the value of **JobTimeout** in the current user parameters. This is the number of seconds the printer will wait before a job timeout occurs. A value of 0 for integer means the job will never time out.

DEFAULT: 0

ERRORS: stackoverflow

manualfeedtimeout

- manualfeedtimeout *integer*

If defined, returns the current value of **manualfeedtimeout** in the statusdict dictionary. If defined, this value overrides the value of **ManualFeedTimeout** specified in the current page device dictionary; otherwise, **ManualFeedTimeout** is used to determine the timeout value. This is the number of seconds the printer will wait for a page to be manually fed before it will abort the current job and execute a timeout error. A value of 0 for integer means the job will never time out. An attempt to read the value of manualfeedtimeout prior to its definition will result in an undefined error.

DEFAULT: Undefined

ERRORS: stackoverflow, undefined

waittimeout

- waittimeout *integer*

returns the current value of **WaitTimeout** in the current user parameters. This is the number of seconds the printer will wait to receive additional characters from the host before it will abort the current job and execute a timeout error. A value of 0 for integer means the job will never time out. Redefining either waittimeout or the user parameter **WaitTimeout** redefines the other to the same value. In interactive mode, the wait timeout is always 0.

DEFAULT: 40

ERRORS: stackoverflow

Level 1 Page Type Operators

These compatibility operators request a specific paper size. Specifically, they set the **PageSize** entry in the current page device to the specified point-size array (a point is 1/72 of an inch). The **imagingBBox** entry in the page device is set to null to specify the standard bounding box for that paper size. The **PageSize** entry in the **Policies** dictionary of the page device is set to '6' which means "Select the next larger medium but do not adjust the page to fit". . The preferred method of page size selection is to use the **PageSize** key in the page device dictionary. A limitcheck error is possible if there is insufficient memory available for the requested imaging area.

OPERATOR	ARRAY SIZE
letter	a page size of [612 792] points
legal	a page size of [612 1008] points
tabloid	a page size of [792 1224] points
executivepage	a page size of [522 756] points
a4	a page size of [595 842] points
a3	a page size of [842 1190] points

Each operator listed takes no arguments form the stack and leaves none. Any marks made on the page before the operator is executed are erased.

ERRORS: limitcheck

Level 1 Tray Selection Operators

Tray selection operators are used to select trays containing desired media either by tray location or media size. All the operator procedures in this section are in statusdict.

setdefaultpapertray*

integer setdefaultpapertray -

The compatibility procedure for this operator stores the specified tray number into the first entry of the **Priority** array in the **InputAttributes** dictionary of the current page device. It then copies the **PageSize**, **MediaType**, **MediaWeight**, and **MediaColor** attributes of the chosen tray in the **InputAttributes** dictionary to the **PageSize**, **MediaType**, **MediaWeight**, and **MediaColo**r entries in the current page device._An integer value of 0 selects the upper tray, and an integer value of 1 selects the optional lower tray. The default imaging area is set at the beginning of the job based on the media size in the default paper tray. The results of executing this operator take effect immediately.

ERRORS: invalidaccess, rangecheck, stackunderflow, typecheck, configurationerror

defaultpapertray

- defaultpapertray *integer*

returns the first entry in the **Priority** array in the **InputAttributes** dictionary of the current page device. Possible values for integer are 0 and 1. An integer value of 0 means the upper tray is selected, and an integer value of 1 means the optional lower tray is selected.

DEFAULT: 0 (upper tray)

ERRORS: stackoverflow

setpapertray*

integer setpapertray -

This operator behaves in exactly the same way as **setdefaultpapertray**.

ERRORS: rangecheck, stackunderflow, typecheck, configurationerror

papertray

- papertray *integer*

returns the first entry in the **Priority** array in the **InputAttributes** dictionary of the current page device. Possible values for integer are 0 and 1. An integer value of 0 means the upper tray is selected, and an integer value of 1 means the optional lower tray is selected.

DEFAULT: 0 (upper tray)

ERRORS: stackoverflow

setdefaulttrayswitch*

boolean setdefaulttrayswitch -

The compatibility procedure for this operator stores the specified boolean value into the **TraySwitch** entry of the current page device. The results of executing this operator take effect immediately.

ERRORS: invalidaccess, stackunderflow, typecheck

defaulttrayswitch

- defaulttrayswitch *boolean*

returns the value of the **TraySwitch** key of the current page device.

DEFAULT: false (automatic tray selection is not enabled)

ERRORS: stackoverflow

settrayswitch

boolean settrayswitch -

This operator behaves in exactly the same way as **setdefaulttrayswitch**.

ERRORS: stackunderflow, typecheck

trayswitch

- papertray *Boolean*

Returns the value of the TraySwitch key in the current page device.

DEFAULT: false

ERRORS: stackoverflow

manualfeed

- manualfeed *Boolean*

selects the **ManualFeed** entry in the current page device. If true, input media are drawn from the manual feed position. If false, automatic feeding takes place. Setting **ManualFeed** to true is an assertion that the person operating the device will manually feed media that conform to the specified attributes **- PageSize**, **MediaColor**, **MediaWeight**, and **MediaType**. Thus, those attributes are not used to select from available media sources as is done normally.

If **ManualFeed** is selected and the **ManualFeedTimeout** in the current page device is nonzero, a timeout error will occur and the job will be aborted if paper is not fed into the printer during the specified time interval.

Manual feeding can also be enabled via the front panel using MANUAL FEED. At the beginning of each job, the front panel value of MANUAL FEED is copied into the **ManualFeed** entry in the page device dictionary. If the value is true, the **PageSize** entry is also initialized to the front panel MANUAL FEED value. If the **ManualFeed** entry is set persistently in an unencapsulated job, the value is copied back to the front panel value of MANUAL FEED and the current **PageSize** value becomes the new front panel MANUAL FEED size.

There is also a Boolean named **manualfeed** in statusdict that is initially set false. This boolean exists for backward compatibility with the Level 1 cartridge product and is completely independent of the setting of **ManualFeed** in the current page device. If this Boolean is set to true, manual feed is enabled, regardless of the value of the **ManualFeed** entry in the page device. Similarly, if manualfeedtimeout is defined in statusdict, its value overrides the value of **ManualFeedTimeout** specified in the page device.

DEFAULT: false

ERRORS: stackoverflow

papersize

- papersize sizename Boolean

Returns the name of the statusdict compatibility Page Type operator that describes the current media size. This always corresponds to the paper the job will actually print on. For example, 8.5 x 11 inch paper will result in letter being returned by papersize. The boolean returned indicates that the paper in the tray is oriented for short edged feed if equal to true or for long edge feed if equal to false. It should be noted that papersize does not install an imageable area for the paper size reported; it is merely informational. Loading and executing the userdict key will, however, install the imageable area for that paper size. This operator is not compatible with the Level 1 implementation which returned the Tray Type operator with the boolean instead.

ERRORS: stackoverflow

Tray types

The following compatibility operators set the **PageSize** entry in the current page device to the specified point-size array. The **ImagingBBox** entry in the page device is set to null to specify the standard bounding box for that paper size. The operation of these operators is thus quite similar as for the Page Type operators. However, the **PageSize** entry in the **Policies** dictionary of the page device is set to '0' (rather than 6) to indicate that a rangecheck error should be raised if the requested tray is not available. Also, a limitcheck error can occur because of insufficient memory for the requested imaging area. The preferred method of page size selection is using the **PageSize** key in the page device dictionary.

OPERATOR	ARRAY SIZE
lettertray	a page size of [612 792] points
legaltray	a page size of [6121008] points
executivetray	a page size of [522 756] points
a4tray	a page size of [595 842] points
a3tray	a page size of [842 1190] points
tabloidtray	a page size of [792 1224] points

Each operator expects no arguments from the stack and returns none to the stack. Any marks on the page prior to execution of the operator are erased.

ERRORS: rangecheck, limitcheck

Level 1 Output Bin Operators

The output bin control the media destination

setdefaultoutputtray*

integer setdefaultoutputtray -

Sets the default output bin through a call to setpagedevice. If *integer* is 0 (the face down bin) << /OutputFaceUp false >> setpagedevice is called. If *integer* is 1 (the face up bin) << /OutputFaceUp true >> setpagedevice is called. All other integer values cause a rangecheck.

ERRORS: rangecheck, stackunderflow, typecheck, invalidaccess

defaultoutputtray

- defaultoutputtray *integer*

Returns an integer indicating if the output bin is faceup or not. If pagedevice key OutputFaceUp is true, returns 1 otherwise 0.

DEFAULT: 0

ERRORS: stackoverflow

setoutputtray*

integer setoutputtray -

This operator behaves in exactly the same way as **setdefaultoutputtray.**

ERRORS: rangecheck, stackunderflow, typecheck

outputtray

- outputtray *integer*

This operator behaves in exactly the same way as **defaultoutputtray.**

DEFAULT: 0

ERRORS: stackoverflow

setpagestackorder*

boolean setpagestackorder -

This compatibility procedure stores the opposite of the specified boolean value into the OutputFaceUp entry of the current page device.

ERRORS: stackunderflow, typecheck, invalidaccess

pagestackorder

- pagestackorder *boolean*

returns the value of the opposite of the **OutputFaceUp** entry of the current page device.

DEFAULT: true

ERRORS: stackoverflow

Level 2 PostScript Errors

Various sorts of errors can occur during the execution of a PostScript program. Each error has a name and is associated with a PS ERROR number and an error handler. For each error, the error number is given followed by the error name and description. Errors 25-29 are unique to the Level 2 implementation. All others existed in Level 1, although QuitExecuted and Fatal Error have been expanded into 2 separate errors.

Any errors described in the PostScript Language Reference Manual or Supplement that are not included here are not supported in this product. See the PostScript Language Reference Manual pp. 64,99-101, 359, 363, 408 and 616-617 for more information.

00 **dictfull -** Dictionary full. A **def**, **put** or **store** operator attempted to define a new entry in a dictionary that is already full (i.e., whose length and maxlength are already equal). Now that dynamic allocation of dictionary space is available, this error is unlikely to occur very often.

01 **dictstackoverflow** - Dictionary stack overflow. The dictionary stack has grown too large. Too many **begins** (without corresponding **ends**) have pushed too many dictionaries on the dictionary stack.

02 **dictstackunderflow -** Dictionary stack under flow. An attempt has been made to remove the bottom most instance of userdict from the dictionary stack. dictstackunderflow occurs if an **end** is executed for which there was no corresponding **begin**.

03 **execstackoverflow -** Execution stack overflow. The execution stack has grown too large; procedure invocation is nested deeper than the PostScript interpreter permits.

04 **invalidaccess -** Invalid access of array, dictionary, file, or string object. An attempt has been made to reference an array, dictionary, file, or string object in a way that violates its access attribute (e.g., store into a read-only array). This error also occurs if **pathforall** is executed when the current path includes the result of a **charpath**.

05 **invalidexit -** No loop to exit from. An exit has been executed for which there is no dynamically enclosing looping context (**for**, **loop**, **repeat**, or **pathforall**), or it has attempted to leave the context of a **run** or **stopped** operator.

06 **invalidfileaccess** - Invalid file access. The access string specification to the file operator is unacceptable.

07 **invalidfont** - Invalid font name or dictionary. The operand to **makefont** or **setfont** is not a well-formed font dictionary. The invalidfont error may also be executed by other font operators upon discovering a font dictionary to be malformed in some way.

08 **invalidrestore -** An improper **restore** has been attempted. One or more of the operand, dictionary, or execution stacks contains composite objects whose values were created more recently than the **save** whose context is being restored. Since **restore** would destroy those

values but the stacks are unaffected by **restore**, the outcome would be undefined and cannot be allowed.

09 **ioerror -** An exception (other than end-of-file) has occurred during execution of one of the file operators. Attempting to write to an input file or to a file that has been closed will also cause an ioerror error.

10 **limitcheck -** A PostScript implementation limit has been exceeded.

11 **nocurrentpoint -** The current path is empty and thus there is no current point, but an operator requiring a current point has been executed (e.g. **lineto**, **curveto**, **currentpoint**, **show**, etc..). The most common cause of this error is neglecting to perform an initial **moveto**.

12 **rangecheck -** Operand out of bounds. A numeric operand' s value is outside the range expected by an operator (e.g., an array or string index is out of bounds, a negative number appears where a non-negative number is required, etc..)

13 **stackoverflow -** Operand stack overflow. The operand stack has grown too large. Too many objects have been pushed on the stack and not popped off.

14 **stackunderflow -** Operand stack under flow. An attempt has been made to remove an object from the operand stack when it is empty. This usually occurs because some operator did not have all of its required operands an the stack.

