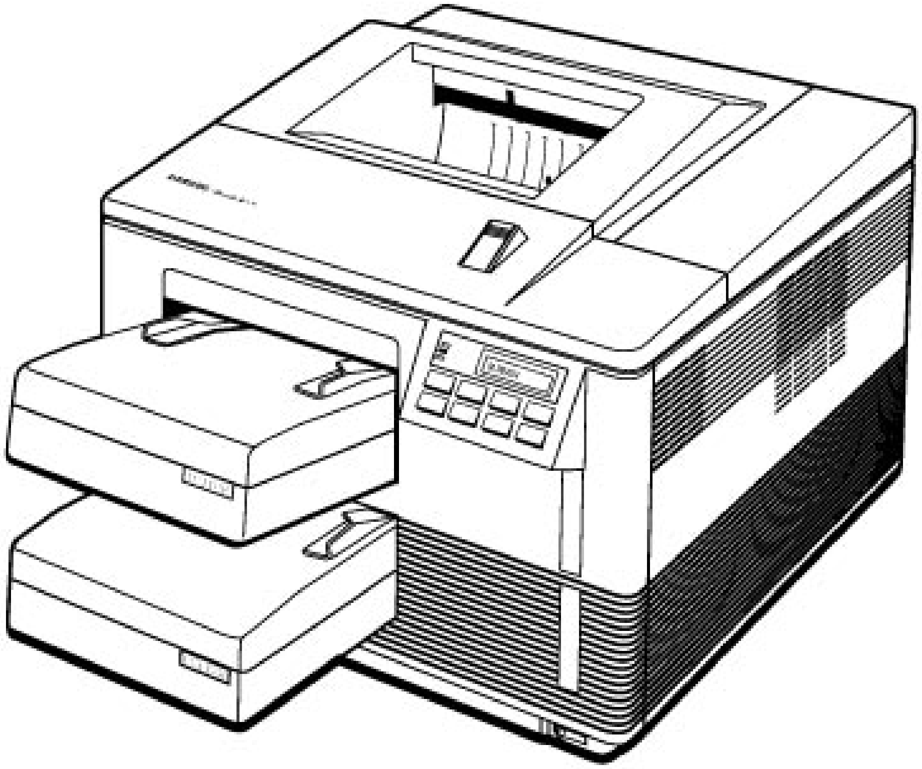
HP LaserJet 4Si Printer
Developer's Quick Reference Guide



HP LaserJet 4Si Mx Printer

Edition 1
04/22/95 01:51 PM

Printed in U.S.A.

Hewlett-Packard

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

The information contained in this document is subject to change without notice.

Printing History

This manual was created using Word for Windows 2.0 software on an HP Vectra Personal Computer. The camera-ready copy was printed on an HP LaserJet 4Si printer and was then reproduced using standard photocopying equipment.

Printed in U.S.A.

First Edition April, 95

Conventions

The following conventions are used throughout this manual:

Bold is used to emphasize important information in the text or to highlight text to be entered from your computer's keyboard.

Italic type is used for titles that refer to manuals or documents, to indicate selections to be made, or for emphasis.

Note ☞ Notes contain important information that is set off from the text.

Inside This Manual

This manual is written to assist you in developing software drivers for the HP LaserJet 4Si printer.

This manual is written for developers who are familiar with Hewlett-Packard's PCL printer language and Adobe's PostScript language up to the level of detail described in one or more of the following manuals:

PCL 5 Printer Language Printer Technical Reference Manual.

PCL 5 Comparison Guide .

PostScript Language Reference Manual, Addison-Wesley Publishing Company, Inc. (ISBN 0-201-10169-6). Also known as the "Red Book".

An understanding of the information provided in these reference manuals, and a familiarity with the LaserJet III printer, will prepare the reader for many of the development needs of the LaserJet 4Si printer. Refer to the PCL 5 Printer Language Technical Reference Manual (PN 5961-0509) to answer questions pertaining to function or terminology.

Support

Hewlett-Packard provides on-line telephone support for developers at the following numbers:

♣ Developers Support (208) 323-3226

You may also use HP Peripherals PowerLine, HP's developer support service on CompuServe. PowerLine provides a forum for getting answers to your questions and file exchange as well as a library containing up-to-date copies of this document, PPD files, etc. It also provides InfoBase, a keyword searched text data base with up-to-date information. If you are already a member of CompuServe, but not a member of PowerLine, you can join PowerLine by sending a CompuServe mail message to 76702,1503 with the following information:

Name
Company Name
Phone Number
CompuServe User ID

If you would like to join PowerLine, but are not currently a member of CompuServe, you can call HP at (619) 592-8308 to receive a free introductory CompuServe membership.

For general PostScript support, please use the Adobe Developer's Association.

Table of Contents

LaserJet 4Si Differences

The following is a brief abstract for those of you who are familiar with the HP LaserJet IIISi & HP LaserJet 4 printer. This section will highlight differences between the three printers.

The LaserJet 4Si control panel consists of eight keys, five Light Emitting Diodes (LEDs), and a 16-character Liquid Crystal Display (LCD).

Features new to LaserJet 4Si (since the LaserJet Series IIISi) include context sensitive personality switching, I/O switching, PostScript level 2, 45 built-in typefaces, status readback, true end-of-job notification, Environment Saving, and job pipelining.

The standard LaserJet 4Si printer has a built in Bi-Directional parallel interface. It also includes two Modular Input/Output (MIO) slots that may be populated with any of the available JetDirect cards or MIO interfaces. LaserJet 4Si supports two paper input trays. It also supports optional Envelope Feeder and an optional duplexer.

Backward Compatibility

The LaserJet 4Si printer is closely compatible with the LaserJet 4 printer product. It is to be noted, however, that the LaserJet 4 has made major enhancements over features in the LaserJet III and IIID product lines.

Feature Comparison Matrix

The following table provides a comparison of LaserJet 4Si features with those of Hewlett-Packard's LaserJet IIISi & LaserJet 4 printers.

Feature	**LaserJet 4Si**	**LaserJet 4**	**LaserJet IIISi**
Printer Language	PCL 5 (with enhancements)	PCL 5 (with enhancements)	PCL 5
Print speed	Up to 17 ppm	Up to 8 ppm	Up to 17 ppm
Text resolution	600x600 DPI (300x300 DPI selectable)	600x600 DPI (300x300 DPI selectable)	300x300 DPI
Resolution Enhancement technology	Yes ON or OFF	Yes OFF,LIGHT,MEDIUM or DARK	Yes OFF,LIGHT,MEDIUM or DARK
Graphics resolution with - standard memory			
75 DPI	1 page	1 page	1 page
100 DPI	1 page	1 page	1 page
150 DPI	1 page	1 page	1 page
200 DPI	1 page	1 page	N/A
300 DPI	1 page	1 page	~ 56 sq. in.
600 DPI	1 page**	1 page**	N/A
Minimum add-on memory required to print full-page 300 DPI graphics	No additional memory required for full page 300 DPI images	No additional memory required for full page 300 DPI images	1 MB memory
Paper Sizes Supported	A4,Letter,Legal,Executive Com10,Monarch,DL.	A4,Letter,Legal,Executive Com10,Monarch,DL,B5,C5	A4,Letter,Legal,Exec Com10,Monarch,DL.
Number of input trays	2 standard (3 with optional envelope feeder installed)	2 standard (4 with optional lower tray and optional envelope feeder installed)	2 standard (3 with optional envelope feeder installed)
Input tray capacity (sheets)	Upper Tray - 500 Lower Tray - 500 Envelope Feeder - 100 Option Feeder - 1500	MP tray - 100 Paper Cassette - 250 Lower Cassette - 500 Option Envelope - 100	Upper Tray - 500 Lower Tray - 500 Envelope Feeder - 100 Option Feeder - 1500
Manual Feed	On top of Upper Tray On top of Lower Tray (feed paper and Envelopes)	MP tray (Paper and Envelopes)	On top of Upper Tray On top of Lower Tray (feed paper and Envelopes)

Output tray capacity (sheets)	Upper Bin - 500 pages Rear Tray -50 Optional Bin 2000 pages	250 pages into top tray	Upper Bin - 500 pages Rear Tray -50
Correct-order output	Yes	Yes	Yes
Output Bin Full Sensor	Yes	No	Yes
Job Offset	Yes (Config Menu Item)	No	Yes (Config Menu Item)
Envelope printing capabilities	Optional Envelope Feeder (100 max.) Manual Feed	Optional Envelope Feeder (100 max.) MP Tray (10 max) Manual Feed	Optional Envelope Feeder (100 max) Manual Feed
Duty Cycle	50,000 pages/month	20,000 pages/month	50,000 pages/month
Duplex printing	Yes (Optional)	No	Yes (Optional)
Memory - Standard User Optional Maximum Optional memory	2 Mb 1600 Kb (600 DPI) 1,2,4 & 8 Mb SIMMs 32 Mb	2 Mb 1474 kb (600 DPI) 1,2,4 & 8 Mb SIMMs 32 Mb	1 Mb 660 Kb 1, 4& 8 Mb SIMMs 16 Mb
Environment Saving	Yes	No	No
Interfaces	Bi-directional Parallel and MIO	RS-232C/RS-422 Serial, bi-directional Parallel and MIO	RS-232C/RS-422 Serial, Centronics or MIO
SIMM connectors	4	4	4
MIO Slots	2	1	1
Cartridge Slots	2 Slots named C1 & C2	1	2 Slots named Upper & Lower
User Display	Liquid Crystal Display	Vacuum Fluorescent Display	Liquid Crystal Display
Jam Sensors	4 Jam sensors (4 Jam Messages) (Jam recovery not supported in PostScript)	1 Jam sensor (1 Jam Message)	1 Jam sensor (1 Jam Message)
Display Localization	Localized in 11 languages	Localized in 11 languages	Localized in 5 languages
Automatic ONLINE	Yes	No	No
Environment Saving	Yes (10 MB memory need for simplex, 16MB for Duplex)	No	No
Bitmapped Fonts - Internal Internal Symbol Sets Downloadable	1 32 limited only by memory	1 32 limited only by memory	14 25 limited only by memory
Scalable Typefaces - Internal Internal Symbol Sets Downloadable	45 32 MSL (27 Unicode) Limited only by memory	45 32 MSL (27 Unicode) limited only by memory	13 40 limited only by memory
Font Scaling Technologies	Intellifont TrueType	Intellifont TrueType	Intellifont
Max. no. of macros/forms	Limited only by memory	Limited only by memory	Limited only by memory
HP-GL/2 Macro support	Yes	Yes	No
Additional Language Support	PostScript Level 2 †† ‡‡ IBM Proprinter/Epson ††	PostScript Level 2 †† ‡‡ IBM Proprinter/ Epson ††	PostScript Level 1† Level 2 Planned
Language Switching between PostScript and PCL	Yes Auto Switching PJL Switching (Recommended)	Yes Auto Switching PJL Switching (Recommended)	Yes PJL Switching

**	Many 600 DPI full pages will print in base memory
†	PostScript ROM or SIMM needed
††	Personality SIMM needed
‡‡	Additional memory is required for 600 DPI PostScript

LaserJet 4Si / LaserJet 4 Menu Differences

♦ In the 'CONFIG MENU', there is a menu item called 'DENSITY' for LaserJet 4. In prior LaserJet printers and in the LaserJet 4Si printer, there is a manual lever inside the printer which can control the actual density of toner laid down on the paper. For LaserJet 4 and future

Cannon engines, the density of the toner will be controlled via software. For LaserJet 4Si, it is still a mechanical lever inside the printer and will not appear in the 'CONFIG MENU'.