15 **syntaxerror -** Syntax error in program text. The scanner has encountered program text that does not conform the PostScript syntax rules. This can occur either during interpretation of an executable file or string object or during explicit invocation of the token operator.

16 **timeout -** A time limit has been exceeded. The PostScript interpreter timed out waiting for media to be manually fed, waiting for data to be received from the host, or timed out due to a job executing too long.

17 **typecheck -** Operand of the wrong type. Some operand's type is different from what an operator expects. The typecheck error is often the result of faulty stack manipulation, such as operands supplied in the wrong order or procedures leaving results on the stack when they aren't supposed to.

18 **undefined -** Name not known. A name used as a dictionary key in some context cannot be found. This occurs if a name is looked up explicit in a specified dictionary (**get**) or in the current dictionary stack (**load**) and is not found. It also occurs if an executable name is encountered by the interpreter and is not found in any dictionary on the dictionary stack. The undefined error is commonly caused by misspelling the operator name or by not having the dictionary containing the operator definition on the dictionary stack.

19 **undefinedfilename -** File not found. A file identified by a name string operand of file or **run** cannot be found or cannot be opened. The undefinedfilename error also occurs if the special file '%statementedit' or '%lineedit' is opened when the standard input file has reached end-of-file.

20 **undefinedresult -** Over/under flow or meaningless result. A numeric computation would produce a meaningless result or one that cannot be represented as a PostScript number. Possible causes include numeric overflow or under flow, division by zero, or inverse transformation of a non-invertible matrix.

21 **unmatchedmark -** Expected mark not on stack. A mark object is sought on the operand stack by the '**]**', **cleartomark**, or **counttomark** operator, but none is present.

22 **unregistered -** Internal error. An operator object has been executed for which the interpreter has no built-in action. This represents an internal malfunction in the PostScript interpreter and should never occur.

23 **VMerror -** An error has occurred in the virtual memory (VM) machinery. The most common problem is virtual memory is exhausted.

24 **quitexecuted -** The systemdict **quit** operator executed. The PostScript interpreter re-initialized virtual memory. All non-persistent information known by the PostScript interpreter was lost. This error causes a Front Panel Reset to occur.

25 **Fatal Error -** A firmware error internal to PostScript was detected.

26 **configurationerror** - A **setpagedevice** request cannot be satisfied.

27 **interrupt -** external interrupt request (e.g., control-c).

28 **undefinedresource -** resource instance not found.

See the PostScript Language Reference Manual pp.99-101 for a discussion of the handleerror name

Level 1 Compatibility

The intent of our strategy for Level 1 compatibility is that all files that printed on HP Level 1 products will still print using HP Level 2 products. This doesn't mean that all operators should or will behave in exactly the same way in this implementation as they did with the Level 1 products. For example, error reporting is not as robust for the Level 1 compatibility operators as it was before. It is possible to do some things that would have raised an error in the Level 1 implementation. However, no new errors should be generated that would not have occurred before. The legal Level 1 file would have ensured that the operator was in the correct execution state or the job would not have printed in the Level 1 environment. Behavior changes are described in this document, although in most cases they do not result in problems with compatibility.

NOTE: THE FOLLOWING LEVEL 1 OPERATORS HAVE DIFFERENT BEHAVIOR IN THE LEVEL 2 IMPLEMENTATION:

Level 1 invalidaccess and compatibility operators

In level 1, these operators caused an invalidaccess when done inside the server loop: setmargins, setdojamrecovery, setdefaultpapertray, setdefaultoutputtray, setpagestackorder, setdoprinterrors, setdostartpage, setprintername, setdoautocontinue, setdefaulttimeouts, setdoidlefonts, setsccbatch.

Because of the mapping that is done to level 2 operators, there are some changes. In level 2, this set will ALWAYS cause an invalidaccess when attempted inside the server loop: setdojamrecoverry, setdefaultpapertray, setdefaultoutputtray, setpagestackorder. This is the set that will yield an invalidaccess when done inside the server loop AND a SystemParamsPassword is set: setdoprinterrors, setdostartpage, setdoautocontinue, setdefaulttimeouts. Setdoidlefonts, setmargins, and setsccbatch "don't do anything" and will not be invalidaccess checked.

NOTE: distinction between setdefaultXXX and setXXX (for example setdefaultpapertray and setpapertray)

Other than the invalidaccess checking discussed above, PostScript makes NO distinction between the setdefaultXXX and setXXX versions of the operator. This has the side effect of setXXX done OUTSIDE the server loop will last persistently/semi-persistently. In level 1, the setXXX lasted only for that job even when outside the server loop.

cexec

cexec is an undocumented operator in Level 1. In Level 2 it will now produce a undefined error. Driver writers/PostScript programmers should not use this operator.

checkpassword

checkpassword in Level 1 returned true if the password argument matched the Level 1 system password. In Level 2 **checkpassword** returns true if the password argument matches either of the Level 2 passwords.

#copies

#copies is now an entry in systemdict and userdict. See the discussion at the end of Chapter 2 on **NumCopies** for an explanation of how #copies behaves in the Level 2 environment.

manualfeed

For more information on **manualfeed** operation, see chapter 9.

manualfeedtimeout

This is the only Level 1 operator in the **statusdict** dictionary that is not initialized to some default value. The timeout algorithm for Level 2 will thus not use its value. It can, however, be initialized and used. It should be initialized prior to being read to avoid generating an undefined error.

papersize

papersize is not compatible with Level 1 products. See Chapter 9 for more information.

product

The product string is now found in the **systemdict** as well as the **statusdict** dictionaries.

setdefaulttimeouts

setdefaultpapertray

setdostartpage

setdefaulttrayswitch

In Level 1 these operators did not take effect until the end of the job when executed in an unencapsulated job. They have an immediate effect when executed in a Level 2 unencapsulated job.

setdoidlefonts

doidlefonts

idlefonts

setdoidlefonts in Level 1 turned on idlefont caching. This is an internal mechanism in Level 2 and so has no function except to pop the argument off of the stack. **doidlefonts** in Level 1 indicated whether idlefont caching is enabled. In Level 2 it always returns false.

setdefaulttimeouts

setjobtimeout

Larger persistent values can be set using these operators in Level 2 than in Level 1:

OPERATOR	LEVEL I LIMITS	LEVEL II
manualfeedtimeout	0 <<= x<<= 255	0 <<= x <<= 65535
jobtimeout	0 or 15 <<= x <<= 255	0 or 15 <<= x <<= 4095
waittimeout	0 or 15 <<= x <<= 255	0 or 15 <<= x <<= 4095

setpassword

This operator will not be supported in the Level 2 products as there is no reasonable way to map this operator into the dual password architecture of Level 2 PostScript. Since this feature is not heavily used, it is reasonable to ask system administrators to use the new Level 2 password scheme. See Chapter 4 for more information.

setprintername

printername

setprintername can be used to set either the AppleTalk (%LocalTalk%) or non-LocalTalk (%Parallel%, %Serial%, %OptionalIO) printername. The AppleTalk printername behavior is unchanged.

setsccbatch

sccbatch

setsccbatch in Level 1 allowed the user to attempt to set the baud rate and appear to set the parity value for a particular channel. The **sccbatch** operator would return the new baud rate and the parity value that the user had requested even though the hardware was settings were unchanged. In Level 2, setsccbatch accepts values as before, and does not affect the hardware in any way. **sccbatch** returns the actual hardware settings, rather that the values input by the user.

setdojamrecovery

setmargins

setdoret

setdefaultrayswitch

settrayswitch

setdefaulttimeouts

setdefaultpapertray

setpapertray

setdefaultoutputtray

setoutputtray

setpagestackorder

all pagetype operators

all papertray operators

These operators do an implicit call to initgraphics, having the effect of clearing any marks already made on the page. This wasn't the case in level I.

setscreen

The **setscreen** operator should not be used to create patterns or undesirable behavior may result. Instead, use the **setpattern** operator to create patterns. See the PostScript Language Reference Manual pp. 315 and 742. It should be noted here that many level 1 applications used the setscreen operator to create patterns. Because of the orientation of the HP Color LaserJet printer (feeds "sideways"), patterns on the HP Color LaserJet (unless done "smartly") are rotated 90 degrees from those printed on other devices.

level 1 pagesize operators (letter, legal, etc)

Because of engine limitations (on the HP Color LaserJet, the image given to the engine can't be bigger than the paper it is to be printed on), the pagesize operators could not be implemented as a "pure level 1 emulation". A "pure level 1 emulation" would be to accept the pagesize request but don't change the tray from the current one, EVER, for any reason. The implementation we choose for the pagesize operators is to use pagesize policy 6 (select the next larger medium but do not adjust the page to fit). This retains the level one characteristic of NOT scaling the image. The side effect is that a different tray may be selected (may not be the highest priority tray).

Other - Previously Defined Operators

Drivers or applications that existed prior to Level 2 definition may have defined operators or variables that subsequently became reserved words in the Level 2 architecture. An example of this is the use of "setcolor" in Aldus Freehand 2.02. These jobs will not print correctly and the user will have to obtain an update to the driver or application at fault.

Other - spot function

The Level 2 spot function algorithm has changed. This means that some jobs using fills will have a slightly different appearance that they would have if printed using the Level 1 product.

Level 2 Compatibility

resourceforall

The Level 2 **resourceforall** operator returns the enumerated resources as its description implies. However, in the case of regular resources and resources used in defining new resource categories, it also returns an error. See the PostScript Language Reference Manual pp. 85-86 and 478-479 for more information on the behavior of **resourceforall**.

Chapter 7:   Front Panel Errors

Front Panel Errors

The control panel has many uses:

- Communication of device status to users via the display or LEDs

- Configuration of language specific (PCL or PostScript) environment parameters.

- Configuration of printer specific parameters (i.e. RET, I/O Switching, Personality Switching).

- Allows printing of self tests and typeface lists

This chapterdocument applies to both PCL and PostScript (where applicable). Any functionality difference between these two languages will be noted.

Device Status

Device status is communicated by messages displayed on the VFD or the LED indicators (most often device status will use both methods in conjunction).

LED Indicators

The four LEDs on the HP Color LaserJet control panel are used to convey information to the user. Many printer states will illuminate/extinguish these LEDs, therefore their behavior can not be summed without explaining each in detail.

1. ON LINE LED

The green ON LINE indicator has three states: ON (illuminated), OFF (extinguished), and FLASHING (alternating between illuminated and extinguished). The printer is in a on line state, able to accept and process data, when this indicator is ON. The printer is in a off line state, unable to accept and process additional data, when this indicator is OFF.

Note: There can be circumstances where the ON LINE indicator will be OFF, and paper will still be moving through the printer. Previous (LJ4 and 4si) flash the ON LINE indicator until paper is no longer moving through the printer.

This LED will FLASH when a user pressed the on line key to go off line, and there is still media moving through the printer (the printer really can not go off line until the paper path is clear). So the FLASHING state will inform the user that the printer is going to go off line when possible (after all strips on the current page have been rendered and sent for printing).

2. DATA LED

The amber DATA indicator has three states: ON (illuminated), OFF (extinguished), and FLASHING (alternating between illuminated and extinguished). When this LED is ON, the printer has buffered data (data that is waiting to be printed). An OFF state means there is no buffered (or pending) data to be printed. If the printer is accepting data from any physical I/O connection or processing buffered data[1], this LED will be FLASHING.

3. ATTENTION LED

The red ATTENTION indicator has two states: ON (illuminated), and OFF (extinguished). When a printer needs user interaction to correct a problem or there is a service failure, this LED will be ON[2]. If this LED is OFF, the printer is not in need of attention. When this LED is ON, the printer will be off line.

4. MANUAL FEED LED

The amber MANUAL FEED indicator has two states: ON (illuminated) and OFF (extinguished). This LED is ON when manual feed is selected via the control panel or data stream. If this LED is ON, by a control panel setting, it may be turned OFF by a print job that changes the current paper source. However, after the print job is finished, the LED will return to the illuminated (ON) state.

This LED is ON when manual feed is selected via the data stream regardless of the control panel configuration. However, when media motion completes and an end-of-job reset (of any type) occurs, the LED returns to the current control panel configuration.

In the case of a successful manual feed paper mount override[3], this LED will be OFF until media motion completes. At this time, this LED would return to the state that reflects the current control panel configuration.

This LED is OFF if manual feed has not been selected by the control panel or data stream. Job boundaries, data stream resets and control panel resets set this LED to the current control panel configuration. Cold and Menu resets set this LED to the OFF state (OFF is factory default).

[1] PCL must sometimes process "buffered data" when it receives a print job that does not end with a valid PCL terminator. PostScript jobs always end in CTRL-D, therefore the "processing buffered data" scenerio does not apply.