- LaserJet 4Si doesn't have an MP Tray, this key will not appear on the control panel. LaserJet 4Si allows paper tray selection within the 'PRINTING MENU'.

- In the 'CONFIG MENU', the ' LOW TONER' factory defaults and selections do not match between LaserJet 4 and LaserJet 4Si. In the LaserJet II, IID, IIP, IIIP, III, and IIID, when toner low was detected, an advisory message was placed on the LCD and printing continued. LaserJet 4 default is to place the warning message on the display and to continue printing. The LaserJet IIISi and LaserJet 4Si have a three tiered toner low display. When toner low is first detected, '16 TONER LOW 1' is displayed and the printer halts (if LOW TONER = STOP) all printing until the operator intervenes (change toner cartridge, or override by clicking the Continue key). Note that if the printer is in duplex mode, the printer will finish placing toner on both sides of the last sheet and deliver it to its output bin. In the case of manual feed, the operator will still have to intervene in some manner after placing some media in a manual feed slot. Assuming the operator overrides, after 200 more pages, the system will display '16 TONER LOW 2', and the printer halts (if LOW TONER = STOP) all printing again. Again, the operator must intervene. If the operator overrides, the cycle is repeated for 200 more pages and then the message "16 TONER LOW 3' is displayed. If the operator continues to override, after each 200 pages, the "16 TONER LOW 3' message will be displayed and the printer will halt (if LOW TONER = STOP). Therefore, the default for the 'TONER LOW' menu item for LaserJet 4Si will be STOP (the machine will display message and halt, waiting for operator intervention). If the operator sets this menu item to CONT, then the message "16 TONER LOW 1' will be displayed, but the machine will continue to print. It is possible that data could be lost due to the toner cartridge running completely dry.

LaserJet 4Si / LaserJet IIISi Differences.

- The actual control panel electronics board, keys, and LCD display will be physically the same as LaserJet IIISi. However, the key naming will be different and there is an added Shift key. In LaserJet IIISi, many user control panel functions are obtained by holding down a given key. For example, holding down the MENU key for approximately five seconds will bring up a configuration menu. In LaserJet 4Si, the premise is to eliminate (or greatly reduce) hold-down keys. So, the Shift key was designed to allow access to functions formerly obtained through hold-down. **Note:** Some of the LaserJet IIISi hold-down key functions have been transferred to actual menus and items within menus.

- LaserJet 4Si menuing system provides for a top-down approach to menuing. Each click on the Menu key will display a new menu group. In

LaserJet IIISi, access to Config menus and parts of the self-test function is via hold-down key and access to font printout was via key click on a separate menu.

- Front panel default personality selection is different in LaserJet 4Si from that of the LaserJet IIISi printer. It is a menu item in LaserJet 4Si.

- LaserJet 4Si has changed the way the parallel I/O will be menued. There will be no menu item for 8 bit parallel (or 7 bit parallel) in LaserJet 4Si. The LaserJet IIISi did have such a menu item.

- The LaserJet IIISi printer contained a menu item which allowed the operator to select the default operating system (in this case either PCL system or PostScript system). LaserJet 4Si has no such provision. The new mechanism is to set up a default personality (rather then a system). This is accomplished through a menu item called 'PERSONALITY'. The current choices are still PCL and PostScript 2, but additional personalities will probably be added in the future.

- Most of the '79 SERVICE' type of messages will be different from LaserJet IIISi simply because LaserJet 4 and LaserJet 4Si are using different processors (the ordering of these messages will be as compatible with LaserJet 4 as possible).

- LaserJet 4Si 'COPIES' has expanded from 1 to 999. LaserJet IIISi supported 'COPIES=' from 1 to 99 only.

- LaserJet 4Si Resolution Enhancement (RET) is such that at 600 DPI, the engine has only enough resolving power to handle half dots. Whether the printer is set to 300 or 600 DPI, the LaserJet 4Si engine always prints at 600 DPI and will resolve half dots at all times if RET is ON. Therefore, it will be either OFF or ON in LaserJet 4Si. LaserJet IIISi could resolve 'OFF', 'DARK', 'MEDIUM', and 'LIGHT' at 300 DPI.

- The IIISi printer automatically switched the factory default paper size to A4 whenever the configured language was not English. LaserJet 4Si will be backward compatible to LaserJet 4 in that there will be a Customization variable located in the Service Mode menu. The Menu key will allow the HP factory or service personnel to change the value for paper size. If the operator clicks the Menu key four times, they will see the non-localized message 'CR PAPER=LETTER' in the display. The abbreviation CR stands for "Cold Reset". The allowed choices are LETTER and A4. When A4 is selected, when the operator performs a cold reset at power on, the factory default will be A4 size paper.

HP LaserJet 4Si Features 1

HP LaserJet 4Si Features

This Chapter provides a comparison of HP LaserJet 4Si printer features with those of Hewlett-Packard's LaserJet III printer. This information will aid you in determining the effort required to support the LaserJet 4Si printer.

The LaserJet 4Si printer provides benefits which include a higher resolution print engine, faster processor, enhanced paper handling capabilities and a robust Printer Job Language (PJL). The LaserJet 4Si printer supports PCL 5, optional PostScript Level 2 language.

The LaserJet 4Si printer has been designed to maintain a high level of compatibility with previous models of LaserJet printers and a new level of language and platform flexibility. This means you may consider leveraging parts of your existing LaserJet III/IIID/IIIP or LaserJet IIISi printer drivers into a LaserJet 4Si printer driver.

Included in this Chapter, you will find:

- An overview of the LaserJet 4Si printer features
- A printer feature comparison matrix
- Information on internal scalable typefaces and symbol sets
- Information regarding paper handling
- Standard interface information
- DPI printing information
- LaserJet 4Si print engine information
- An overview of language enhancements

Features of The LaserJet 4Si Printer

- LaserJet 4Si print quality of 600 dots per inch with Resolution Enhancement ™ technology, producing hard copy at a maximum speed of 17 pages per minute.
- Intel 80960CF 25 MHz RISC processor for fast throughput.
- Scalable typefaces in 32 symbol sets.
- Two standard paper trays that hold 500 sheets of paper each, and an optional power envelope feeder (capacity 75 envelopes).
- 2 MB standard memory.
- Firmware raster compression, allowing for enhanced graphics performance in standard memory (typically, all 300 DPI images and many 600 DPI images will print in base memory).
- A top (correct order) output tray, holding up to 500 pages.
- Enhanced Printer Job Language (PJL) providing access to language control, printer status and front panel features from the data stream.
- PCL 5 Printer Language enhanced to provide support for type and raster images at 600 DPI, HP-GL/2 macro support and new HP-GL/2 features.
- SIMM connectors, providing space for additional memory (up to 32 Megabytes), personalities, such as PostScript, or ROM SIMM based type products

- Two cartridge slots, supporting LaserJet printer family font cartridges and scalable typeface cartridges.
- Ability to accept downloaded bitmapped and scalable fonts, limited only by the amount of available printer memory.
- Environment Saving

Internal Fonts

The set of internal fonts offered by the LaserJet 4Si printer is designed to fully meet the printing needs of many LaserJet customers. It differs from previous printers in quantity of type and type formats supported.

LaserJet 4Si Font Offering

The LaserJet 4Si printer contains 1 fixed space bitmapped font, 2 fixed space scalable typefaces and 43 proportionally spaced scalable typefaces. The scalable font offering of the LaserJet 4Si printer is a superset of scalable fonts offered in the LaserJet III/IIID/IIIP printers. The LaserJet 4Si printer supports Intellifont and TrueType font scaling technologies. The resident printer type offering consists of 35 Intellifont scalable typefaces and 10 TrueType scalable typefaces.

Bitmapped Fixed Space Font

Typeface	Treatment
Line Printer	Regular

Fixed Space Scalable Typefaces

Typeface	Treatment	Technology
Courier	(medium), Bold, Italic, Bold Italic	Intellifont
Letter Gothic	Regular, Bold, Italic	Intellifont

Proportionally Spaced Scalable Typefaces

Typeface	Treatment(s)	Technology
Albertus	Medium, Extrabold	Intellifont
Antique Olive	(medium), Bold, Italic	Intellifont
Arial	(medium), Bold, Italic, Bold Italic	TrueType
CG Omega	(medium), Bold, Italic, Bold Italic	Intellifont
CG Times	(medium), Bold, Italic, Bold Italic	Intellifont
Clarendon Condensed	--	Intellifont
Coronet	--	Intellifont
Garamond	Antiqua, Kursiv, Halbfett, Kursiv Halbfett	Intellifont
Marigold	--	Intellifont
Symbol	--	TrueType
Times New Roman	(medium), Bold, Italic, Bold Italic	TrueType
Univers	(medium), Bold, Italic, Bold Italic	Intellifont
Univers Condensed	(medium), Bold, Italic, Bold Italic	Intellifont
Wingdings	--	TrueType

Symbol Sets

The following tables list the internal symbol sets of the LaserJet 4Si printer. Supported symbol sets are indicated by bullets.

Symbol Set	Command	Intellifont	TrueType
Roman-8	EC(8U	•	•
ECMA 94 (Latin 1)	EC(0N	•	•
PC-8 (IBM-8)	EC(10U	•	•
PC-8 DN (Denmark Norway)	EC(11U	•	•
PC-850 (IBM-850)	EC12U	•	•
Legal	EC(1U	•	•
ISO 4 (United Kingdom)	EC(1E	•	•
ISO 6 (ANSI ASCII)	EC(0U	•	•
ISO 11 (Swedish: names)	EC(0S	•	•
ISO 15 (Italian)	EC(0I	•	•
ISO 17 (Spanish)	EC(2S	•	•
ISO 21 (German)	EC(1G	•	•
ISO 60 (Norwegian v1)	EC(0D	•	•
ISO 69 (French)	EC(1F	•	•
Ventura Math	EC(6M	•	
Ventura International	EC(13J	•	•
Ventura US	EC(14J	•	•
PS Math	EC(5M	•	
PS Text	EC(10J	•	•
Math-8	EC(8M	•	
Pi Font	EC(15U	•	
Microsoft Publishing	EC(6J	•	
Windows	EC(9U	•	•
DeskTop	EC(7J	•	•
ISO 8859/2 Latin 2 *	EC(2N	•	•
ISO 8859 /2 Latin 5 *	EC(5N	•	•
Windows 3.1 Latin 5 *	EC(5T	•	•
Windows 3.1 Latin 2 *	EC(9E	•	•
PC-8 TK	EC(9T	•	•
Macintosh	EC(12J	•	•
PC-852 Latin 2 *	EC(17U	•	•
Windows 3.1	EC(19U	•	•

* These symbol set can only be used with CG Times, Univers, Courier and Letter Gothic

Paper Handling

The LaserJet 4Si printer comes standard with 2x500 sheet paper trays. In addition, the LaserJet 4Si printer can be equipped with an optional power envelope feeder or an optional 1500 sheet input bin. The LaserJet 4Si also has an optional 2000 sheet output bin.