[2] Most often, the VFD will indicate what actions are necessary to extinguish the ATTENTIION LED

[3]

Vacuum Fluorescent Display (VFD)

Most messages displayed on the VFD are multiple component messages, prefixed with a number. These digits that prefix many messages correspond to numbers used in previous LaserJets. There are four (4) categories of printer messages displayed on the VFD. All messages communicate the current state of the printer, but some are considered "informative" messages, while other messages describe printer problems or fatal errors.

1. Device Status Messages

Device status messages are displayed to reflect the current printer state. This class of message merely informs the user of normal printer operation. Users are not required to interact with the printer to clear these messages from the VFD[4]. You will not see these messages if the printer is in an error state. Messages in this class are most often identified by mnemonics 00-09, 11, 15, 16,18, 19, 23, 24, and 48. Refer to Devices Status Messages section for a complete list.

2. Device Attendance Messages

Device attendance messages inform the user that printer attention is needed, such as adding media or clearing a media jam. Messages in this class are most often identified by mnemonics 10-14 and 19. These messages disappear once the printer detects that the correct actions were taken. Refer to device attendance messages.

3. Error Messages

Error messages inform users of data and print errors, usually requiring some device attendance. Messages in this class are identified by the mnemonics 20, 22 and 30. In cases where the AUTO CONTINUE print menu variable is ON, these errors will clear themselves after ten (10) seconds (the printer return to the on line state). If AUTO CONTINUE is OFF, the printer will continue to display the error message until cleared by an on line key press. 30 PS ERRORs are not displayed if AUTO CONTINUE is ON.

4. Service Messages (Fatal/Non-fatal)

Service messages inform the user of printer failures. Messages in this class are identified by mnemonics 50-89. If the failure is continuable (meaning the user can recover from the error, without powering off the printer), the bottom line of the display will read "ERROR". If the error is fatal, the bottom line of the display will read "SERVICE ERROR". Fatal errors must be cleared by power cycling the printer. The AUTO CONTINUE print menu variable has no effect on this class of messages. Refer to service errors for a complete list .

Note: All service errors are recorded into the error log.

5. 79, 80 & 89 Service Errors

All errors in this class are fatal errors and halt printer operation. They are not continuable, other than cycling power.

The numeric prefix of a all device attendance and error messages flash at approximately a 1 Hz rate, 0.5 seconds on and 0.5 seconds off. The characters of the message are not flashed.

[4] Sometimes, interaction will not make the message go away

Control panel messages are prioritized in the following order:

1. Service Messages

2. Error Messages

3. Device Attendance

4. Device Status

For example, if there are two (2) printer status conditions, one being a Device Attendance message and the other a Device Status message, the Device Attendance message would be displayed first (meaning the Device Status message would not be displayed until the Device Attendance message was "cleared").

After configuring a menu item the user may choose to configure another menu item using the procedure described above or the user may choose to exit the menu by pressing the ON LINE key, returning the printer to the on line state.

Device Status Messages

Device status messages communicate general printer status to the user (status that does not require control panel interaction to clear). Many Device Status messages are temporary messages, appearing in a particular printer state momentarily (i.e. "06 PRINTING PCL TYPEFACES"). However, some Device Status messages communicate status in what is called a background condition. This is condition that will most likely result in a device attendance message if not attended (i.e. "11.4 FRONT TRAY EMPTY" or "TONER LOW OFF LINE TO FILL"). Other background conditions occur when a user "continues" from a device attendance error (i.e. "19.2 COLOR DEVEL USER MAINTENANCE"). Most Device Status messages are identified by mnemonics 00-09, 11, 15, 16,18, 19, 23, 24, and 48.

```
00 PCL           00 POSTSCRIPT
   READY            READY
```

This message is displayed whenever the device is ready for data and no status or device attendance messages are pending.

```
02 PCL           02 POSTSCRIPT
   WARMING UP       WARMING UP
```

This message is displayed while the engine has not reached operating temperature. The ON LINE key may be used to place the printer on line. However, the menus can't be accessed while on line.

```
04 CONTINUOUS
   SELF TEST
```

This message indicates that the printer is in continuous self test mode. The mnemonic portion of the message flashes at a 1Hz rate while the printer is aborting the continuous self test. If not in abort state, the mnemonic does not flash. When the continuous self test is completed, the printer will return to the off line state.

```
05 INTERNAL
   SELF TEST
```

This message is displayed during the non-printing portion of a normal printer self test (i.e. SELF TEST from the TEST MENU or power-on self test) or when the printer is powered on.

```
06 PRINTING PCL   06 PRINTING PS
   SELF TEST         SELF TEST
```

This message is displayed during the printing portion of a normal printer self test (a self test initiated from the control panel or data stream). The mnemonic portion of the message flashes at a 1Hz rate while the printer is aborting the self test. When the self test is completed, the printer will return to the off line state.

```
06 PRINTING PCL    06 PRINTING PS
   TYPEFACES          TYPEFACES
```

This message is displayed while the PCL or PostScript FONT PAGE is being formatted and printed. Upon successful completion of the FONT PAGE the printer will return to the off line state.

```
06 PRINTING PCL    06 PRINTING PS
   DEMO PAGE          DEMO PAGE
```

This message is displayed while the PCL or PS Demo Page is being formatted and printed. Upon successful completion of the Demo Page, the printer will return to the off line state.

```
07 RESET
```

This message is displayed during execution of a control panel reset. When the reset is complete, the printer will return to the on line state.

```
08 COLD RESET
```

This message is displayed during the execution of the cold reset power on mode. When the reset is completed, the printer will return to the off line state.

```
09 MENU RESET
```

This message is displayed during the execution of a control panel menu reset. When the reset is completed, the printer will return to the on line state.

```
11.4 FRONT TRAY    11.4 FRONT TRAY
     EMPTY              EMPTY TRN
11.5 REAR TRAY     11.5 REAR TRAY
     EMPTY              EMPTY TRN
```

This message is displayed when the front or rear tray is empty or not present.

```
TONER LOW
OFF LINE TO FILL
```

This message is displayed whenever a toner sensor detects a low toner supply for one or more colors (C,M,Y,K). On the HP Color LaserJet engine, all engine movement must stop before the appropriate toner lock is released. Therefore, the user must take the printer off line before the color(s) will be indicated (and wait for paper motion to stop). If the printer is in an off line state, this message will not appear, as the engine components are not moving.

```
TONER LOW
OPENING INTRLOCK
```

This message is displayed when there is toner low or out, the user has pressed the ON LINE key to take the printer off line, but the printer has not opened a hopper interlock. Without the hopper interlock open, the user can't fill the hopper.

```
TONER LOW   KCMY
ADD [color]
```

This message is displayed whenever a toner sensor detects a low toner supply for one or more colors (K,C,M,Y). The sequence "KCMY" denotes all hoppers that need filling. [Color] denotes the hopper, out of the "KCMY" sequence, that is unlocked and ready to fill immediately. Although the color is abbreviated in the upper line, the lower line will describe the [color] as BLACK, MAGENTA, YELLOW, or CYAN. The printer will stay off line until the problem is corrected by adding toner into the hopper indicated by [color] or if the user places the printer on line.

```
15 PRINTING
   ENGINE TEST
```

This message is displayed whenever the printer is printing the special engine test page, produced when the user presses a small button located in the right rear of the printer.

```
16.5 REPLACE
  COLLECTION KIT
```

This message indicates that members of the collection box kit are nearing the end of specified life . Installing a new collection box kit will correct this problem. Since this is a warning message, the printer will not automatically go off line when this message is displayed.

```
16.6 REPLACE
 COATING PAD KIT
```

This message indicates that members of the coating pad kit are at the end of specified life. The printer will allow users to continue printing, even though the coating pad kit is past life. This message is displayed as a reminder, since the user has selected to continue using the old coating pad kit, that the coating pad kit is past life, and should be replaced.

```
19.1 BLACK DEVEL
USER MAINTENANCE
```

This message indicates that the black developer is past its specified life . The printer will allow users to continue printing, even though the black developer is past life. This message is displayed as a reminder, since the user has selected to continue using the old black developer, that the black developer is past life, and should be replaced.

```
19.2 COLOR DEVEL
USER MAINTENANCE
```

This message indicates that the color developer is past its specified life. The printer will allow users to continue printing, even though the color developer is past life. This message is displayed as a reminder, since the user has selected to continue using the color developer, that the color developer is past life, and should be replaced.

```
19.3 DRUM
USER
MAINTENANCE
```

This message indicates that the drum is past its specified life (as determined by Konica and the engine). The printer will allow users to continue printing, even though the drum is past life. This message is displayed as a reminder, since the user has selected to continue using the old drum, that the drum is past life, and should be replaced.

```
19.4 FUSER
USER MAINTENANCE
```

This message indicates that the fuser is past its specified life. The printer will allow users to continue printing, even though the fuser is past life. This message is displayed as a reminder, since the user has selected to continue using the old fuser, that the fuser is past life, and should be replaced.

```
19.5 TRANS BELT
USER MAINTENANCE
```

This message indicates that the transfer assembly is past life. The printer will allow users to continue printing, even though the transfer assembly is past life. This message is displayed as a reminder, since the user has selected to continue using the old transfer assembly, that the transfer assembly is past life, and should be replaced.

```
21 MIO
   INITIALIZING
```

This message indicates that the MIO card is unable to accept data because it is initializing. For more information, refer to the MIO card specification. A self test page can be printed to refer to current MIO card settings.

```
23 MIO
   NOT READY
```

This message indicates that the MIO card is unable to accept data. For more information, refer to the MIO card specification. A self test page can be printed to refer to current MIO card settings.

```
24 BUSY MOVING
   TONER
```

This message indicates that the printer is moving toner from the hoppers to the developers. This is cleared once the printer has the proper toner density for all four colors. This occurs without any interruption to the current state of the printer.

```
48 INVALID JOB
   ABORTING JOB
```

This message indicates that the printer received data that cannot be processed by any installed personality. It will continue to be displayed until data is received that can be processed by an installed personality.

```
INITIALIZING
NVRAM
```

This message occurs when NVRAM must be re-initialized with factory defaults because one or more values were erroneous.[5] After initialization is complete, the printer self test messages ("05 INTERNAL SELF TEST") will appear.

```
CONFIG LANGUAGE
```

This Message is displayed when the ENTER key is held down while turning on the printer. It is displayed as a confirmation that the printer will boot into a language configuration mode. This is the mode in which users can set the language that displayed messages will use.

```
ERROR LOG
```

This Message is displayed when the FORMFEED key is held down while turning on the printer. It is displayed as a confirmation that the printer will boot into error log mode. This is the mode in which users can access a listing of the last 50 printer errors.

```
DEMO MODE
```

[5] This message can only occur when powering on the printer. It should not occur during normal printer operation (after the printer is powered on).

This Message is displayed when the '+' key is held down while turning on the printer. It is displayed as a confirmation that the printer will boot into demo mode. This is a mode in which users can select (using front panel keys) to printer various print samples that exist on a specially installed SIMM.[6]

```
SERVICE MODE
```

This message is displayed when the ONLINE, FORMFEED and ITEM keys are held down while turning on the printer. It is displayed as a confirmation that the printer will boot into service mode. This is a mode in which service personnel use to fix special problems that may occur.

```
EXTENDED
DIAGNOSTICS MODE
```

This message is displayed when the RESET, MENU and '+' keys are held down while turning on the printer. It is displayed as a confirmation that the printer will boot into extended diagnostics mode. This is a mode in which service personnel can extensively test the printer firmware.

```
PCL PRINT MENU    PS PRINT MENU
```

This message is displayed on the top line when the user enters the PCL or PostScript Print Menu. The second line of the display shows the appropriate menu item.

```
PCL CONFIG MENU   PS CONFIG MENU
```

This message is displayed on the top line when the user enters the PCL or PostScript Configuration Menu. The second line of the display shows the appropriate menu item.

```
PARALLEL MENU
```

This message is displayed on the top line when the user enters the Parallel Menu. The second line of the display shows the appropriate menu item.

```
MIO CONFIG MENU
```

[6] The "DEMO MODE" message is not localized. Also, this message is only displayed if the special demo SIMM is installed.