Input Paper Tray Selection

PCL required no change to support the LaserJet 4Si paper sources. The following tray selection commands provide access to all available sources on the LaserJet 4Si printer.

Command	Selected Tray
EC&l1H	Upper Tray
EC&l4H	Lower Tray
EC&l6H	Envelope feeder

Paper Output

LaserJet 4Si printer has two output locations. The LaserJet 4Si printer output location provides correct order output, located at the top of the print engine and straight paper path located at the back of the print engine. The correct order output holds 500 sheets. The rear (straight path) output holds 50 sheets or with an optional rear output bin, it holds 2000 sheets.

Manual Feed Printing

Manual feed printing with LaserJet 4Si is handled in the same fashion as for LaserJet IIISi, using the guides on the paper tray cover.

Envelope Printing

Power feeding and manual feeding of envelopes is supported by the LaserJet 4Si printer. Power feeding is available using the power envelope feeder. The supported envelopes are Com10, Monarch, D5, and DL.

Envelope size and source can be selected from the printer's front panel. However, customers will expect envelope printing capability from within their applications. An application which supports this functionality should include both an envelope size and source request to ensure that proper selections are made.

NOTE ☞ The printer will automatically pull from an envelope source when an envelope size is requested and only one paper source containing that size is available. With multiple envelope sources available, it is important to issue the appropriate envelope size and source selection commands, for the envelope feeder Ec&l6H in order to ensure that the proper envelope size and source selections are made.

Standard Interfaces

The LaserJet 4Si printer has a standard built-in bi-directional parallel interface. The printer also has 2 MIO slots for JetDirect Peripheral Interface cards or Third Party MIO cards. For more information on LaserJet 4Si connectivity, refer to Chapter 2.

Bi-Directional Parallel

The LaserJet 4Si printer supports bi-directional parallel communication. The LaserJet 4Si printer will ship with a system level driver which will allow bi-directional communications across the parallel port. This utility software is also provided to developers as part of the developers kit.

MIO

The LaserJet 4Si printer's MIO slots allow users to easily adapt the printer to specialized environments such as local area networks or simple shared. The following JetDirect Peripheral Interface cards are available or are planned for the LaserJet 4Si printer:

Novell
Lan Manager
EtherTalk
Localtalk
TCP/IP
Parallel

More information on MIO is in Chapter 2.

Hot I/O's

The LaserJet 4Si printer is capable of simultaneous data reception from all the interfaces. The details of this mechanism are described in Chapter 2, *Connectivity*.

600 DPI

The LaserJet 4Si printer engine always runs at 600 DPI, but the printer has the capability of accepting and processing 300 DPI data when operating in either 300 or 600 DPI mode. This provides backward compatibility with 300 DPI LaserJet printer data. The printer resolution is selectable either from the printer's control panel (see the LaserJet 4Si User's Manual) or from the data stream via PJL command (see @PJL SET RESOLUTION= and @PJL DEFAULT RESOLUTION= commands in this guide). The default device resolution is 600 DPI.

Increase in data

In general, 600 DPI mode requires processing four times the data (by the printer or by the application) to produce a page as does 300 DPI. To render a 600 DPI raster graphic image or bitmapped downloadable font will require that the application generate and transmit four times the data as the same 300 DPI image or font. Similarly, when a scalable font character is requested or a device independent graphic (HP-GL/2) is rendered at 600 DPI the printer will be required to generate four times the data as for the same image rendered at 300 DPI.

Fonts

Scalable fonts are rendered from device independent outlines. When the printer generates a bitmapped representation from a scalable outline, it converts it into a device specific, and resolution specific, object. This means that the rendition of the font will be as true to the outline as is possible on a specific device at a given resolution. Since the LaserJet 4Si printer supports both 300 and 600 DPI resolutions, the rendering of a font may appear slightly different at the two resolutions. For instance the stem weight of a font may appear lighter (or darker) at one resolution as opposed to the other. This rendering will be accurate for the current printer resolution.

Metrics

Font metrics are similarly affected by device resolution. A character may be 1/600th inch narrower (or wider) when rendered at 600 DPI than the same character rendered at 300 DPI. It is extremely important that software developers who require a high degree of precision in character metric calculation and data placement be "resolution aware" when generating data for LaserJet 4Si printers. This will require developers to generate font metric data (from device independent TFM files) for the current operating resolution and selected unit of measure. (Refer to Chapter 3, Language Enhancements, for more information regarding the Units of Measure command.)

HP-GL/2

Like scalable fonts, HP-GL/2 provides a device independent method for generating graphic images. HP-GL/2 line weights rendered at 600 and 300 DPI can also appear to be lighter or darker depending on the current pen width. A line generated from an HP-GL/2 command will be rendered as close to the requested pen width as the device is capable given the current operating resolution.

Resolution Awareness

There are two methods by which an application can be resolution aware, actively and/or passively. To be actively aware means that an application specifically sets, via @PJL SET RESOLUTION={300|600} command, the resolution that the printer should use for the current job. Developers are encouraged not to change values using the @PJL DEFAULT command as this changes the printer default environment of the printer. The @PJL SET command is a job level command and does not change the user default environment. The passively aware application will ask the printer for its current resolution status and composes the page for that resolution or asks the user which resolution the page should be composed in.

NOTE ☞ If a PJL command is issued which results in a change of resolution from the printer's status prior to the request, perishable data will be lost (perishable data is data which is lost on power down).

Print Engine

The LaserJet 4Si printer is a true 600 DPI printer based on a Canon print engine. This engine provides advanced paper handling and imaging capabilities.

Image system

The LaserJet 4Si printer toner cartridge uses micro-fine toner and has a rated life of 8,000 pages at 5% coverage. The printer uses a transfer roller rather than a transfer corona wire.

Resource Saving

LaserJet 4Si supports resource saving of each personalities' permanent resources by setting aside printer memory dedicated to holding personality resources. The amount of memory set aside can be different for each installed personality. Some personalities may have memory set aside for resource saving without requiring all personalities to do so. Anytime the amount of memory dedicated to a specific personality is changed all personalities will lose any and all resources that had previously been saved including unprocessed print jobs. More information on Resource Saving is in Chapter 4.

Language Enhancements

The LaserJet 4Si printer supports Hewlett-Packard's Printer Control Language (PCL), Hewlett-Packard Graphics Language (HP-GL/2) and Printer Job Language (PJL). You may well be familiar with one or more of these languages. The following is a list of the new features for each of the languages supported by the LaserJet 4Si printer. Specific details regarding implementation and use of features new to the LaserJet 4Si printer will be discussed in Chapter 3, *Language Enhancements*.

PCL

The following are new PCL features for the LaserJet 4Si printer (or were introduced in recent printers).

Mode 5 Raster Compression

This feature was introduced in the LaserJet IIIP printer. Mode 5 raster compression allows raster graphics to be block packaged. Mode 5 compression can greatly reduce the amount of raster data which must be sent to the printer from an application. For more information on this feature, refer to the *PCL 5 Comparison Guide* (P/N 5961-0510).

User Defined Patterns

User defined patterns were introduced with the LaserJet IIIP printer. This feature has been modified to now accept resolution specified patterns (300 DPI or 600 DPI). For more information on this feature refer to Chapter 3, *Language Enhancements*.

User Defined Symbol Sets

This feature was introduced in the LaserJet IIIP and is supported by the LaserJet 4Si printer. For more information regarding implementation of this feature refer to the *PCL 5 Comparison Guide* (P/N 5961-0510)

PCL Status ReadBack

This feature allows the printer to provide status to the host device regarding available fonts and memory in the printer. This feature is new to the LaserJet 4Si printer. Information regarding this feature is in Chapter 3, *Language Enhancements*.

HP-GL/2 in Macros

HP-GL/2 vector graphics are now supported in macros. This feature is new in the LaserJet 4Si printer. Supporting this feature involves no new commands.

Raster Graphics Resolution

The raster graphics resolution command has been modified to support 600 and 200 DPI images. For more information regarding this command refer to Chapter 3, *Language Enhancements*.

PCL Encapsulated True Type outlines

The LaserJet 4Si printer supports both Intellifont and TrueType scalable typefaces. TrueType typefaces are supported as PCL Encapsulated TrueType Outlines (PCLETTOs). For more information regarding PCLETTOs refer to Chapter 3, *Language Enhancements*.

Bitmapped soft fonts

The LaserJet 4Si printer supports all LaserJet compatible soft fonts. In addition, LaserJet 4Si will support resolution specified bitmap soft fonts. These fonts can be defined in either 300 or 600 DPI. For more information regarding resolution specified bitmap fonts, refer to Chapter 3, *Language Enhancements*.

Units of measure command

The units of measure command is new to the LaserJet 4Si printer. This command is designed to allow the data stream to control the "dot move" of the printer. The default unit of measure command is 300 (1/300 inch). This command affects font metrics and commands specified in dots. For more information regarding units of measure, refer to Chapter 3, *Language Enhancements*.

HP-GL/2

HP-GL/2 has been enhanced to provide additional functionality requested by software developers and customers. The following commands have been either added to, or enhanced, in the LaserJet 4Si implementation of HP-GL/2.

Bezier curves (BZ and BR)

This function allows software to specify cubic parametric curves using four control points (the current pen position and three additional control points.). Bezier curves are a graphic primitive common to many software applications. Bezier curves provide developers an efficient method of describing curves which previously had to be decomposed into individual vectors by the application prior to being sent to a LaserJet printer. Bezier curves also use printer resources more efficiently than standard arc and circle functions. For more information regarding Bezier curves, refer to Chapter 3, *Language Enhancements*.

Non-zero winding fills (FP1)

The LaserJet 4Si printer now supports non-zero winding fills for polygons. Non-zero winding fills give software greater flexibility and control over fill patterns as they are applied to polygons. This feature, in some cases, will simplify polygon representation. For more information regarding non-zero winding fills, refer to Chapter 3, *Language Enhancements*.

PCL Compatible Label Origin (LO21)

This feature provides a PCL compatible label origin for label placement in HP-GL/2. When using LO21, labels will be rendered from the character reference point as they are in PCL. For more information regarding label origins, refer to Chapter 3, *language Enhancements* or the *PCL 5 Technical Reference Manual*.

Fill Type and Screened Vectors (FT22 and SV22)

This feature was introduced with the LaserJet IIIP and user defined raster fill patterns, but overlooked in previous documentation. Refer to Chapter 3, *Language Enhancements* for more information on FT22 and SV22.

PJL

Printer Job language was first introduced with the LaserJet IIISi printer. The LaserJet 4Si printer provides a greatly enhanced version of PJL. In addition to the kernel commands (supported by all PJL printers), extensive device control commands have been implemented. These commands are discussed in depth in Chapter 3, Language Enhancements.

Job Control Commands

These commands are provided to allow the host to control print jobs by providing support for job status and page status. Applications should encapsulate the print job with the JOB and EOJ commands even if you are printing through the spooler.

Environment Commands

These commands are used to set the values for the environment variables. Environment commands can be used to change the current (or default) values of the user default environment (front panel settings). The control panel values have the lowest priority and will only be used in the event that neither the PJL nor the page description language (PDL) specifies a default.

Status ReadBack Commands

PJL supports printer to host computer status read back. It is not inconsistent for the individual personalities to provide personality dependent status read back if appropriate, and even to duplicate some of the PJL status read back functionality. All PJL status read back requests will generate an appropriate response, assuming the printer supports status read back. All PJL status read back responses from the printer will start with @PJL and will be terminated with <<ff>> (a form feed control code ASCII 12). By using the <<ff>> to terminate the response, it is possible to have multiple line responses.

Device Attendance Commands

An additional feature that may be desirable in a shared printing environment is device attendance messages. These messages, sent by the host, will cause a message to be displayed at the control panel and take the printer offline. The operator will then respond to the message and put the printer back on-line to resume printing. A special category, no attendance, also allows the user to customize the printer's control panel ready message.

Connectivity

This chapter discusses new connectivity features of the LaserJet 4Si printer. The LaserJet 4Si printer is equipped with a bi-directional parallel interface. In addition, the LaserJet 4Si printer also supports two Modular I/O (MIO) slots. This chapter will cover:

♣ Bi-Directional Parallel
♣ Automatic I/O switching

The Boise (IEEE.P1284) Specification

The Boise Specification (IEEE P1284) is the name of the hardware protocol that provides bi-directional communications via the parallel port on the LaserJet 4Si printer. The *Boise* specification was jointly developed by Hewlett-Packard and Microsoft Corporation. The parallel port of the LaserJet 4Si printer can be configured with advanced functions either OFF, (parallel compatible) or ON, Enhanced Mode. Compatible mode uses standard parallel protocol, while Enhanced Mode provides support for bi-directional communication using the Boise protocol. Using Boise means that the LaserJet 4Si printer can now "talk" to the host CPU. LaserJet 4Si can provide information about installed features such as available memory, print job status, and error conditions. For more information on software commands supporting bi-directional communication refer to Chapter 3, *Language Enhancements*.

High Speed Parallel

Hewlett-Packard has recognized and responded to the need for higher performance data transfer. The LaserJet 4Si printer has modified hardware and firmware to allow much higher I/O rates. When connected to an IBM-PC (or compatible), LaserJet 4Si's I/O rates have been improved to the point that the printer I/O is no longer a bottleneck. The parallel port of LaserJet 4Si can accept data faster than standard IBM-PC hardware using standard BIOS or DOS interrupt drivers. The maximum measured rate of the LaserJet 4Si printer is greater that 150 Kbytes/second for uncompressed graphics data. The data transfer rate for text is somewhat slower because text data must be parsed for page description language and PJL commands. Printer throughput is always data dependent. Complex commands that LaserJet 4Si must execute can reduce throughput.

How to Take Advantage of Boise spec

The printer and host PC must be properly configured to take advantage of the bi-directional communication provided by the Boise protocol. The steps discussed in this section that pertain to the host side of the I/O will center on the MS-DOS operating system running on the IBM-PC (or compatible) platform. Similar steps must be taken on other platforms to achieve successful bi-directional communications. The parallel port hardware drivers in the MS-DOS operating system currently do not provide facilities to achieve bi-directional communications. The BIOS interrupt 17H which drives the parallel port is write-only. MS-DOS file system interrupts (like DOS int 21h function 40) that can read from files are prevented from doing so if the file is a logical printer port. Software drivers must be added to provide this advanced functionality. Hewlett-Packard will provide this driver to developers as part of the development kit and will be shipping the driver with the printer. The driver will provide an interface for software applications, so that the software application will be able to receive the data from the printer. The printer must also be configured for bi-directional parallel from the printer's front panel (ADV FNCTNS=ON).

There are two control panel settings specific to the parallel port in the LaserJet 4Si printer. These settings are ADV FNCTNS and HIGH SPEED. Refer to the LaserJet 4Si Users Guide for control panel setting instructions. The HIGH SPEED mode setting changes the response time of the parallel port. This is accomplished by simply changing the minimum width of the Busy signal (the LaserJet 4Si printer response time is still within IBM's published timing specifications). As parallel port speeds improve, we have found that some software and hardware products are unable to accommodate the higher data transfer rates. To solve this problem, the LaserJet 4Si printer allows the user to set HIGH SPEED = NO. When parallel port High Speed is enabled, the minimum Busy pulse width is changed from 10 microseconds to 0.5 microseconds. The 10 microseconds more closely matches the response time of the HP LaserJet III family of printers. The LaserJet 4Si printer ships from the factory with High Speed = NO as a default. Setting HIGH SPEED=YES will work on most software and hardware platforms and potentially increase system throughput. For more information on parallel signal timings consult either the PCL 5 Technical Reference manual, or the BOISE Bi-Directional Parallel Hardware Specification.

Protocol Description

The hardware specification is available to developers on request.

I/O Switching

On the LaserJet 4Si printer, all I/O ports can be connected and active at the same time (parallel and both MIOs, also multiple connections on a MIO card, 5.1 only). The LaserJet 4Si printer scans the I/O ports looking for input data.   I/O switching cannot be disabled. The Contention for active port is handled by first come first serve algorithm. Once data is received at a port, that port becomes the active port, and the other ports are disabled (or shown as BUSY) until the active port's job is complete. If no port is active, all ports are ready to accept data.

Control Panel Settings

The active port stays active until a job boundary is detected. A job boundary can be explicit in the data stream or implicit when the job TIMEOUT is reached. Explicit job boundaries require well formed jobs. Hewlett-Packard recommends that, at a minimum, each job begin and end with the Universal Exit Language/Start of PJL command (see Chapter 3, *Language Enhancements* for more information). In most cases, this will signal a job boundary releasing the printer to immediately search for available data from all sources.

The LaserJet 4Si printer will also allow the active port to be switched if the job Timeout value is reached. The job Timeout value is configurable from the printer's control panel. The default setting is TIMEOUT = 15 seconds. Valid selections for job timeout range from 5 - 300 seconds. For instructions on changing the TIMEOUT setting please refer to the LaserJet 4Si Printer Users Guide.

MIO and Networks

The LaserJet 4Si printer is another of the growing number of HP peripherals that supports Modular I/O (MIO). MIO is a hardware slot that can accept compatible MIO adapter cards for providing I/O options not available in the standard printer product. The hardware and software specification for development of MIO cards is available to qualified hardware developers.

Current HP Products

Hewlett-Packard develops and sells several MIO cards. HP's products center on popular network connections like Novell's Netware and Microsoft's Lan Manager, and popular wiring topologies like EtherNet, EtherTalk and Token Ring. This by no means states that HP MIO products will be confined only to these types of products. HP continues to develop new MIO solutions for peripheral products and the list of available products will continue to grow.

Third Party Developers Program

There are third party MIO cards available that provide a variety of MIO solutions. Qualified hardware developers can apply to obtain the MIO hardware specification by contacting the Hewlett-Packard's Peripherals Developers Program at 619-592-8308.

Language Enhancements 3

Language Enhancements

The following sections detail features which are unique to the LaserJet 4Si printer or have been modified from previous implementations. These sections, when used in conjunction with the PCL5 Printer Language Technical Reference Manual (PN 5961-0509) and the PCL5 Comparison Guide (PN 5961-0510), provide a complete list and description of the available LaserJet 4Si printer features.

PCL Commands

The PCL commands section provides a list of commands which have been either added to or modified in the LaserJet 4Si printer. Additional PCL commands supporting status read back are covered later in this chapter.

Number of Copies Command

Designates the number of printed copies of each page. The range of values supported by the LaserJet 4Si printer has been increased from previous printers. This command can be received anywhere within a page and affect the current page as well as subsequent pages.

$^{E}_{C}$ & l # X

# = Number of copies
Default = 1
Range = 1 - 32767
values < 1 are ignored
values > 32767 map to 32767

Set Raster Graphics Resolution

The set raster graphics resolution command defines the resolution in dots per inch at which raster graphics data is to be printed. The LaserJet 4Si printer can operate in either 300 or 600 DPI mode (selectable from the printer's control panel or PJL command). The available raster graphic resolutions differ depending on the current resolution setting for the printer. Refer to the table below for supported values. This command is ignored after the Start Raster Graphics (**$^{E}_{C}$*r#A**) command is received, until the next End Raster Graphics (**$^{E}_{C}$*rB** or **$^{E}_{C}$*rC**) is received.

$^{E}_{C}$* t # R

# = Resolution in dots per inch
Default = 75
Range = 0 - 32767

Values (#)	Resolution = 300	Resolution = 600
75	supported	supported
100	supported	supported
150	supported	supported
200	N/A	supported
300	supported	supported
600	N/A	supported

Units of Measure

Defines the unit of measure for PCL dot commands. The value field defines the number of units per inch for PCL dot moves, Horizontal and vertical rectangle size, bitmap and scalable font metrics. This command affects only the interpretation of movement and size not the interpretation of binary raster data.

$^{E}{}_{C}$& u # D

Value (#) - number of units per inch
Default - device dependent (for LaserJet 4Si the default is 300)
Range - 96, 100, 120, 144, 150, 160, 180, 200, 225, 140, 288, 300, 360, 400, 450, 480, 600, 720, 800, 900, 1200, 1440, 1800, 2400, 3600 and 7200

The units value is part of the user environment. Thus it is saved and restored whenever a macro is called or an overlay is invoked, and defaulted when establishing the user environment in preparation for an overlay. Control panel and **$^{E}{}_{C}$E** resets will default the unit of measure to the device default setting.

Values out of range are mapped to the supported value with the minimum error. For example, the relative error of a unit selection of 4801 is closer to 7200 than to 3600 (abs(4801-7200)/7200 < abs(4801-3600)/3600).