This message is displayed on the top line when the user enters the MIO Menu. The second line of the display shows the appropriate menu item.

```
PCL TEST MENU     PS TEST MENU
```

This message is displayed on the top line when the user enters the PCL or PostScript Testing Menu. The second line of the display shows the appropriate menu item.

```
SWITCHING TO
XXXXXXXXXXX
```

This message is displayed when the user changes the "SYS=XXXXXXXXXX" configuration menu item (and pressing enter, submitting the change to NVRAM). It is displayed until the selected personality is ready.

```
SM CONFIG MENU
```

This message is displayed on the top line when the user enters the Service Mode Configuration Menu. The second line of the display shows the appropriate menu item.

```
SM TEST MENU
```

This message is displayed on the top line when the user enters the Service Mode Testing Menu. The second line of the display shows the appropriate menu item.

```
LJ4c DEMO MODE
PRESS ANY KEY
```

This message is displayed when the demo SIMM is installed and the appropriate key was held during power on. This message will be displayed until the printer is powered off. The demo SIMM must be installed **and** the '+' key must be held down for this message to appear (see `"DEMO MODE"` message above).

```
INSTALLING NEW
DEVELOPER    /
```

This message will show the user that the new developer sequence has started. The installation sequence will last for approximately 3 minutes per color. The pinwheel will spin for the duration of the developer installation. The stages of the pinwheel are '-' '\' '-' '/'. The first message in this sequence is "PRESS ON LINE TO CONTINUE".

Device Attendance Messages

Device attendance messages communicate to the user that they must perform some action such as adding media or clearing a media jam. Device attendance messages disappear from the display once attended. In the case of developer and drum conditions (user maintenance), pressing ON LINE will place the message in the background (making it a Device Status Message). All device attendance messages are displayed in the selected

If these messages cause the printer to go off line, the ATTENTION indicator will turn on in many cases, informing the user that some action is necessary.

The digital mnemonic portion of device attendance messages flashes at a 1 Hz rate, 0.5 seconds on and 0.5 seconds off. The alpha portion of these device attendance messages are continuously displayed.

```
10 PRESS RESET
   TO ACTIVATE
```

This message is displayed when the user changes a print menu item while there is buffered data and/or temporary macros or fonts present in printer memory. It will also occur if the default system (SYS=) item in the configuration menu is changed. The user can press the RESET key to reset the printer and go on line (buffered data, temporary macros, and temporary fonts will be deleted), or press the ON LINE key to go on line without a reset. In either case, the print menu items will be updated in NVRAM. The attention LED is **not** lit when this message is displayed.

11.1 FRONT TRAY LOAD [paper]	11.1 FRONT TRAY LOAD [paper] TRN
11.2 REAR TRAY LOAD [paper]	11.2 REAR TRAY LOAD [paper] TRN
11.3 MANUAL FEED [paper]	11.3 MANUAL FEED [paper] TRN

This message is displayed when the printer needs a specific media size loaded into a tray. [paper] is one of the supported paper sizes. The first two digits of this message flash. Loading the desired media size into the desired tray will correct the problem.

```
12 CLOSE TOP
   OR SIDE DOOR
```

This message is displayed whenever the engine senses that the upper main body of the printer or the auxiliary side door is not closed properly. After the appropriate action is taken, the device status message `"02 PCL WARMING UP"` may appear in the display, depending on whether the fuser has reached operating temperature. After warming up, the printer will return on line automatically. When the printer is on line, the display will read `"00 PCL READY"`. The attention LED is lit when this message is displayed.

13.1 CLEAR DRUM WINDING JAM	13.5 FUSER JAM *see note*

13.2 CLEAR OUTPUT JAM	13.6 CLEAR REAR TRAY INPUT JAM
13.3 CLEAR MANUAL JAM	13.7 CLEAR REAR TRAY PAPER JAM
13.4 CLEAR FRONT TRAY INPUT JAM	13.8 CLEAR PAPER JAM

Note: The fuser jam message will have instructions that will "scroll" across the bottom line. These fuser jam instructions, when used, will prevent the user having to remove the entire fuser assembly.

One of these messages is displayed when the engine senses a media jam. The user must remove the jammed media and close the printer door (if it was opened to clear the jam) to resume printing. Note the printer must be opened to clear the media jam error. The attention LED is lit when this message is displayed.

```
TONER OUT   KCMY
ADD [color]
```

This message is displayed whenever a toner sensor detects an empty toner supply for one or more colors (K,C,M,Y). The sequence "KCMY" denotes all hoppers that need filling. [Color] denotes the hopper, out of the "KCMY" sequence, that is unlocked and ready to fill immediately. Although the color is abbreviated in the upper line, the lower line will describe the [color] as BLACK, MAGENTA, YELLOW, or CYAN. The printer will stay off line until the problem is corrected by adding toner into the hopper indicated by [color]. The printer will return ONLINE when toner is placed into the hopper. The ATTENTION indicator is lit when this message is displayed.

```
14.5 REPLACE
  COLLECTION KIT
```

This message indicates that members of the collection box kit are at the end of specified life (as determined by Konica and the engine). An operator CANNOT continue printing (Upon receipt of this condition, the printer will go automatically off line). Replacing all members of the collection box kit will correct this problem. The attention LED is lit when this message is displayed.

```
14.6 REPLACE
 COATING PAD KIT
```

This message indicates that members of the coating pad kit are at the end of specified life (as determined by Konica and the engine). An operator must press the ON LINE key to continue printing (Upon receipt of this condition, the printer will go automatically off line). Replacing all members of the coating kit will correct this problem. The attention LED is lit when this message is displayed.

```
17.1 INSTALL
 BLACK DEVELOPER
```

This message indicates that the black developer is either not installed or not correctly installed in the printer. To correct this error, the user must insert a black developer or make sure the installed developer is properly seated. The attention LED is lit when this message is displayed.

```
17.2 INSTALL
 COLOR DEVELOPER
```

This message indicates that the color developer is either not installed or not correctly installed in the printer. To correct this error, the user must insert a color developer or make sure the installed developer is properly seated. The attention LED is lit when this message is displayed.

```
17.4 INSTALL
     FUSER
```

This message indicates that the fuser is either not installed or not correctly installed in the printer. To correct this error, the user must insert a fuser or make sure the installed fuser is properly seated. The attention LED is lit when this message is displayed.

```
17.6 INSTALL
  COLLECTION BOX
```

This message is displayed when the toner collection box (or drum) is not installed. The printer will stay off line until the problem is corrected by placing an empty collection box (or new drum with collection box) in the printer. The attention LED is lit when this message is displayed.

```
17.7 REMOVE
     DRUM COVER
```

This message indicates that the protective drum cover (used in shipping) is still on the drum. The printer will stay off line until the problem is corrected by removing the drum cover. The attention led is lit when this message is displayed.

```
17.8 REINSTALL
  DRUM
```

This message indicates that the drum is either not properly seated (the electrical connections) or the drum winding sensor is broken. The printer will stay in this state until the drum has been re-inserted into position three (3) times. After the third time, a 66.2 JAM SENSOR SERVICE ERROR, indicating the sensor must be replaced or the drum connection is faulty. In either case, service is required.

```
17.9 REINSTALL
     FUSER
```

This message indicates that the fuser is not properly seated (the electrical connections). The printer will stay in this state until the fuser has been re-inserted into position. The attention LED is lit when this message is displayed.

```
18.1 CLEAN TRANS
     CORONA WIRE
```

This message indicates the printer received a F27 or F28 fatal power supply error. By cleaning the transfer assembly corona wire, the user will open the top door, thereby resetting the fatal error latch in the power supply. This error message should not re-appear if power cycled. The attention LED is lit when this message is displayed.

```
18.2 CLEAN DRUM
     CORONA WIRE
```

This message indicates the printer received a F29 fatal power supply error. By cleaning the drum corona wire, the user will open the top door, thereby resetting the fatal error latch in the power supply. This error message should not re-appear if power cycled. The attention LED is lit when this message is displayed.

```
18.3 REINSERT
     DEVELOPERS
```

This message indicates the printer received a F30 fatal power supply error. By re-inserting both developers, the top and side doors must be opened, thereby resetting the fatal error latch in the power supply. This message should not re-appear if power cycled. The attention LED is lit when this message is displayed.

```
19.1 BLACK DEVEL
USER MAINTENANCE
```

This message indicates that the black developer is at the end of its specified life . Users must press the ON LINE key to continue printing. Replacing the old developer with a new developer will correct this problem. The attention LED is lit when this message is displayed.

```
19.2 COLOR DEVEL
USER MAINTENANCE
```

This message indicates that the color developer is at the end of its specified life (as determined by Konica and the engine). Users must press the ON LINE key to continue printing. Replacing the old developer with a new developer will correct this problem. The attention LED is lit when this message is displayed.

```
19.3 DRUM
USER MAINTENANCE
```

This message indicates that the drum is at the end of its specified life. Users must press the ON LINE key to continue printing. Replacing the old drum with a new drum will correct this problem. The attention LED is lit when this message is displayed.

```
19.4 FUSER
USER MAINTENANCE
```

This message indicates that the fuser is at the end of its specified life. An operator must press ON LINE to continue printing. Replacing the old fuser with a new fuser will correct this problem. The attention LED is lit when this message is displayed.

```
19.5 TRANS BELT
USER MAINTENANCE
```

This message indicates that the transfer assembly is at the end of its specified life. An operator must press ON LINE to continue printing. Replacing the old transfer assembly with a new assembly will correct this problem. The attention LED is lit when this message is displayed.

```
PRESS ON LINE
TO CONTINUE
```

This message indicates that the printer will go through the new developer installation sequence, when the user presses the ON LINE key (no other key will developer installation to start). The attention LED is **not** illuminated when this message occurs.

```
NEW DRUM INST.
DRUM NUMBER=XXX
```

This message indicates that the printer is awaiting the user to enter the drum number stamped on the drum unit. The '+' and '-' keys can be used to input this information. The number will be an integer in the range of [0-127].

```
PLEASE VERIFY
DRUM NUMBER=XXX
```

This message indicates that the user pressed the ENTER key, expressing that the displayed number is correct. The user must verify the number with that on the drum (this is just a suggestion, not a requirement) adjust the number and press ENTER again to send the integer to the engine.

```
POWERSAVE MODE
```

This message occurs when the printer transitions into the reduced energy consumption mode.

```
FE FONT CART ERR
CYCLE POWER
```

This message indicates that the font cartridge was removed when the printer was on line. This error occurs whether or not the cartridge was being accessed. This is a fatal error condition. The control panel keys are inoperative and depressing any/all keys causes no action. To clear this error condition, the user must power cycle the printer. The attention LED is lit when this message is displayed. The `"FE"` portion of this message flashes at a 1Hz rate.

REINSERT TOP FONT CARTRIDGE	REINSERT BOTTOM FONT CARTRIDGE	REINSERT BOTH FONT CARTRIDGES

These messages indicate that either the top or bottom font cartridge or both font cartridges were removed while the printer was off line and contained buffered data. To clear this error the user must re-insert the specified cartridge or cartridges and then press the ON LINE key to return the printer to the on line state. The attention LED is lit when this message is displayed. Nothing flashes with this message.

NO FONTS TOP FONT CARTRIDGE	NO FONTS BOTTOM FONT CARTRIDGE	NO FONTS BOTH FONT CARTRIDGES

These messages indicate that either the top or bottom font cartridge or both font cartridges could not be read by the printer. To clear this error the user must re-insert the specified cartridge or cartridges and then press the ON LINE key to return the printer to the on line state. If the problem persists, the cartridge(s) is(are) bad and should be replaced. This message does not flash. The attention LED is lit when this message is displayed. Nothing flashes with this message.

Continuable Error Messages

Printer error messages communicate the necessity for printer interaction or printer failure. Error messages in the continuable printer class can be cleared by pressing the ON LINE key. Also, these errors are affected by the auto continue item of the configuration menu. If auto continue is ON, these errors will be automatically cleared after approximately ten (10) seconds. If auto continue is OFF, these errors must be continued by user interaction.[7]

The mnemonic portion of all error messages flashes at a 1 Hz rate, 0.5 seconds on and 0.5 seconds off. The alpha portion of all error messages are continuously displayed.

```
20 MEMORY
   OVERFLOW
```

This message indicates that more data has been received from the host than fits in the printer's internal memory. To continue printing, press the ON LINE key. Only the amount of data that fits in the printer's internal memory is printed. The attention LED is lit when this message is displayed.

```
21 MEMORY
   OUT
```

This message indicates that the engine has run out of memory. The imaging of the current job cannot continue until the engine clears memory. To clear memory, press the ON LINE key. This will result in data loss. If this error occurs often, increase the amount of memory in the printer. The attention LED is lit when this message is displayed.

```
22 I/O CONFIG
   ERROR
```

This error indicates that the printer's receiving buffer has overflowed during a busy state. The host CPU is not obeying pacing mechanism of MIO link. Pressing the ON LINE key resumes printing but results in a loss of data. The attention LED is lit when this message is displayed.

```
30 POSTSCRIPT
   ERROR XX
```

This error indicates that a PostScript error, indicated by the integer XX, occurred. Pressing the ON LINE key lets PostScript do whatever it does with it. The attention LED is lit when this message is displayed.