User-defined pattern

This command is used to download binary data that defines a user-defined pattern. This command was first introduced with the LaserJet IIIP printer. The LaserJet 4Si printer supports a new Resolution Specified Pattern Data Descriptor has been added.

$^{E}{}_{C}$* c # W [pattern data]

The value field of this escape sequence contains the number of binary bytes of pattern data. The pattern is assigned the current value of Pattern ID (set via **$^{E}{}_{C}$*c#G**). If a pattern is already associated with this ID, the existing pattern will be deleted before the new pattern is downloaded.

Any missing pattern data is filled with nulls, while any excess data is discarded. If an invalid pattern is received, it will be discarded. The downloaded pattern must be word aligned (16 bit).

The pattern data contains a complete description of the pattern as specified in the following tables. The Specified Resolution Pattern Data Description is a new descriptor for LaserJet 4 which includes resolution information for pattern management.

Pattern Data Description

Byte	15 - MSB	7
0	Format (0)	Continuation (0)
2	Pixel Encoding (1)	Reserved (0)
4	Height in Pixels	
6	Width in Pixels	
8	Pattern Image . . .	

Resolution Specified Pattern Data Description

Byte	15 - MSB	7
0	Format (20)	Continuation (0)
2	Pixel Encoding (1)	Reserved (0)
4	Height in Pixels	
6	Width in Pixels	
8	X Resolution	
10	Y Resolution	
12	Pattern Image . . .	

NOTE ☞ LaserJet 4Si printer may operate in either 300 DPI or 600 DPI. When a pattern is requested, while operating in 600 DPI, the pattern descriptor format will be checked. If the format number is 0 (zero) the pattern is assumed to be a 300 DPI image. If the format number is 20 this field is checked for the resolution. Any 300 DPI image requested while operating in 600 DPI mode will be scaled to the correct size. Any 600 DPI pattern downloaded while operating in the 300 DPI mode is discarded.

Format

0 = Standard Pattern Description
20 = Resolution Specified Pattern Description

Continuation - This is 0 (zero) in LaserJet 4Si.

Pixel Encoding - This should be set to 1 (one).

Height in Pixels - The number of raster rows in the pattern.

Width in Pixels - The number of pixels in each pattern row.

X Resolution - This field specifies the horizontal resolution.

Y Resolution - This field specifies the vertical resolution.

NOTE ☞ The values for X resolution and Y resolution must be equal (either 300 or 600).

Pattern Image - raster data for the pattern. The raster rows must be word aligned.

Create Font

Creates a font descriptor and assigns it the current font ID. The LaserJet 4Si printer will accept all previously defined font descriptors (refer to the PCL 5 Printer Language Technical Reference Manual). In addition, the LaserJet printer allows for resolution specified bitmapped fonts and PCL Encapsulated TrueType Outlines (PCLETTOs).

$^{E}{}_{C}$)s# W [font descriptor]

# = number of bytes in the font descriptor
Default = NA
Range = 0 - 32767

NOTE ☞ This escape sequence must be sent prior to downloading the characters in the font.

If the font specified by Font ID already exists, it is deleted. If the font descriptor is invalid or if there is insufficient memory to create the font, the command is ignored and the font descriptor and downloaded character data are discarded; otherwise, the new font is defined.

Because the descriptor is used during the font selection process and by some application packages to present font choices to the user, it should be defined as accurately as possible. Missing descriptor data should be set to zeros; excess data should be discarded. Font vendors should be advised to set "reserved" descriptor fields to zero.

Font Descriptor for Resolution Specified Bitmapped Fonts

Byte	**15 - MSB**	**7**
0	Font Descriptor Size (68)	
2	Descriptor Format (20)	Font Type
4	Style MSB	Reserved
6	Baseline Distance	
8	Cell Width	
10	Cell Height	
12	Orientation	Spacing
14	Character Set	
16	Pitch (Default HMI)	
18	Height	
20	x Height	
22	Width Type	Style LSB
24	Stroke Weight	Typeface LSB
26	Typeface MSB	Serif Style
28	Quality	Placement
30	Underline Position	Underline Height
32	Text Height	
34	Text Width	
36	First Code	
38	Last Code	
40	Pitch Extended	Height Extended
44	Cap Height	

44-46	Font Number	
48-63	Font Name	
64	X Resolution	
66	Y Resolution	
68	Copyright (optional)	

Font Descriptor for TrueType Fonts

Byte	**15 - MSB**	**7**
0	Font Descriptor Size	
2	Descriptor Format (15)	Font Type
4	Style MSB	Reserved
6	Baseline Position	
8	Cell Width	
10	Cell Height	
12	Orientation	Spacing
14	Character Set	
16	Pitch	
18	Height	
20	x Height	
22	Width Type	Style LSB
24	Stroke Weight	Typeface LSB
26	Typeface MSB	Serif Style
28	Quality	Placement
30	Underline Position	Underline Height
32	Text Height	
34	Text Width	
36	First Code	
38	Last Code/Number of Characters	
40	Pitch Extended	Height Extended
42	Cap Height	
44		
46	Font Number	
48		
.		
.		
.		
62	Font Name	
64	Scale Factor	
66	Master Underline Position	
68	Master Underline Thickness	
70	Font Scaling Technology	Variety
72	Additional data	

Font Descriptor Definitions

The following notation is used to define the data type of each field of the font descriptor:

Notation	**Data Type**
(BOOL)	Boolean (0,1)
(UBYTE)	Unsigned Byte (0 .. 255)
(SBYTE)	Signed Byte (-128 .. 127)
(UINT)	Unsigned Integer (0 .. 65535)
(SINT)	Signed Integer (-32768 .. 32767)
(ULINT)	Unsigned Long Integer (0 ..(2^{32}-1))
(SLINT)	Signed Long Integer (-2^{31} ..(2^{31}-1)
(ASCxx)	ASCII string (array (0 .. (xx-1)) of characters

Font Descriptor Size: (UINT) Specifies the number of bytes in the font descriptor. The following table lists appropriate values for both resolution specified bitmap fonts and PCLETTOs.

Value	Description
68	Resolution Specified Bitmap Font
72	PCLETTO

Descriptor Format: (UBYTE) This field indicates:

Value	Format
0	bitmapped
10	Scalable Intellifont (bound PCLEO)
11	Scalable Intellifont (unbound PCLEO)
15	Scalable TrueType (PCLETTO)
20	Resolution Specified Bitmap

NOTE ☞ Unrecognized descriptor values invalidate font creation.

Font Type: (UBYTE) This field should be compatible with the character set for the font. More values may be defined in the future.

Value	Character Set Organization
0	7-bit (32-127 are printable)
1	8-bit (32-127 and 160-255 are printable)
2	PC-8 (All codes are printable except 0, 7-15, and 27) *
10	No character set affiliation **
11	Character codes correspond to Unicode numbers

* All character code positions print in transparency mode.
** This value is only valid for LaserJet IIIP & LaserJet 4Si unbound scalable fonts. This value is not valid for TrueType fonts.

Style Word: (UINT) The style word is composed of the style MSB (byte 4) and the style LSB (byte 23). This value, used in the font selection process, specifies the style of the font. The partial sums for posture, width, and structure are added together to determine the desired value (#). More values may be added in the future. The composition of the style word (bytes 4 and 23 of font descriptor) is shown below:

15		**9**	**4**	**1**
X	Reserved	Structure	Appearance width	Posture

Value	Posture (style word partial sum)
0	Upright
1	Italic
2	Alternate Italic
3	Reserved

Value	**Appearance Width** **(multiply by 4 for StyleWord partial Sum**
0	Normal
1	condensed
2	Compressed or Extra Condensed
3	Extra Compressed
4	Ultra Compressed
5	Reserved
6	Extended or Expanded
7	Extra Extended or Extra Expanded

Value	**Structure** **(Multiply by 32 for Style Word partial sum)**
0	Solid
1	Outline
2	Inline
3	Contour
4	Solid with Shadow
5	Outline with Shadow
6	Inline with Shadow
7	Contour with Shadow
8-11	Patterned (complex patterns subjective to type family)
12-15	Patterned with Shadow
16	Inverse
17	Inverse in Open Border
18-31	Reserved (should be set to zero)

If a value is requested and a match not made, the request is ignored and the current font selection process continues as if the parameter had never been requested.

For TrueType fonts, Style is derived from the PCLT table in the TrueType font file.

Baseline Distance: (UINT) Bitmapped Font - Specifies the distance from the baseline to the top of the cell. The baseline is a dot row (not a line between the dot rows) used to place all the characters in a line. The units are dots in the logical coordinate system (i.e. the value is the same for all orientations).

The design cell for a font must be tall enough to accept the tallest character and greatest descender, and wide enough to accept the widest character. The baseline position must be contained within the cell; thus the legal value for the baseline position may range from 0 to cell height minus one.

Intellifont - Specifies the distance from the bottom of the design window to the baseline position in design window units which are defined within Intellifont. This value is ignored by the printer.

TrueType - This value should be set to zero.

Cell Width: (UINT) Specifies the width of the character cell. The legal range for the cell width is 1 to 65535.

Bitmapped Font - Specified in dots in the logical coordinate system.
Intellifont - Specified in dimensional units.
TrueType - Derived from the header table of the TrueType font file (xmax - ymin).

Cell Height: (UINT) Specifies the height of the character cell. The legal range for the cell height is 1 to 65535.

Bitmapped Font - Specified in dots in the logical coordinate system.
Intellifont - Specified in dimensional units.
TrueType - Derived from the header table of the TrueType font file (ymax - ymin).

Orientation: (UBYTE) Specifies the orientation of the font. Zero implies 0 degrees (the orientation of the bitmapped scan of the engine), one implies 90 degrees counter clock wise (CCW), two implies 180 degrees CCW and three implies 270 degrees CCW. All images within the set must have the same orientation; otherwise they will be discarded as they are downloaded.

Bitmapped Font - Unsupported values invalidate font creation.
Intellifont - set to zero.
TrueType - set to zero.

Spacing: (BOOL) This value, used in the font selection process, specifies the class of spacing for the font. More values may be defined in the future.

Value	Type of spacing
0	Fixed
1	Proportional

Character Set: (UINT) Specifies the character set for the font. This descriptor field is derived from the value(#) and ID of the $^{E}_{C}$**(#ID** or $^{E}_{C}$**)#ID** command as follows:

Character Set Field Value = (# * 32) + (ID - 64)

The character set ID is always a number followed by an uppercase letter (e.g., 8U).

Pitch (UINT) Bitmapped font - Specifies the pitch of the font in quarter-dot units. It combines with Pitch Extended to specify the pitch of the font in 1/1024 dots. Pitch is the default HMI (horizontal motion index) for the font.

Intellifont - Contains the master design pitch of the font in dimensional units which, when multiplied by design height, gives the design aspect ratio of the font.

TrueType - This value can be derived from the PCLT table in the TrueType font file.