[7] 30 PS ERRORs are not displayed if AUTO CONTINUE is ON.

```
40 I/O DATA
   ERROR
```

This error indicates that a data error (parity, framing, or line overrun) has occurred during the reception of data from the computer. There is an abnormal break in communication. To continue printing, press the ON LINE key. The attention LED is lit when this message is displayed.

Service Messages

Service Messages are another class of error messages used in HP Color LaserJet. They usually indicate printer failure. However, some errors are clearable by pressing the ON LINE key, while others require powering off the printer. In general, if "SERVICE ERROR" is displayed on the second line, the error is continuable. If "ERROR" is displayed on the second line, the error is not continuable. If the error persists through a power cycle, service is required. Service Errors are not affected by the current value of AUTO CONTINUE.

The numeric portion of all service messages flashes at a 1 Hz rate. The alpha portion of all service messages are continuously displayed. The attention LED is lit for all of these messages (continuous or not).

```
50.X FUSER
   SERVICE ERROR
```

This error indicates the fuser terperature too high/low or the temperature sensor is open/shorted. Clear the error by power cycling the printer. If the error persists, service is required.

```
52.X ENG NVRAM    52.X ENG NVRAM
     ERROR           SERVICE ERROR
```

This message indicates the engine reported a NVRAM failure (not to be confused with the NVRAM on the formatter). Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
53 LASER          53 LASER
   ERROR             SERVICE ERROR
```

This messages indicates the engine reported a laser/scanner error. Attempt to clear the printer by power cycling the printer. If the error persists, service is required.

```
54.X ENGINE       54.X ENGINE
     ERROR           SERVICE ERROR
```

This message indicates the engine reported one of five types of general engine failures or errors. Attempt to clear the failures by power cycling the printer. If the error persists, service is required.

```
55.X ENGINE CMD   55.X ENGINE CMD
     ERROR           SERVICE ERROR
```

This message indicates a bad command was received from the engine. If the error persists, service is required.

```
56 LASER INDEX    56 LASER INDEX
   ERROR             SERVICE ERROR
```

This message occurs when toner accumulates on the scanner lens, resulting in the shut down of the main drive motor. Cleaning the scanner lens will correct this problem. If the error persists, service is required.

```
57 FAN MOTOR        57 FAN MOTOR
   ERROR               SERVICE ERROR
```

This message indicates that a cooling fan is either unplugged or dead. Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
59 MANUAL FEED
   SERVICE ERROR
```

This message indicates that either the manual feeder or Rear Feed Unit was removed/inserted while the power was on. Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
60.X SIMM           60.X SIMM
     ERROR             SERVICE ERROR
```

This message indicates that the firmware has detected a bad CRC in a SIMM (the afflicted SIMM is noted by X). The non-fatal version will occur at power up, and the bad SIMM will be ignored. If this error occurs after power up, the printer will display the fatal version and halt the printer.

```
61.X SIMM COMPAT    61.X SIMM COMPAT
     ERROR             SERVICE ERROR
```

This message indicates that there is faulty size or incompatible speed (timing) in a SIMM. Finding a SIMM with the correct size or speed will correct the problem. If this condition occurs at power up, the printer will not use the SIMM (all base RAM plus usable SIMM space will be used). If this condition occurs after power up, the SIMM is faulty and the fatal version will be displayed.

```
62.X INTERNL ROM
   SERVICE ERROR
```

This message indicates a checksum error in the internal ROM fonts. If X=0, the error is in the code ROMs. If X=1, the error is in the part A PCL font ROMs. If X=2, the error is in the part B PCL font ROMs. Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
63.0 DRAM
   SERVICE ERROR
```

This message indicates faulty onboard DRAM (base DRAM). Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
63.X SIMM          63.X SIMM
     ERROR            SERVICE ERROR
```

This message indicates a faulty SIMM expansion card. If this occurs at power up, this error can be continued. The printer will still use the base memory plus all of the memory other than the faulty location. If this occurs after power up, the error can be cleared by removing the faulty SIMM expansion card.

```
64.X DMA TIMEOUT
   SERVICE ERROR
```

This message indicates that the engine has detected a video direct memory access error. Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
65.X DEVELOPER     65.X DEVELOPER
     ERROR            SERVICE ERROR
```

This message indicates that either the black or color developer is not rotating. Attempt to clear the error by re-inserting the developer or by power cycling the printer. If the error persists, service is required.

```
66.X JAM SENSOR    66.X JAM SENSOR
     ERROR            SERVICE ERROR
```

This message indicates that either the drum winding sensor has malfunctioned or there was a drum calibration error. Attempt to clear the error by power cycling the printer. If the error persists, service is required.

```
68 NVRAM
   ERROR
```

This message indicates a recoverable error has been detected in NVRAM. Press the ON LINE key to clear the error, then verify the control panel settings; one or more fields may have been reset to their factory defaults during error recovery.

```
68 NVRAM
   SERVICE ERROR
```

This message indicates a NVRAM failure, requiring replacement of the part. Attempt to clear the error by power cycling the printer. The printer cannot be operated without replacement of the part.

```
68 COLD RESET
```

This message indicates that all control panel values have been reset to their factory defaults. This indicates that the printer is being powered-on for the very first time or if this message persists at each power-on sequence, it indicates that the NVRAM part is bad.

```
79 SERVICE(XXXX)
```

This message indicates the printer firmware has detected an unrecoverable error.   Attempt to clear the error by power cycling the printer. The XXXX indicates the error type. This number should be reported to your authorized service vendor. NOTE:

```
80 SERVICE(00XX)
```

This message indicates an unrecoverable Modular I/O (MIO) protocol error. Attempt to clear the error by power cycling the printer. This number should be reported to your authorized service vendor.

```
80 SERVICE(01XX)
```

This message indicates an indeterminate Modular I/O (MIO) error. The XX value can be found in the technical reference of the specific MIO interface. Attempt to clear the error by power cycling the printer. This number should be reported to your authorized service vendor.

```
80 SERVICE(02XX)
```

This message indicates a Modular I/O (MIO) self test failure. The XX value can be found in the technical reference of the specific MIO interface. Attempt to clear the error by power cycling the printer. This number should be reported to your authorized service vendor.

```
80 SERVICE(03XX)
```

This message indicates a fatal Modular I/O (MIO) error by an `Inquire I/O Status Response'. The XX value can be found in the technical reference of the specific MIO interface. This number should be reported to your authorized service vendor.

Attempt to clear the error by power cycling the printer.

```
80 SERVICE(04XX)
```

This message indicates a fatal Modular I/O (MIO) error by an `Asynchronous Status Request.' The XX value can be found in the technical reference of the specific MIO interface. This number should be reported to your authorized service vendor. Attempt to clear the error by power cycling the printer.

```
89 SERVICE(XXXX)
```

This message indicates that the PostScript firmware has encountered an unrecoverable error. Attempt to clear the error by power cycling the printer. The XXXX indicates the error type. This number should be reported to your authorized service vendor. Attempt to clear the error by power cycling the printer.

PostScript Error Messages

The following table provides a list of the PostScript 30 error messages which may appear on the control panel along with recommended actions. PostScript errors will cause the current print job to be canceled. Most 30 PostScript error messages are caused by driver incompatibilities, however some may be the result of data stream corruption or incorrect system settings. If you are sending PostScript files directly to the printer using the DOS "copy" command or a utility such as LaserTalk many of the below remedies will not apply. The actual cause of most PostScript errors are difficult to predict in advance and there will be times when the remedies described here will not be correct. In those cases you should call for assistance.

<u>**PostScript Note:**</u> The PostScript 30 error messages will not appear on the Control Panel if auto continue in the configuration menu is set to ON.

```
30 POSTSCRIPT
   ERROR ##
```

Press on line to resume operation if the following errors occur. The current job will be canceled. See the PostScript Language Reference Manual Second Edition pages 99 and 359 for more information on these errors.

Number	Description
00	dictfull - An attempt has been made to add an item to a dictionary that is already full. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
01	dictstackoverflow - An attempt to place too many dictionaries on the dictionary stack (Too many begins without corresponding ends). This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
02	dictstackunderflow - An attempt has been made to remove more dictionaries from the dictionary stack than were on the stack (more ends than begins).This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
03	execstackoverflow - The execution stack has grown too large; procedure invocation is nested deeper than PostScript permits. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
04	invalidaccess - An attempt has been made to access an array, dictionary, file, or string object in a way that violates its access attribute such as trying to write to a read only dictionary. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
05	invalidexit - No loop to exit from. An exit has been executed for which there is no dynamically enclosing looping construct. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
06	invalidfileaccess - The access string specification to the file operator is unacceptable.This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.

07	invalidfont - The operand to makefont or setfont is not a well formed font dictionary. This could be caused by the driver downloading bad font information or the font information being corrupted during the download. If you are experiencing other postscript errors that could be a result of data stream corruption check for problems with the connection between the printer and the host such as cabling, spoolers, sharing devices, or LANs. If no problems exist in the connection between the printer and the host try selecting a different PostScript printer driver and re-sending the print job.
08	invalidrestore - An improper restore has been attempted. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
09	ioerror - An error has occurred during the execution of one or more of the file operators such as attempting to write to a read only file. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
10	limitcheck - A PostScript implementation limit has been exceeded. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
11	nocurrentpoint - An operator requiring a current point has been executed while the current path is empty. This is frequently the result of graphics jobs being too complex. The driver should divide this job up to reduce the complexity. Try selecting a different PostScript printer driver and re-send the print job.
12	rangecheck - Operand out of bounds. A numeric operand's value is outside the range of acceptable operand values. This is frequently caused by a driver selecting a printer feature that is not supported on the printer such as attempting to duplex on a printer that does not have duplex capabilities. Check your printer setup or try selecting a different PostScript printer driver and re-send the print job.
13	stackoverflow - An attempt has been made to push too many objects on the operand stack. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
14	stackunderflow - An attempt has been made to pop an item from an empty operand stack. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
15	syntaxerror - The PostScript scanner has encountered text that does not conform to the PostScript syntax rules. This could be caused by the driver sending bad data to the printer the data being corrupted during the transmission to the printer. Another frequent cause of this error are PCL jobs, such as Banner Pages, being sent to PostScript, or extra characters being added to the data stream by spoolers or LANs. Be sure these features are disabled. If you are experiencing other POSTSCRIPT ERRORs that could be a result of data stream corruption check for problems with the connection between the printer and the host such as cabling, spoolers, sharing devices, or LANs. If no problems exist in the connection between the printer and the host try selecting a different PostScript printer driver and re-sending the print job.

16	Timeout - A time limit has been exceeded. The PostScript interpreter timed out waiting for the user to manually feed paper, waiting for data from the host, or because a job took too long to complete. This is frequently caused by extra characters such as end of file markers, or end of job markers being added to the job by spoolers, sharing devices, or LANs. Be sure that these features are disabled. Time limits can be selected by the user in most applications or the timeout feature can be completely disabled. If your application does not support setting or disabling the timeout feature you can disable it by sending a simple PostScript program to the printer. If you are not familiar with PostScript programming call for assistance.
17	typecheck - An operand has been encountered that is not the correct type such as a number is found when a string is required. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
18	undefined - A name has been encountered that cannot be found (has not been previously defined). This could be caused by the driver sending bad data to the printer or by the data being corrupted during transmission to the printer. Another frequent cause of this error are PCL jobs, such as Banner Pages, being sent to PostScript, or extra characters such as FORMFEED being added to the data stream by spoolers or LANs. Be sure these features are disabled. If you are experiencing other POSTSCRIPT ERRORs that could be a result of data stream corruption check for problems with the connection between the printer and the host such as cabling, spoolers, sharing devices, or LANs. If no problems exist in the connection between the printer and the host try selecting a different PostScript printer driver and re-sending the print job. (See Note #1 below).
19	undefinedfilename - A file identified by a name string operand of the file or run operators cannot be found or cannot be opened. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
20	undefinedresult - Overflow/underflow or meaningless result of a numerical calculation. Possible causes include division by zero or inverse transformation of a matrix that cannot be. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
21	unmatchedmark - A ']', cleartomark, or counttomark operator could not find the required mark on the stack. This is frequently a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job.
22	unregistered - An operator object has been executed for which the interpreter has no built in action. This represents an internal malfunction in the PostScript interpreter and should never occur. Call for assistance.
23	VMerror - An error has occurred in the virtual memory (VM) machinery. The most common problem is virtual memory is exhausted. Your printer does not have enough memory to print this job. Try reducing the number of fonts being downloaded to the printer or add memory to your printer.
24	quitexecuted - The systemdict quit operator was executed causing the PostScript interpreter to re-initialize virtual memory. All non-persistent information has been lost. NOTE This PS ERROR should never be displayed on the control panel. The only indication that it has occurred will be that the printer goes to 07 RESET and then back to 00 POSTSCRIPT READY and does not print the job. The PostScript printer driver issued a quit operator. This is not really an error but a warning that non-persistent information has been lost including soft fonts and header files.
25	internal - PostScript firmware failure. This should never occur. NOTE: This error is not listed in the PostScript Language Reference Manual. Call for assistance.

26	configurationerror - A setpagedevice request cannot be satisfied. This will most frequently occur if the PostScript printer driver requests a paper size or paper type that is not currently installed in the printer. Insure that the paper size or type you are selecting is installed in the printer or select a different paper size or type and re-send the job.
27	interrupt - An external interrupt request has been received by the PostScript interpreter. The driver or the application has generated an interrupt request such as a request for printer status. External interrupts are handled like an error. If this continues to happen try selecting a different Postscript printer driver.
28	undefinedresource - A named resource sought by the findresource operator does not exist. If this error occurs it could be a driver incompatibility problem. Try selecting a different PostScript printer driver and re-send the print job. See Note #1 below.

PostScript Note: Some Postscript jobs require a "header" file to be sent to the printer before the job (such as LaserPrep in the AppleTalk environment). These header files contain the definitions of the names used by the driver. Some drivers will assume that if they have previously sent down the header file they do not have to send it again for each subsequent print job. However for some reason, such as a system language switch between PCL and PostScript or another user downloading fonts or a different header file on top of yours, the header file is no longer in the printer. Some applications allow the user to require the header file be re-sent with each print job. If your application does not allow this then try exiting the application and restarting it. The header file will be sent to the printer before the first job.

PCL 79 SERVICE Codes

Code	Task	Description
0BFF	DG	Invalid opcode received while in normal state processing
1101	DG	General state error
0809	DM	Memory allocation error
080X	DM	Selftest message being returned to DM from the wrong task
081X	DM	Initialization message being returned to DM from the wrong task
082X	DM	A task failed initialization
083X	DM	A task failed to initialize in the required time
084X	DM	A task failed to initialize in the required time, and the task is not the current task initializing
085X	DM	A task failed self test
086X	DM	A task failed self test in the required time
087X	DM	A task failed self test in the required time, and the task is not the current self test task
088X	DM	Received a language switch acknowledgement from a task that was not asked.
08CF	DM	Character to convert does not appear in the futaba to roman8 symbol map
08D0	DM	Character to convert does not have a futaba equivalent
040A	EI	Engine command queue full
040B	EI	In illegal state when receiving status bytes from the formatter
040D	EI	PCL engine firmware receiving status for an unknown object
040E	EI	PCL engine firmware told to update status for an unknown object, and at the next check having a write in progress(contradiction).
040F	EI	EI requested undefined engine status
0410	EI	EI received status for an unknown variable
0411	EI	EI tried to print, in color, using the wrong media size
0412	EI	EI received a continuation message on an error that is not continuable
0413	EI	EI received a continuation message on an error that is not continuable (this variation used for fatal/non-fatal types)
0414	EI	EI received a continuation message on an undefined engine communication error
0415	EI	EI received a continuation message on an undefined error
0416	EI	EI sending data for a new page, when the previous page has not finished
0417	EI	Cyan plane does not have enough data
0418	EI	Black plane does not have enough data
0419	EI	Yellow plane does not have enough data
041A	EI	Magenta plane does not have enough data
041B	EI	EI detects a fatal error with the current page
041C	EI	EI detects a fatal error with the current page
041D	EI	Current page too small top margin
041D	EI	Image height (for current page size) - top margin < 1

041E	EI	Top margin + image length > total image length (for the current page size)
041E	EI	Top margin + image length does not fit on the physical page
041F	EI	Current page too small left margin
041F	EI	Left margin exceeds the maximum value
0420	EI	Left margin + image width > total image width (for the current page size)
0420	EI	Left margin + image width exceeds a maximum value
0421	EI	DMA color specified does not exist
0422	EI	The DMA channel is idle when setting up a DMA block
0423	EI	Current DMA block to send has an invalid number of bytes
0424	EI	EI received an unknown opcode
0425	EI	Status ready interrupt signal not functioning
0426	EI	CRDY signal asserted when trying to initialize engine communications
0427	EI	SRDY signal asserted when sending a command to the engine
0428	EI	DMA timeout error
0429	EI	EI received a video signal done, with the DMA flag not set
042A	EI	SRDY signal asserted when handling SRDY XISR
042B	EI	Image width (for current page size) - left margin < 1
042C	EI	EI received a know message, but not from the correct sender
042D	EI	EI receives an abort page command, when there is no page to abort
042E	EI	Hopper lock open when no toner low/out
042F	EI	DM asking EI to start processing a command that DM can't initiate
0430	EI	Invalid RET setting
04A0	EI	Engine service mode detects a key press, but it is not one of the pre-defined 8
04A1	EI	Engine detects problems in processing the last command
04A2	EI	Engine detects problems in processing the last command
0201	IOC	Memory allocation error
0201	IOC	Memory allocation error
0202	IOC	IOC invalid state
0203	IOC	IOC received an undefined opcode
0203	IOC	PIO received an inalid opcode
0204	IOC	Invalid opcode while in nop processing state
0205	IOC	Invalid opcode while in normal processing state
0206	IOC	MIO card requests invalid information
0207	IOC	IOC received an undefined sub-opcode
0208	IOC	MIO card sends an undefined command to IOC
0220	IOC	PIO detected a protocol error from other than sending or receiving
0264	IOC	MIO return information buffer overflow
0601	IP	IP gets another strip to process, but the strip is NULL
0602	IP	Not enough data in the next plane to process
0603	IP	IP given a NULL plane
0605	IP	A color plane has a color other than the four primaries (Yellow, Magenta, Cyan or Black)
0606	IP	An image plane is not 4 word aligned
0608	IP	Logical OR taking place with a scalar of 0 or a scalar not properly aligned
0609	IP	A raster block operation does not have a raster block supplied
0665	IP	Page intermediate processor detects a pattern mask not word aligned

06FF	IP	Invalid opcode processed by IP.
80A0	OS	Unexpected memory problems - V_BLOCK_OK
80A1	OS	Unexpected memory problems - A_PARAM_NUM_BYTES
80A2	OS	Unexpected memory problems - A_PARAM_POOL_ID
80B0	OS	Unexpected memory problems - CSP_PARAM_NUM_BYTES
80B1	OS	Unexpected memory problems - CSP_PARAM_POOL_ID
80B2	OS	Unexpected memory problems - DSP_PARAM_SUBPOOL_ID
80B3	OS	Unexpected memory problems - LARGEST_FREE_BLOCK_SIZE
80B4	OS	Unexpected memory problems - FL_BAD_FREE_BLOCK
80B5	OS	Unexpected memory problems - FB_BAD_FREE_BLOCK
80C0	OS	Unexpected memory problems - V_BAD_ADDR
80C1	OS	Unexpected memory problems - V_BAD_TOP_TAG
80C2	OS	Unexpected memory problems - V_BAD_NEXT_PTR
80C3	OS	Unexpected memory problems - V_BAD_PREV_PTR
80C4	OS	Unexpected memory problems - V_BAD_BOTTOM_TAG
80C5	OS	Unexpected memory problems - V_BAD_FILENAME
80C6	OS	Unexpected memory problems - V_BAD_LINENUM
80C7	OS	Unexpected memory problems - V_BAD_TIMESTAMP
80D0	OS	Unexpected memory problems - D_BAD_START_ADDR
80D1	OS	Unexpected memory problems - D_BAD_FREE_BLOCK
80D2	OS	Unexpected memory problems - D_NEXT_ALLOC_BLOCK_SIZE
80D3	OS	Unexpected memory problems - D_NEXT_TAG
80D4	OS	Unexpected memory problems - D_PREV_ALLOC_BLOCK_POOL_ID
80D5	OS	Unexpected memory problems - D_PREV_TAG
80E0	OS	Unexpected memory problems - R_PARAM_NUM_BYTES
80E1	OS	Unexpected memory problems - R_BAD_START_ADDR
80E2	OS	Unexpected memory problems - R_BAD_ALLOC_BLOCK
FD00	OS	Memory page pointer (to deallocate) is beyond the # range of all valid page addresses
FD01	OS	Memory page pointer (to deallocate) is not in the allocation list
FF00	OS	Unexpected trap received in the firmware.
0E63	PERS	This error occurs when the personality must terminate for unknown reasons.
0E71	PERS	Negative vector or termination coordinates
0E72	PERS	Vector not oriented top to bottom and width
0EFC	PERS	There is no more macro data, and the parser is NOT at the top state.
0EFD	PERS	There is no more macro data, and the next parser state is top state
0EXX	PERS	There is no more macro data, and the parser is at the top state. XX >= 0x80
050A	PG	PG was send a NULL page to process
050C	PG	Error in 2nd system initialization
050D	PG	Current strip has no color
050E	PG	No more temporary queues in which to compress the current image
050F	PG	PG task expected a message, and it came from the wrong sender
0510	PG	A particular strip of an image is not as wide as the others (from the same image)
0511	PG	Too many bitmap bands for a given image
0513	PG	PG tried to compute the toner mask of a NULL strip
0514	PG	NULL image request is being placed on the compress queue

0515	PG	A given color in an image structure does not have any allocated bands
0516	PG	Undefined color band bit field was set
0517	PG	PG task received an invalid opcode
0519	PG	Strip pointers were not cleared when another bitmap was to be processed
051A	PG	Can't test if in AIR (adaptive image rendering) mode
051B	PG	PG is working on more than one strip
051C	PG	Too many strips to compress for a given page
051D	PG	Current compression stop state not defined
051D	PG	An error occured after compressing the first row of an image
051E	PG	Too many compression overflow bytes
051F	PG	Too small of an image was being compressed
0520	PG	First strip of an image to compress is NULL
010C	PS	PS received an I/O clear complete (used in resets), when it did not expect one
010E	PS	PS received an inquire acknowledgement when it was not expecting one
0113	PS	Initialization error
0115	PS	PS received a wait acknowledgement when it was not expecting one
0117	PS	PS initialized OK, but detected an error
0118	PS	PS received a read complete message when not waiting for a read complete
0119	PS	Undefined state
0145	PS	A block of memory is being moved with negative size
0146	PS	A block of memory is being deallocated with negative size
0147	PS	This error occurs when pslibgen (the personality support library) detects a general error.
0151	PS	AIR processor detects a strip has already been rendered
0152	PS	AIR processor could not terminate a strip correctly
0153	PS	AIR processor has been sent an empty strip
0154	PS	AIR processor has been sent intermediate in the wrong state
0155	PS	Target intermediate does not have the correct state (CLOSED_FRAME_STATE).
0156	PS	AIR processor could not allocate enough memory to initiate AIR
0157	PS	Strip intermediate processor can't allocate necessary memory
0158	PS	Page intermediate processor found a NULL strip
0159	PS	Strip initializer did not find a strip
015A	PS	Invalid STATE opcode in psl_process_bucket_state
015B	PS	Invalid STATE opcode in psl_process_bucket_state
015C	PS	Invalid DATA opcode in psl_process_data_intermediate()
015D	PS	Invalid opcode, in AIR mode, in psl_process_data_intermediate
015E	PS	Invalid opcode in psl_process_data_intermediate
015F	PS	Invalid STATE opcode in psl_process_state_intermediate
0160	PS	Invalid opcode (while in AIR mode) in psl_process_state_intermediate
0161	PS	Invalid opcode in psl_process_state_intermediate
0162	PS	Bucket state processor received a pattern command when it is invalid to do so
0163	PS	Bucket state processor received a color command when this command is not valid.
0164	PS	Page intermediate processor detects a pattern not word aligned

0166	PS	Page intermediate processor detects a color image not word aligned
0180	PS	Color Pipeline not being able to access the current palette.
0180	PS	Cannot push raster values into black & white reference ranges
01BB	PS	Unsuccessful in creating a DIC color halftone
01BC	PS	General Color Pipeline Error (i.e. palette, user defined matricies, setting render algorithm, setting viewing illuminant, set finish mode, processing CIELab).
01BC	PS	Removed a pallette that was used by the DIC color pipeline
01FF	PS	Invalid opcode processed by PS.
8105	PS	The personality wants to switch to the background task when it does not exist (the background task must be created).
8106	PS	The personality wants to switch from the background task to the background task (only FG => BG and BG => FG).
0D99	PSLIB	Cannot allocate memory for pushing back I/O data
0900	SD	SD cannot allocate a queue position for a pending page
0901	SD	SD received an invalid opcode.
0906	SD	SD can't use the desired finish (glossy or matte)
0907	SD	SD cannot satisfy the desired paper size
0908	SD	SD cannot satisfy the desired output tray
0909	SD	SD detects the desired source tray is invalid
090B	SD	SD wants to initiate printing of a page, and there is not a page pending
090D	SD	Paper mount override is executed from a non-defined input tray
090E	SD	SD wants to remove a page from the pending queue but there is not one
092C	SD	SD received a particular message from an invalid task
7X01	XX	Unexpected error during NVRAM get/put - Bad feature during get_default()
7X02	XX	Unexpected error during NVRAM get/put - Bad feature during get()
7X03	XX	Unexpected error during NVRAM get/put - Bad feature during put()
7X04	XX	Unexpected error during NVRAM get/put - Invalid index
7X05	XX	Unexpected error during NVRAM get/put - Invalid byte ID
7X06	XX	Unexpected error during NVRAM get/put - Queue overflow
8X01	XX	OS is not able to allocate a new MCB for a task asking for it
8X02	XX	OS is not able to allocate a new MCB for a task asking for it
8X03	XX	OS has recorded too many keypresses (caused by the keypresses not being recognized.
8X04	XX	OS has recorded information for too many font cartridges
8X07	XX	A task is calling OS kernel routines when the firmware is currently in a C trap handler.
XXAA	XX	Problems attaching a particular queue element.