Height (UINT) Bitmapped Font - Contains the height in quarter-dot units. It combines with height Extended to specify the design height of the font in 1/1024 dots.

Intellifont - Contains the master design height of the font which when multiplied by design pitch gives the design aspect ratio of the font.

TrueType - For a TrueType font this field is set to zero.

xHeight (UINT) Bitmapped Font - Specifies the height of the lower case "x" in quarter dots.

Intellifont - Specifies the distance from the baseline to the lower case "x" height in dimensional units

TrueType - The height of the optical line describing the height of the lowercase "x" in FUnits derived from the PCLT table of the TrueType font file.

Width Type (SBYTE) Defines the proportionate width of characters in the font. This field is ignored by LaserJet 4Si printers.

Stroke Weight (SBYTE) This value, used in the font selection process, specifies the thickness of the strokes used in designing the font. The supported stroke weight values are -7 through 7. This value can be derived from the PCLT table of the TrueType font file.

Typeface [LSB/MSB] (UBYTE) Identifies the typeface of the font. This value can be derived from the PCLT table of the TrueType font file.

Serif Style (UBYTE) Specifies the style of the serif used in the design of the font. for TrueType fonts, this value is extracted from the PCLT table in the True Type font file. This value field is ignored by LaserJet 4Si printers.

Quality (UBYTE) Specifies the quality or density of the font. This value is ignored by the printer.

Placement (SBYTE) Specifies the placement of the font. This value is ignored by the printer.

Underline Position (SBYTE) Bitmapped font - Specifies the distance in dots of the underline from the baseline to the top dot row of the underline.

Intellifont - Set to zero. See Master Underline Distance.

TrueType - Set to zero. See Master Underline Distance.

Underline Height (UBYTE) Bitmapped Font - Specifies the thickness of the underline in dots.

Intellifont - Set to zero. See Master Underline Distance.

TrueType - Set to zero. See Master Underline Distance.

Text Height (UINT) This is the optimum interline spacing for the font. The typical value of Text Height is 120% of the height of the font.

Bitmapped Font - Specified in quarter-dot units

Intellifont - Specified in dimensional units

TrueType - may be derived from the sum of TypoLineGap from the OS/2 table and units per EM of the head table.

This field is ignored by the printer.

Text Width (UINT) Specifies the font's average lowercase character width. This field is ignored by the printer.

Bitmapped Font - Specified in quarter dots

Intellifont - Specified in dimensional units.

TrueType - value may be derived from the xAvgCharWidth from the OS/2 table.

First code (UINT) Specifies the first downloadable character code in the font. This value will be between 0 and 255 inclusive. The space character may be printable, and will print an image if one is defined, otherwise a space control code is executed. This field is ignored by the LaserJet 4Si printer.

Last Code/Number of Characters (UINT) For a bound font (font types 0, 1 or 2), this field specifies the last downloadable character code in the font. For an unbound font, it specifies the maximum number of characters that can be downloaded into the font. This field is ignored by the LaserJet 4Si printer.

Pitch Extended (UBYTE) Bitmapped Font - This extends the Pitch field an extra 8 bits to all 10 bits of fractional dots. The value of this field is in 1024ths of a dot.

Intellifont - this field is ignored and should be set to zero

TrueType - this field is ignored and should be set to zero.

Height Extended (UBYTE) Bitmapped Font - This extends the Height field an extra 8 bits to all 10 bits of fractional dots. The value of this field is in 1024ths of a dot.

Intellifont - this field is ignored and should be set to zero

TrueType - this field is ignored and should be set to zero.

Cap Height (UINT) Bitmapped Font - Cap height is a percentage of the em of a font used to calculate the distance from the capline (top or an unaccented, upper-case letter) to the baseline. Fonts containing an 0 in this field are assumed to have a cap height percentage of 70.87%. This field is ignored by the LaserJet 4Si printer.

Intellifont - contains the cap height in dimensional units.

TrueType - this value can be obtained from the PCLT table of the font file.

Font Number (ULINT) This field contains the vendor typeface number in the lower three bytes. The most significant byte consists of a flag in the most significant bit indicating whether the font is in its native format (0) or has been converted (1) followed by the ASCII decimal value for the first initial of the font vendor's name in the remaining 7 bits. If a vendor does not assign numbers to their typefaces, then this field should be set to zero.

Font Name (ASC16) This is a 16 byte ASCII character field in which the font name is contained. The printer uses this field when printing the name on the PCL Typeface List printout. This field is also used by several applications to provide more user friendly identification.

X Resolution - This field specifies the horizontal resolution.

Y Resolution - This field specifies the vertical resolution.

NOTE ☞ The values for X Resolution and Y resolution must be equal (either 300 or 600).

Scale Factor (UINT) This field defines the size of the EM in scalable units. For TrueType soft fonts, this value is obtained from the unitsPerEm field of the head table (typical value is 2048 for a TrueType font).

Master Underline Position (SINT) The traditional PCL value for this field is - approximately -1/5 Em. Alternatively, for TrueType fonts, values can be derived from the UnderlinePosition field of the post table.

Master Underline Thickness (UINT) The traditional PCL value for this field is approximately 1/20 Em. Alternatively, for TrueType fonts, values can be derived from the Underline Thickness field of the post table.

Font Scaling Technology (UBYTE) The following values for this field have been defined:

Value	**Font Scaling Technology**
0	Intellifont
1	TrueType

Variety (UBYTE) This value is 0 for TrueType fonts.

Segmented Font Data

The Segmented Font Data section immediately follows the descriptor of a format 15 font. It consists of a list of segments. Each segment contains three parts: a Segment Identifier, Segment Size and Data Segment.

The Segmented Font Data section is terminated by the Null Segment. If no Null Segment is encountered prior to the end of the font header -- as defined in the initial escape sequence -- the font is invalidated. Encountering the Null Segment too soon (prior to byte # - 6, as shown in Figure 2 below) also invalidates the font.

Byte	**15 - MSB**	**7**
x	Segment Identifier (first segment)	
x + 2	Segment Size (first segment)	
x + 4 . . .	Data Segment (first segment) . . (s1 = size of segment one)	
x + 4 + s_1	Segment Identifier (second segment)	
x + 6 + s_1	Segment Size (second segment)	
x + 8 + s_1	Data Segment (second segment) . . .(s2 = size of segment 2)	
x+4n-4 $(s_1+....+s_{n-1})$	Segment Identifier (nth segment)	
x+4n-2 $(s_1+....+s_{n-1})$	Segment Size (nth segment)	
x+4n+ $(s_1+....+s_{n-1})$	Data Segment (nth segment)	
# - 6	Null Segment Identifier (0xFFFF)	
# - 4	Null Segment Size (0)	
# -2	Reserved (0)	Checksum

x - Font Descriptor Size (bytes 0 - 1 of descriptor)
# - Bytes in font descriptor (specified in EC)s#W)

Segment Identifier (UINT) Each entry in the Segmented Font Data section has its own unique identification number. The following values have been defined:

Value	Mnemonic	Data Segment
17219	CC	Character compliment
17232	CP	Copyright
18260	GT	Global TrueType data
20545	PA	PANOSE description
65535		Null Segment

Segment Size (UINT) For each entry in the Segmented Font Data section, the Segment Size indicates the number of bytes in the following Data Segment. The size for the Null Segment is 0.

Formats of Data Segments

CC (Character Complement) This information is derived from the PCLT table in the TrueType font.

CP (Copyright) This field will consist of ASCII data and is an optional field for listing copyright information.

GT (Global TrueType Data) This data segment contains first a Table Directory and then 5 or more tables used by the TrueType font scaler. This is a required data segment for TrueType fonts.

The Table Directory is patterned after the initial segment of the TrueType font file as described on page 224 of *TrueType Font Files, Version 1.00,* Microsoft Corporation, September 1991. The Table Directory has a 12-byte header and 16 bytes per entry in the Table Directory. The Table Directory will be organized in alphabetical order by the 4-byte table names. For each entry, there will be an offset relative to the beginning of the soft font's Global TrueType Data Segment. The TrueType Data Segment for every TrueType font entity must contain a head, hhea, hmtx and maxp table. These tables are defined in *TrueType Font Files, Version 1.00* Microsoft Corporation, September 1991.

Another required table is the gdir table. At the time of the downloading of the font header, the gdir table should have a size of 0 and an offset of 0. The gdir table will then be built in RAM to accommodate the maximum number of glyphs to be downloaded to the given font -- with 2 or 4 bytes of offset and 2 bytes of length per glyph. This maximum number of glyphs is obtained from the numGlyphs field of the maxp table. Entries in the gdir table will be filled in by the TrueType rasterizer as characters are downloaded. (the gdir table is a replacement for the local table.)

The optional cvt, fpgm and prep tables, as defined in *TrueType Font Files, Version 1.00,* Microsoft Corporation, September 1991, will typically appear in the Global TrueType Data Segments of hinted TrueType soft fonts, but ought not to appear in unhinted fonts.

PA (PANOSE Description) This data segment can be extracted from the OS/2 table of the TrueType font file. Its definition continues to evolve.

Font Header Checksum The value of this byte, when added to the sum of all of the bytes from byte 64 of the descriptor through the Reserved byte, should equal 0 in modulo 256 arithmetic.

Character Code

$^{E}{}_{C}$*c# E

Value(#)=Character code
Default=0
Range=0 - 65535

For unbound fonts, the character code for a given character will equal its Unicode index as provided in the cmap table of the TrueType font file.

For a bound font, the Character Set determines what the appropriate code will be for a given character. An ASCII font, for example, would have a code of 65 for the Uppercase A.

A special code (0xFFFF) will be used to download glyphs which never stand alone as characters.

Download Character

$^{E}{}_{C}$(s# W [character data block]

Value(#)=Number of bytes in character data block
Default=N/A
Range=0 - 32767
(for scalable TrueType characters, 10 is the minimum value)

For scalable TrueType fonts, the format of character data blocks is shown below.

Byte	**15 - MSB**	**7**
0	Format (15)	Continuation (0)
2	Descriptor Size	Class (15)
4	Character Data Size	
6	Glyph ID	
8	TrueType Glyph Data	
.		
.		
.		
# - 2	Reserved (0)	Checksum

# - Bytes of character descriptor and data (given in $^{E}{}_{C}$(s#W))

NOTE ☞: If a continuation block will be used, the reserved and checksum bytes should only appear at the end of the continuation block.

Character Data Definitions

Format (UBYTE) This number specifies the format for character downloading. For TrueType characters this value is 15.

Continuation (BOOL) This specifies whether the following data is the character descriptor block (false or 0) or a continuation of the data (true or non-zero) associated with the previous character descriptor.

Bitmapped fonts - as needed
Intellifont - set to zero
TrueType - set to zero

Descriptor Size (UBYTE) This field specifies the size of the character descriptor in bytes. The character descriptor includes everything that is both after the continuation byte and prior to the Character Data Size field. Currently, the TrueType character descriptor includes this Descriptor Size byte plus the following Class byte. The minimum value for Descriptor Size is 2.