80 SERVICE Codes (00XX)

Code	Task	Description
0002	MIO	Unexpected buffer configuration request.
0003	MIO	I/O card does not support host-to-peripheral transfers.
0005	MIO	I/O card sent a pgp packet that was never sent as a new pgp packet.
0006	MIO	TM_OUT or CLR_RSP bits were not set after waiting the 2 second initialization time.
0007	MIO	TM_OUT or CLR_RSP bits were set when not in selftest.
0009	MIO	I/O card did not respond to pgp request in specified time.
0019	MIO	I/O card did not complete buffer configuration within 10 seconds of the peripheral clearing the CLR_RSP bit.
001F	MIO	I/O card did not clear the IMB_VLD bit in specified time.
0020	MIO	I/O card did not set the CLR_RSP bit within the original S seconds+1.
0033	MIO	Unknown read packet return status.
0034	MIO	Unknown write packet return status.
0036	MIO	Unknown ascii_io_configuration return status.
0037	MIO	Unknown inquire_io_status return status.
0039	MIO	Unknown io_description return status.
003A	MIO	Unknown set_peripheral_display return status.
003F	MIO	Unknown write_retrieve return status.
0040	MIO	Unknown ascii_status return status.
0041	MIO	Unknown set_io_configuration return status.
0042	MIO	Unknown channel_description return status.
0050	MIO	Invalid length response after ascii_io_configuration request.
0051	MIO	Invalid length response after identify request.
0052	MIO	Invalid length response after async_io_status request.
0053	MIO	Invalid length response after inquire_io_status request.
0054	MIO	Unexpected read packet received from I/O card which was not part of agreed configuration.
0055	MIO	io_description string not null terminated or length zero or length > 33 characters.
0057	MIO	1st ascii_io_configuration string not null terminated or length zero or length > 13 characters.
0058	MIO	2nd ascii_io_configuration string not null terminated or length zero or length of 1st+2nd string is > 14 characters.
005D	MIO	I/O card MIO protocol revision is less than 4.0.
005E	MIO	io_description did not understand command modifiers of 0 & 1.

005F	MIO	Unexpected write packet received from I/O card which was not part of agreed configuration.
0060	MIO	I/O card said it was a bi-directional link, but has not given the peripheral any WRITE_DATA packets before the switch to PostScript.
0061	MIO	I/O card sent a set_io_configuration item that was never sent.
0062	MIO	set_io_configuration item string was not null terminated or length zero or length > 24 characters.
0063	MIO	set_io_configuration item string was not null terminated or length zero or length > 24 characters.
0064	MIO	set_io_configuration value+item exceeds min pgp buffer size

0065	MIO	new packet max data len > min pgp buffer
0066	MIO	long channel description string not null terminated, longer than packet data length, or zero length
0067	MIO	short channel description string not null terminated, longer than 10 characters, or zero length
01XX	MIO	Byte #5, peripheral mailbox error byte, was non zero. The "XX" will be replaced by the hex digits that represent the I/O card error.
03XX	MIO	Buffer byte #1 in the response packet for an inquire_io_status request is not zero. The "XX" will be replaced by the hex digits that represent the I/O card error.
04XX	MIO	Buffer byte #1 in the response packet for an async_io_status request is not zero. The "XX" will be replaced by the hex digits that represent the I/O card error.

PostScript 89 SERVICE Codes

If your printer recieves one of these errors call Color LaserJet Developers Support at (208)-396-3226.

Factory Defaults

I.1 PCL Print Menu

- COPIES: 1
- FONT SOURCE: INT (Internal)
- FONT NUMBER: 0
- PITCH: 10.00
- PT. SIZE: 12.00
- TRAY: FRONT
- PAPER: LETTER
- OUT BIN: UPPER
- ORIENTATION: P (Portrait)
- 60 LINES

  VMI: 8
- MANUAL FEED: OFF
- SYMBOL SET: ROMAN-8

I.2 PostScript Menu

- COPIES: 1
- TRAY: FRONT
- FT MEDIA: PAPER
- RT MEDIA: PAPER
- MF MEDIA: PAPER
- OUT BIN: UPPER

I.3 PCL Config Menu

- SYS: AUTO.
- AUTO CONTINUE: OFF
- TIMEOUT: 15
- RET: MEDIUM

- PWRSAVE: 60MIN

I.4 Postscript Config Menu

- SYS: PCL
- AUTO CONTINUE: OFF
- TIMEOUT: 15
- RET: MEDIUM
- PWRSAVE: 60MIN
- START PAGE: OFF
- JAM RECOVERY: OFF
- POSTSCRIPT ERROR PRINTING: OFF

I.5 MIO Config Menu (PCL and Postscript)

I/O: As determined by the installed MIO interface.

I.6 Default Display Language (PCL and Postscript)

DISPLAY LANGUAGE: ENGLISH

Print Menu ConfigMessages

The following list contains the (English) Print Menu messages in the order in which they are displayed when a user steps through the PCL PRINT MENU using the ITEM key. The values in square brackets are the available choices for each configurable item.

`COPIES=[1] [2] ...[999]'

The value field for this item can be up to a three (3) digit, left-justified number. The factory default for this item is 1.

`FONT SOURCE=[INT] [TOP] [BTM] [SFT] [M1] [M2] [M3] [M4]'

Only font sources which are included in the FONT SAMPLE PRINTOUT are selectable. The factory default for this item is INT.

`FONT NUMBER=[0] [1] [2]...[999]'

Only font numbers which are included in the FONT SAMPLE PRINTOUT (up to and including 999) are selectable for this item. The font numbers of the FONT SAMPLE PRINTOUT may exceed 999. The value field for this item is a one to three digit, left-justified number. The factory default for this item is 0 (zero) which is a portrait, 10-pitch, 12-point, upright, medium, Courier font.

`PITCH=[0.44] [0.45] [0.46]...[99.97] [99.98]...[99.99]'

This message only appears in the display if the font selected by the font source and font number is a scalable, fixed spaced, font. The value field for this item is a three or four digit, left-justified number with two decimal places. The default is 10.00 pitch.

`PT. SIZE=[4.00] [4.25] [4.50] [4.75] [5.00]...[999.75]'

This message only appears in the display if the font selected by the font source and font number is a scalable, proportional spaced, font. The value field for this item is a three to five digit, left-justified number with two decimal places. The default is 12.00 point.

`TRAY=[FRONT] [REAR] [BOTH]'

The factory default for this item is FRONT.

`PAPER=[LETTER] [LEGAL] [A4] [EXEC] [TABLOID] [A3]'

The factory default for this item is LETTER

`OUT BIN=[UPPER] [LOWER]'

The factory default for this item is UPPER.

`ORIENTATION=[P] [L]'

The factory default for this item is P (Portrait).

`FORM=[5] [6] [7]...[128] LINES'

The minimum Print Menu default vertical form length which may be selected is 5 lines. The maximum Print Menu default vertical form length which may be selected is 128 lines. If, because of a subsequent change in media size, the value the printer calculates to display is greater than 128 lines, the printer limits the form length to 128 lines and recalculates the Print Menu default VMI based on the current media size and a 128 line form length. If, because of a subsequent change in media size, the value the printer calculates to display is less than 5, the printer limits the form length to 5 and recalculates the Print Menu default VMI based on the current media size and a 5 line form length. The factory default for this item is VMI=8 (6 lines/inch).

`MANUAL FEED=[ON] [OFF]'

The factory default for this item is OFF.

`SYM SET=[ROMAN-8] [ISO L1] [ISO L2] [ISO L5] [PC-8] [PC-8 DN][PC-850] [PC-852]

[PC-8 TK] [WIN L1] [WIN L2] [WIN L5] [DESKTOP][PS TEXT] [VN INTL] [VN US] [MS

PUBL] [MATH-8] [PS MATH][VN MATH] [PI FONT] [LEGAL] [ISO-4] [ISO-6] [ISO-11]

[ISO-15][ISO-17] [ISO-21] [ISO-60] [ISO-69] [WIN 3.0]'

Refer to the 'AUTOMATIC ISO SUBSTITUTION' document dated February 7, 1986 for a definition of the numbers used in the preceding list. The factory default for this item is ROMAN-8.

The following list contains the (English) Print Menu configuration messages in the order in which they are displayed when a user steps through the PostScript Print Menu using the ITEM key. The values in square brackets are the available choices for each configurable item.

`COPIES=[1] [2] ...[999]'

The value field for this item can be up to a three (3) digit, left-justified number. The factory default for this item is 1.

`TRAY=[FRONT] [REAR] [BOTH]'

The factory default for this item is FRONT.

`FT MEDIA=[PAPER] [TRN]'

The factory default for this item is PAPER.

`RT MEDIA=[PAPER] [TRN]'

The factory default for this item is PAPER.

`MF MEDIA=[PAPER] [TRN]'

The factory default for this item is PAPER.

`OUT BIN=[UPPER] [LOWER]'

The factory default for this item is UPPER.

Config Menu Configuration Messages

The following list contains the device configuration messages in the order in which they are displayed when a user steps through the PCL Config Menu using the ITEM key. The values in square brackets are the available choices for each configurable item.

`SYS=[AUTO] [PCL] [POSTSCRIPT]'

The factory default for this item is AUTO.

Note: The AUTO selection will only appear if there is more than one (1) personality installed.

`AUTO CONT=[ON] [OFF]'

The factory default for this item is ON.

`TIMEOUT=[5] [6] . . . [300]'

The factory default for this item is 15.

`RET=[DARK] [MEDIUM] [LIGHT] [OFF]'

The factory default for this item is MEDIUM.

`PWRSAVE=[15MIN] [30MIN] [60MIN] [2HRS] [3HRS]'

The factory default for this item is 60MIN.

The following list contains the device configuration messages in the order in which they are displayed when a user steps through the PostScript Config Menu using the ITEM key. The values in square brackets are the available choices for each configurable item.

`SYS=[AUTO][PCL] [POSTSCRIPT]'

The factory default for this item is AUTO.

`AUTO CONT=[ON] [OFF]'

The factory default for this item is ON.

`TIMEOUT=[5] [6] . . . [300]'

The factory default for this item is 15.

`RET=[ON] [OFF]'

The factory default for this item is MEDIUM.

`PWRSAVE=[15MIN] [30MIN] [60MIN] [2HRS] [3HRS]'

The factory default for this item is 60MIN.

`JAM RECOVER=[ON] [OFF]'

The factory default for this item is OFF.

`START PAGE=[ON] [OFF]'

The factory default for this item is ON.

`PRT PS ERRS=[ON] [OFF]'

The factory default for this item is OFF.

Parallel Menu Configuration Messages

The following list contains the parallel menu messages in the order which they are displayed when a user steps through the parallel menu using the ITEM key. The values in square brackets are the available choices for each configurable item.

`HIGH SPEED=[NO] [YES]'

The factory default for this item is YES.

`ADV FNCTNS=[NO] [YES]'

The factory default for this item is YES.

PCL Message Priorities

Each displayed message has an importance (or priority) associated with it. Priority 1 messages are displayed regardless of printer state (while larger priority message communicate status only). Device status messages can easily be overridden by device error or service error messages.

Messages at the same priority level do not have rank or order within themselves. However, messages with the same priority CAN replace eachother.

Note: You may notice that device status messages most often have low priority (denoted by a large number). Device service errors have a high priority (denoted by a low number).