Class (UBYTE) This field is used to distinguish different character data types within a given character format. All TrueType scalable characters will be handed to the TrueType font scaler in the same format. For TrueType scalable character this value is 15.

Character Data Size (UINT) The value of the Character Data Size should equal the sum of the sizes of the Character Data Size, Glyph ID, and TrueType Glyph Data fields.

Glyph ID (UINT) This field will be used by the TrueType font scaler as an ID number for the glyph data associated with the given character.

TrueType Glyph Data This field will contain the data segment associated with the given character as found in the glyph table of the original TrueType font file. See the description on page 233 and following, TrueType Font Files, Version 1.00, Microsoft Corporation, September 1991.

Checksum (UBYTE) The value of this byte, when added to the sum of all of the bytes in the Character Data Size, Glyph ID, and TrueType Glyph Data fields, should equal 0 in modulo 256 arithmetic. The Checksum is found in the last character data block associated with a given character.

HP-GL/2 Commands

The HP-GL/2 commands section provides a list of commands which have been either added to, or modified, in the LaserJet 4Si printer.

BZ, Bezier Absolute

BZ $X2_{ctrl}, Y2_{ctrl}, X3_{ctrl}, Y3_{ctrl}, X4_{ctrl}, Y4_{ctrl}$ *[,params[,params]][;]*

This command draws a Bezier curve using the present pen position and the defined control points. The present pen position is used as the first control point. Control points are specified in absolute coordinates.
The curves are drawn with the current pen, line type, current line attributes, and the pen status (up or down). These are clipped to the hard clip limits and the soft clip window. The current position is updated to the end point of the Bezier curve. This command is allowed in Polygon Mode.

After the initial 4 control points are defined (one present position control point and 3 control points) the optional params are used to define subsequent Beziers by adding 3 additional control points. After each new Bezier the last control point is used by the next Bezier as the first control point. If an invalid number of additional control points are included, that Bezier segment will be discarded.

Parameter	Format	Functional Range	Default
X_{ctrl}, Y_{ctrl}	Current Units	-2^{30} to 2^{30} - 1	No Default

BR, Bezier Relative

BR $X2_{ctrl}, Y2_{ctrl}, X3_{ctrl}, Y3_{ctrl}, X4_{ctrl}, Y4_{ctrl}$ *[,params[,params]][;]*

This command draws a Bezier curve using the present pen position and the defined control points. The present pen position is used as the first control point. Control points are specified using coordinates relative to the first control point (initial pen position).

Parameter	Format	Functional Range	Default
X_{ctrl}, Y_{ctrl}	Current Units	-2^{30} to 2^{30} - 1	No default

FP, Fill Polygon

FP*method[;]*

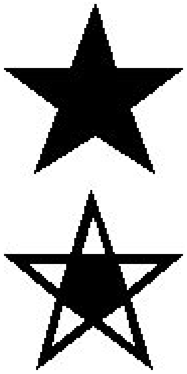
Fills the polygon currently in the polygon buffer. Use FP to fill polygons defined in polygon mode or with the Edge Rectangle or Edge Wedge commands (EA, ER and EW).

The fill polygon command has been modified to support both non-zero winding fill and even/odd fill methods. Previous PCL 5 LaserJet printers only support the even/odd fill method.

NOTE ☞

Non-Zero winding fill use the direction of the vector to determine fill area instead of the counting vectors to determine the fill area. The non-zero algorithm determines whether a point is inside a polygon by (conceptually) drawing a ray horizontally and adding 1 to an internal counter for every polygon up-vector that bounds the region of a given point; likewise, 1 is subtracted for each down-vector that bounds the region. If the count is a non-zero, then the point should be rendered.

Consider a five point star drawn with straight lines that intersect each other. The non-zero winding fill considers all of the star as internal to the polygon. The whole star would be filled. Odd/even fill would consider either the outside triangles or the center as internal to the polygon and fill one or the other.



Parameter	Format	Functional Range	Default
Fill method	clamped integer	0 or 1	0

Fill Method	Description
0	Fills using even odd rule
1	Fills using non-zero winding

FT, Fill Type

FT*fill type[,option1[,option2]][;]*
Selects the shading pattern used to fill polygons (FP), rectangles (RA or RR) or wedges (WG).

A previously undocumented feature of this command is FT22, used to select PCL user defined patterns. This command is valid for all printer supporting PCL user defined patterns.

Parameter	**Format**	**Functional Range**	**Default**
Fill type option1, option2	clamped integer, clamped real	1 to 4, 10, 11, 21, 22, type dependent	1 type dependent

For a description of fill types 1 - 4, 10, 11 and 21, refer to the PCL 5 Printer Language Technical Reference Manual.

Fill Type	**Description**	**Option 1**	**Option 2**
22	PCL user defined pattern	Pattern ID	ignored

LO, Label Origin

LO*position[;]*

Positions labels relative to the current pen location. Use the LO command to center, left justify or center labels.

This command has been modified to provide a PCL compatible Label Origin (LO21) for the placement of HP-GL/2 labels (that is, the current pen position is at the PCL reference point for the character cell). For a description of label origin values of 0 - 19, refer to the PCL 5 Printer Technical Reference Manual.

Parameter	**Format**	**Functional Range**	**Default**
Position	Clamped Integer	1 to 9, 11 to 19, 21*	1

* PCL compatible label origin

SV, Screened Vectors

SV*[screen_type[,option1[,option2]]][;]*

The SV command selects the type of screening area fill to be applied to vectors, arcs, circles and edges of polygons, rectangles and characters or edges of characters.

A previously undocumented feature of this command is SV22, used to select PCL user defined patterns. This command is valid for all printers supporting PCL user defined patterns.

Parameter	**Format**	**Functional Range**	**Default**
Screen type option1option2	clamped integer clamped real	1, 2, 21, 22, type dependent	no screening type dependent

For a description of screened vector types 1, 2 and 21, refer to the PCL 5 Printer Language Technical Reference Manual.

Fill Type	Description	Option 1	Option 2
22	PCL user defined pattern	Pattern ID	ignored

PCL Status Readback

This section describes the PCL status readback features. Status readback allows the user to obtain PCL status information from the printer, such as: available printer (user) memory, a list of fonts and symbol sets, and the ID numbers of macros and user-defined patterns.

To obtain status information, the user must initiate a request by sending either a memory status request or an entity status request, as described in the following pages. All responses will start with PCL<cr><lf> and end with <ff>, where <cr> represents the carriage return character (ASCII 13), <lf> represents the line feed character (ASCII 10) and <ff> represents the form feed character (ASCII 12).

Entity Status

Entity status provides status for the printer's entities. An entity is a font, symbol set, macro, or user-defined pattern. To request entity status, the user must identify a location type, a location unit and then request the status.

Location type refers to the memory locations which store entities. These memory locations include internal ROM, RAM (for downloaded entities), cartridges, user-installable ROM (SIMMs), and one additional location referred to as "currently selected." Currently selected identifies the currently selected entity (such as the currently selected font, it does not apply to macros or symbol sets).

Location unit refers to a specific location (or device) within the location type. For example, location unit 1 for location type "cartridge," identifies the Top cartridge on a printer with two cartridges; or, unit 1 for location type "downloaded," identifies the temporary fonts (as opposed to permanent).

The location type and unit are described in detail under the location type and unit status readback command descriptions provided later in this chapter.

Once the location type and unit are specified, the status can be requested using the Inquire Status Readback Entity command. This command identifies the type of entity (font/font extended, symbol set, macro, or user-defined pattern) and causes the printer to send the response.

The basic flow for an entity status request is shown in the table below. The example commands on the right side of the table identify, in this case, a status request for downloaded temporary fonts (combining the commands the request would appear as $^{E}{}_{C}$***s4t1u0I)**. An example status readback response is also provided at the bottom of the table to show a sample of the response syntax. Note that this example shows only one possible request/response. For more detailed information, refer to the status readback command descriptions provided later in this chapter.

Entity Status Request Command Flow

Operation	Operation Parameters (Value Field)	Example PCL - Command	Example Comments
Set Location Type	All Locations Currently Selected Internal Downloaded Cartridge User-Installable ROM	**EC*s4T**	A command value field of "4" sets Location Type to "Downloaded." If this command is not sent the current location type setting will be used.
Set Location Unit	*All Locations;* Currently *Selected;* Internal; *Downloaded;* All Temporary or Permanent; *Cartridges:* All or Specific One; *User-Installed ROM:* All or Specific One	**EC*s1U**	A command value field of "1" sets the location unit to temporary (Downloaded). If this command is not sent the current location unit setting will be used.
Identify entity and initiate status readback (Inquire Entity)	Font Macro User-defined pattern or Symbol Set	**EC*s0I**	A command value field of 0 selects status for Fonts.

Printer returns status

```
PCL<cr><lf>
INFO FONTS<cr><lf>
SELECT="<Esc>(8U<Esc>(s1p__v0s0b4120T <Esc>(7X"<cr><lf>
SELECT="<Esc>(10U<Esc>(s1p__v0s0b4157T<Esc>(21X"<cr><lf>
<ff>
```

Two temporary downloaded fonts exist in the printer: University Roman (4120T) and Dom Casual (4157T).

In addition to the normal status responses, if an invalid request is made, the printer will return an error response. Error responses are described under the *Inquire Status Readback Entity* command, later in this chapter.

Memory Status Request

It is possible to identify the amount of available user memory in the printer using the memory status request. Knowing the amount of available memory is useful when downloading entities. It enables a user to identify whether sufficient memory is available for the entity being downloaded, thus, avoiding a potential printer memory error (20 MEM OVERFLOW) caused by a download.

There are three commands used for a memory request: Free Space command, Flush All Pages command, and the Echo command. The Free Space command may be used alone or with either or both, of the other two commands. The table below shows the flow for a memory request. For detailed information regarding the memory status request, refer to the individual command descriptions provided later in this chapter.

Operation	Parameters	Example PCL - Command	Example Comments
Send Echo command to identify the beginning of my status response (if multiple user's requesting status on printer port)	Specify any number within range (-32767 to 32767)	**EC*s321X**	321 - user number selected at random

Clear memory (stop receiving data and process existing page data from memory) - optional	All full pages all page data	E_C*sl0B	A command value field of "0" informs the printer to process completed pages from user memory
Request available memory (Free Space command)	N/A	E_C*sℓM	Printer returns the available user memory
Echo my identification number to identify the end of my status response	Any number within range	E_C*s322X	Selected 322

Returned Status

```
PCL <cr><lf>
ECHO 321<cr><lf>
PCL<cr><lf>
INFO MEMORY<cr><lf>
TOTAL=100000<cr><lf>
LARGEST=25000<cr><lf>
PCL<cr><lf>
ECHO 322<cr><lf>
<ff>
```

100,000 bytes of memory currently available with the largest continuous free block being 25,000 bytes.