Priority	Message	Description
1	`79 SERVICE`	Unrecoverable firmware error
1	`80 SERVICE (00XX)`	Unrecoverable MIO error
1	`80 SERVICE (01XX)`	Indeterminate MIO error
1	`80 SERVICE (02XX)`	MIO self test failure
1	`80 SERVICE (03XX)`	Fatal MIO response by an 'Inquire I/O Status Response'
	`80 SERVICE (04XX)`	Fatal MIO error by an 'Asynchronous Status Request'
1	`68 COLD RESET`	All control panel defaults have been reset to factory defaults

2	`FE FONT CART ERR`
	`CYCLE POWER`
		A font cartridge was removed while the printer was on line

3	`07 RESET`	Displayed during a control panel reset
3	`09 MENU RESET`	Displayed during a control panel menu reset
4	`12 CLOSE TOP OR SIDE DOOR`	Printer senses that the top or side door is open
5	`52.X ENG NVRAM SERVICE ERROR`	NVRAM failure
5	`53 LASER SERVICE ERROR`	Fatal laser/scanner error
5	`54.X ENGINE SERVICE ERROR`	Fatal general engine failure
5	`55.X ENGINE CMD SERVICE ERROR`	Bad command was received from the engine and the engine can not continue

5	56 LASER INDEX SERVICE ERROR	A dirty scanner lens is preventing the main drive motor from operation
5	57 FAN MOTOR SERVICE ERROR	The cooling fan is either unplugged or not operational
5	58.X PWR SUPPLY SERVICE ERROR	Indicates excessive voltage discharge
5	59 MANUAL FEED SERVICE ERROR	Manual feeder or RFU was removed while printer was on
5	60.X SIMM SERVICE ERROR	A SIMM has a bad CRC and the printer can not be used
5	61.X SIMM COMPAT SERVICE ERROR	A SIMM is incompatible or has inappropriate timing. The printer can not be used
5	62.X INTERNL ROM SERVICE ERROR	Indicates a checksum error in the internal ROM fonts
5	63.0 DRAM SERVICE ERROR	The printer has bad DRAM
5	63.X SIMM SERVICE ERROR	A SIMM has a bad expansion card. The printer can not be used
5	64.X DMA TIMEOUT SERVICE ERROR	The engine has detected a video DMA error
5	65.X DEVELOPER SERVICE ERROR	The black or color (CMY) developer is not rotating
5	66.X JAM SENSOR SERVICE ERROR	The drum winding sensor has malfunctioned or there was a drum calibration error
5	68 NVRAM SERVICE ERROR	A NVRAM failure occurred, requiring replacement of the part
6	53 LASER ERROR	Non-fatal laser/scanner error
6	54.X ENGINE ERROR	Non-fatal general engine failure
6	55.X ENGINE CMD ERROR	Bad command was received from the engine and the engine can not continue
6	60.X SIMM ERROR	A SIMM has a bad CRC, but the printer can be used

6	61.X SIMM COMPAT ERROR	A SIMM is incompatible or has inappropriate timing. The affected areas are mapped out
6	63.X SIMM ERROR	A SIMM has a bad expansion card. The affected areas are mapped out
6	68 NVRAM ERROR	An unrecoverable error has occurred with NVRAM
7	17.1 INSTALL BLACK DEVELOPER	The black developer is not installed or not correctly installed
7	17.2 INSTALL COLOR DEVELOPER	The color developer is not installed or not correctly installed
7	17.4 INSTALL FUSER	The fuser is not installed or not correctly installed
7	17.5 INSTALL TRANSFER BELT	The transfer belt is not installed or not correctly installed
7	17.6 INSTALL COLLECTION BOX	The toner collection box or drum is not installed or not correctly installed
7	17.7 REMOVE DRUM COVER	The protective drum cover (used in shipping) is still on the drum
7	17.8 REINSTALL DRUM	
7	17.9 REINSTALL FUSER	The drum is not properly seated (the electrical connections) or the drum winding sensor is broken
7	18.1 CLEAN TRANS CORONA WIRE	The printer experienced a high voltage leak. Opening the door will reset the power supply latch (and you might as well do something constructive in the meantime).
7	18.2 CLEAN DRUM CORONA WIRE	The printer experienced a high voltage leak. Opening the door will reset the power supply latch (and you might as well do something constructive in the meantime).
7	18.3 REINSERT DEVELOPERS	The printer experienced a high voltage leak. Opening the door will reset the power supply latch (and you might as well do something constructive in the meantime).
8	TONER OUT KCMY ADD BLACK	The black toner hopper needs refilling
8	TONER OUT CMY ADD CYAN	The cyan toner hopper needs refilling
8	TONER OUT MY ADD MAGENTA	The magenta toner hopper needs refilling

8	TONER OUT Y ADD YELLOW	The yellow toner hopper needs refilling
8	TONER OUT OPENING INTRLOCK	The printer senses toner out, but the appropriate lock is not open
9	PRESS ON LINE TO CONTINUE	The printer wants to install the new developer(s), but must first wait for the ONLINE keypress
10	INSTALLING NEW DEVELOPER /	The printer is installing the new developer(s).
11	NEW DRUM INST. DRUM NUMBER=XXX	The printer wants the drum calibration number, indicated by a sticker on the front of the drum
11	PLEASE VERIFY DRUM NUMBER=XXX	The printer wants verification of the drum number input by the previous message
12	10 PRESS RESET TO ACTIVATE	The PCL or PS print menu has changed while the printer is processing data
13	13.1 CLEAR DRUM WINDING JAM	The printer senses media jammed near the drum
13	13.2 CLEAR OUTPUT JAM	The printer senses media jammed near the exit assembly
13	13.3 CLEAR MANUAL JAM	The printer senses media jammed near the manual feed input
13	13.4 CLEAR FRONT TRAY INPUT JAM	The printer senses media jammed near the front tray
13	13.5 FUSER JAM	The printer senses media jammed near the fuser
13	13.6 CLEAR REAR TRAY INPUT JAM	The printer senses media jammed near the rear tray
13	13.7 CLEAR REAR TRAY PAPER JAM	The printer senses media jammed between the rear tray and the drum
13	13.8 CLEAR PAPER JAM	The printer senses media jammed somewhere in the printer (or more than two sensors detect a media jam)
14	REINSERT TOP FONT CARTRIDGE	The top font cartridge was removed while the printer was offline but processing data
14	REINSERT BOTTOM FONT CARTRIDGE	The bottom font cartridge was removed while the printer was offline but processing data
14	REINSERT BOTH FONT CARTRIDGES	Both font cartridges were removed while the printer was offline but processing data
15	21 MEMORY OUT	The engine has run out of memory

16	20 MEMORY OVERFLOW	The printer has run out of memory
17	40 I/O DATA ERROR	A data error has occurred (parity, framing or line overrun)
18	15 PRINTING ENGINE TEST	The printer is printing the special engine test page
19	02 PCL WARMING UP	The printer's fuser is reaching operating temperature (while the current language is PCL)
20	14.5 REPLACE COLLECTION KIT	Members of the collection box kit are old and need replaced
20	14.6 REPLACE COATING PAD KIT	Members of the coating (oil) pad kit are old and need replaced
20	19.1 BLACK DEVEL USER MAINTENANCE	The black developer is at the end of its specified life (19.1 flashes)
20	19.2 COLOR DEVEL USER MAINTENANCE	The color developer is at the end of its specified life (19.2 flashes)
20	19.3 DRUM USER MAINTENANCE	The drum is at the end of its specified life (19.3 flashes)
20	19.4 FUSER USER MAINTENANCE	The fuser is at the end of its specified life (19.4 flashes)
20	19.5 TRANS BELT USER MAINTENANCE	The transfer assembly is at the end of its specified life (19.5 flashes)
21	11.1 FRONT TRAY LOAD LETTER	The printer needs letter paper loaded into the front tray
21	11.1 FRONT TRAY LOAD EXEC	The printer needs exec paper loaded into the front tray
21	11.1 FRONT TRAY LOAD A3	The printer needs A3 paper loaded into the front tray
21	11.1 FRONT TRAY LOAD A4	The printer needs A4 paper loaded into the front tray
21	11.1 FRONT TRAY LOAD LEGAL	The printer needs legal paper loaded into the front tray
21	11.1 FRONT TRAY LOAD TABLOID	The printer needs tabloid paper loaded into the front tray
21	11.2 REAR TRAY LOAD LETTER	The printer needs letter paper loaded into the rear tray
21	11.2 REAR TRAY LOAD A4	The printer needs A4 paper loaded into the rear tray

21	11.2 REAR TRAY LOAD EXEC	The printer needs exec paper loaded into the rear tray
21	11.3 MANUAL FEED LOAD LETTER	The printer needs letter paper loaded into the manual feeder
21	11.3 MANUAL FEED LOAD EXEC	The printer needs exec paper loaded into the manual feeder
21	11.3 MANUAL FEED LOAD A3	The printer needs A3 paper loaded into the manual feeder
21	11.3 MANUAL FEED LOAD A4	The printer needs A4 paper loaded into the manual feeder
21	11.3 MANUAL FEED LOAD LEGAL	The printer needs legal paper loaded into the manual feeder
21	11.3 MANUAL FEED LOAD TABLOID	The printer needs tabloid paper loaded into the manual feeder
22	22 I/O CONFIG ERROR	The printer's receiving buffer has overflowed
23	04 CONTINUOUS SELF TEST	The printer is in a continuous self test mode
23	06 PRINTING PCL SELF TEST	The printer is printing a PCL self test
23	06 PRINTING PCL TYPEFACES	The printer is printing a PCL typeface (font) list
23	06 PRINTING PCL DEMO PAGE	The printer is printing a PCL demo page
24	LJ4C DEMO PAGE PRESS ANY KEY	The printer is in demonstration mode
25	23 MIO NOT READY	The MIO card is unable to accept data
26	21 MIO INITIALIZING	The MIO card is unable to accept data because it is initializing
27	48 INVALID JOB ABORTING JOB	The printer received data that cannot be processed by any personality
28	NO FONTS TOP FONT CARTRIDGE	There are no usable fonts in the cartridge currently inserted into the top slot
28	NO FONTS BOTTOM FONT CARTRIDGE	There are no usable fonts in the cartridge currently inserted into the bottom slot
28	NO FONTS BOTH FONT CARTRIDGES	There are no usable fonts in both cartridges currently installed in both slots
29	TONER LOW OFF LINE TO FILL	The printer is low on toner, but toner cannot be added unless the printer is off line
29	TONER LOW OPENING INTRLOCK	The printer is low on toner and offline, but the appropriate hopper interlock is not open

29	TONER LOW KCMY ADD BLACK	The printer is low on black toner
29	TONER LOW CMY ADD CYAN	The printer is low on cyan toner
29	TONER LOW MY ADD MAGENTA	The printer is low on magenta toner
29	TONER LOW Y ADD YELLOW	The printer is low on yellow toner
30	16.5 REPLACE COLLECTION KIT	Members of the collection box kit are nearing life and need replaced
30	16.6 REPLACE COATING PAD KIT	Members of the coating (oil) pad kit are nearing life and need replaced
30	19.1 BLACK DEVEL USER MAINTENANCE	The black developer is past the end of specified life (19.1 not flashing)
30	19.2 COLOR DEVEL USER MAINTENANCE	The color developer is past the end of specified life (19.2 not flashing)
30	19.3 DRUM USER MAINTENANCE	The drum is past the end of specified life (19.3 not flashing)
30	19.4 FUSER USER MAINTENANCE	The fuser is past the end of specified life (19.4 not flashing)
30	19.5 TRANS BELT USER MAINTENANCE	The transfer assembly is pas the end of specified life (19.5 not flashing)
31	11.4 FRONT TRAY EMPTY	The front paper tray is either empty or removed
31	11.5 REAR TRAY EMPTY	The rear paper tray is empty
32	05 INTERNAL SELF TEST	The printer is executing an internal test
32	INITALIZING NVRAM	The printer's NVRAM is being re-initialized with factory defaults
32	CONFIG LANGUAGE	The user is entering the language configuration mode
32	ERROR LOG	The user is entering the error log mode
32	DEMO MODE	The user is entering the demonstration mode
32	SERVICE MODE	The user is entering the service mode

32	EXTENDED DIAGNOSTICS MODE	The user is entering the extended diagnostics mode
32	08 COLD RESET	The user executed a cold reset, resetting all user defaults in the print menu to factory default
32	24 BUSY MOVING TONER	The engine is adjusting toner in the pipes from the hoppers to developers
33	00 PCL READY	The printer is on line and ready to process data (in PCL)
33	POWERSAVE MODE	The printer is in power conservation mode

INDEX