Status Response Syntax

All status responses start with the letters "PCL" followed by a <cr> (carriage return character ASCII 13) and <lf> (line feed character ASCII 10<F0L>) control code. In addition, status responses are terminated by a <ff> (form feed character ASCII 12) control code. Also, each line of the response is also terminated by a carriage-return and line-feed control code. The basic syntax for the status response is shown below.

```
PCL <cr><lf>
INFO TITLE <cr><lf>
KEYWORD1=DATA1 <cr><lf>
KEYWORD2=DATA2 <cr><lf>
<ff>
```

TITLE, KEYWORD, and DATA are strings that vary depending on the particular status readback command being executed. Each status readback response has one or more keyword lines associated with it.

NOTE Keywords may be added for future printers that are not documented here. Applications that use status readback should be designed to ignore lines with keywords that they do not recognize.

A couple of status responses are shown below.

```
PCL<cr><lf>
INFO MEMORY<cr><lf>
TOTAL=100000<cr><lf>
LARGEST=25000<cr><lf>
<ff>

PCL <cr><lf>
INFO FONTS <cr><lf>
SELECT="<Esc>(8U<Esc>(s0p10.00h12.00v0s0b3T"<cr><lf>
SELECT="<Esc>(0N<Esc>(s0p16.67h8.5v0s0b0T"<cr><lf>
SELECT="<Esc>(s1p__v1s0b4101T<Esc>(78X"<cr><lf>
SYMBOLSETS="8U,0N,10U,11U,12U,1U"<cr><lf>
SELECT=<Esc>(8U<Esc>(s1p__v0s3b4148T<Esc>(78X"<cr><lf>
<ff>
```

Set Status Readback Location Type

The Set Location Type command sets the status location type to the specified value. If a value outside the range is received, the location type is set to zero. A value field of 0 specifies an invalid location. When the location type is 0 and an Inquire Entity command is received, an error response will be generated. A printer reset returns the location type setting to 0. The printer retains the location type setting. If this command is not sent to change the setting for an entity request then the current location type setting is used.

$^{E}{}_{C}$*s# T

Default 0
Range 0 - 5, 7 (unspecified values set location type to 0)

Value	Location
0	Invalid Location
1	Currently selected
2	All locations
3	Internal
4	Downloaded
5	Cartridge
7	User installable ROM device (SIMM)

Set Status Readback Location Unit

The Set Location Unit command sets the status location unit to the specified value. The unit value is interpreted differently, depending on the location type specified.

$^{E}{}_{C}$*s# U

Value = 0 - All
Default = 0
Range = 0 - 32,767

A value of 0 indicates all units of the current location type. A reset sets the location unit to its default value, 0. The printer retains the location unit setting. If this command is not sent to change that setting for an entity request, the current location unit setting is used.

The location type and unit may be set in any order. Invalid combinations are not determined until the Inquire Entity command is received. Therefore, even if the unit value is out of range, the unit is set to that value so that an appropriate error response will be sent when the Inquire Entity command is received.

Type	Unit	Description
1	*	Currently selected
2	*	All locations
3	0	All internal
4	0	All downloaded
4	1	Temporary downloaded
4	2	Permanent downloaded
5	0	All cartridges
5	1	Highest priority cartridge
5	2	Lowest priority cartridge
7	0	All user installable ROM devices
7	1 - 4	User installable ROM device 1, 2, 3, 4

* For location type values 1 and 2 the unit value is ignored and can be set to any value.

Inquire Status Readback Entity

The Inquire Entity command causes the printer to send back the status for all entities of the specified type in the current status readback location.

$^{E}{}_{C}$*s# I
Default 0
Range 0 - 3

Value	Description
0	Font
1	Macro
2	User defined pattern
3	Symbol set
4	Font extended

Status Responses

The status response for an Inquire Status Readback Entity command varies depending on the type of entity requested. The status responses for font, font extended, macros, user-defined patterns, and symbol sets are described below.

Font

A status request for fonts provides different types of information, depending on the request, the font type, and the location. All possible font status information keywords are listed below and described on the following pages.

SELECT=
SYMBOL SETS=
LOCTYPE=
LOCUNIT=
DEFID=
NAME=

An example inquire entity font (value field 0) status response is shown below. The five fonts returned in this example are listed (in order) below:

Internal bitmap (Courier) font
Internal bitmap (Line Printer) font
Internal unbound scalable (CG Times) font
Downloaded bound scalable (CG Palacio) font
Downloaded bound scalable (Dom Casual) font

NOTE ☞ The examples in the remainder of this chapter will not show the PCL status readback response line termination (<cr><lf>) control code or the termination (<ff>) control code

Inquire entity fonts status response example:

```
PCL
INFO FONTS
SELECT="<Esc>(8U<Esc>(s0p10.00h12.0v0s0b3T"
SELECT="<Esc>(0N<Esc>(s0p16.67h8.5v0s0b0T"
SELECT="<Esc>(s1p__v1s0b4101T"
SYMBOLSETS="8U,0N,10U,11U,12U,1U"
SELECT="<Esc>(1U<Esc>(s1p__v0s0b4111T<Esc>(21X"
SELECT="<Esc>(1U<Esc>(s1p__v0s3b4148T<Esc>(22X"
```

In the above example five fonts are identified by the "SELECT=" keyword. This keyword varies slightly, depending on whether the font is bitmap, bound scalable, unbound scalable, currently selected, or downloaded, as described under "SELECT=" below. Also, additional keywords are added for unbound fonts ("SYMBOLSETS="), for "currently selected" font request ("LOCTYPE=" and "LOCUNIT="), and for font extended requests ("DEFID=" and "NAME=").

NOTE ☞ In the status response an escape character is represented by the five characters: <Esc>* All keywords (other than "SELECT=") following the "SELECT=" keyword, apply to the previous "SELECT=" keyword.

SELECT= is returned for all fonts. It identifies the font by specifying the font selection characteristics (attributes). The characteristics are listed as they would be sent to the printer to select the font (that is, in priority order, highest priority first). For downloaded (soft font) fonts the PCL ID selection number is also included at the end (for example, an ID number of 27 would be presented as <Esc>(27X).

For bitmap fonts, all font selection characteristics are listed. In addition, as mentioned above, if the bitmap font is a soft font then its PCL ID number is also included at the end of the "SELECT=" keyword.

For scalable fonts, only the relevant characteristics are listed. Symbol set is included for bound scalable fonts (for unbound fonts an additional keyword, -SYMBOLSETS=, is added). Orientation is not listed and only one of height or pitch is listed, depending on whether the font is proportional or fixed. Two underscores indicate that the font is scalable.

SYMBOLSET= is added for unbound scalable fonts. (That is, a font for which any number of symbol sets may be selected.) This keyword lists all the symbol sets which can be bound to the font.

If currently selected status is requested then two additional keywords are included: "LOCTYPE=" and "LOCUNIT=." Also, if the currently selected font is a scalable font, the "SELECT=" keyword shows the current symbol set and current size (point size or pitch), since these values have been selected. In addition, if the currently selected font is a secondary font, then the "(" character is replaced by ")."

LOCTYPE= is returned for a currently selected font request only. It identifies the location type of the currently selected font.

LOCUNIT= is returned for currently selected font only. It identifies the location unit of a currently selected font request only.

An example for a font response for the currently selected font might appear as shown below.

```
PCL
INFO FONTS
SELECT="<Esc>(8U<Esc>(s0p18.00h20.0v0s0b4101T"
LOCTYPE=3
LOCUNIT=1
```

Font Extended

The font extended (inquire entity value field 4) response is the same as a font (inquire entity value field 1) status request but with two additional lines: DEFID= and NAME=.

DEFID= identifies the printer font number which can be used to select the font as the default font from either the printer's control panel or from PJL (refer to the Job Language Technical for information on PJL font selection information). The number consists of two parts: A location and an ID number, such as: "I 21." The possible locations are:

I	Internal
C	Cartridge
S	Permanent soft fonts
Mn	SIMM where n is the number of the SIMM slot for example: M2 is SIMM in #2 slot.
NONE	Temporary* soft fonts

* Temporary fonts cannot be selected as the default from the control panel or PJL; they do not have a printer font ID number.

* The printer font ID number is assigned to fonts by the printer. This number is different than the soft font ID number assigned to a downloaded using the Font ID ($^{E}{}_{C}$***c#D**) command.

NAME= is returned for font extended status requests only. It identifies the name of the font, such as, Courier, Times Roman, Univers, etc. and its treatment (BdIt - bold italic) as listed in the font printout listing.

An example for font extended status response is shown below.

```
PCL
INFO FONTS EXTENDED
SELECT="<Esc>(8U<Esc>(s0p10.00h12.0v0s0b3T"
DEFID="I 21"
NAME="Courier"
SELECT="<Esc>(s1p__v1s3b4101T"
SYMBOLSETS="8U,0N,10U,11U,12U,1U"
DEFID="I 45"
NAME="CG Times  BdIt"
SELECT= . . .
                                                              .
                                                              .
                                                              .
```

An example for a font extended response for the currently selected font might appear as shown below.

```
PCL
INFO FONTS EXTENDED
SELECT="<Esc>(8U<Esc>(s0p18.00hv0s3b4099T"
DEFID=I 21
NAME=Courier Bd
LOCTYPE=3
LOCUNIT=1
```

Macros

The status response for macros lists all of the macro IDs ("IDLIST=") for the macros in the specified location.

NOTE ☞ Status readback location type 1 (currently selected) is an invalid location for macros.

A macro status response might appear as shown below.

```
PCL
INFO MACROS
IDLIST="1,8,3,29,32"
```

User Defined Patterns

The status response for user-defined patterns lists all of the user-defined pattern IDs ("IDLIST=") for the patterns in the specified location.

A user-defined pattern response might appear as shown below.

```
PCL
INFO PATTERNS
IDLIST="9,27,2,456,13,1"
```

If the location type is set to 1 (currently selected) then "LOCTYPE=" and "LOCUNIT=" lines are added to identify the location (type and unit) of the pattern.

A user-defined pattern response for the currently selected pattern might appear as shown below.

```
PCL
INFO PATTERNS
IDLIST="88"
LOCTYPE=4
LOCUNIT=2
```

If the current pattern is set to one of the internal HP-defined patterns (assign no pattern ID number) then no number is available and the error response, "ERROR=NONE" will be returned

```
PCL
INFO PATTERNS
ERROR=NONE
```

Symbol Set Response

The response for symbol sets (entity value field 3) lists all of the symbol set IDs ("IDLIST=") for all the symbol sets for all unbound fonts in the specified location (type and unit).

A symbol set response might appear as shown below.

```
PCL
INFO SYMBOLSETS
IDLIST="8U,0U,2K,11U,8M"
```

Status Readback location type 1 (currently selected) is an invalid location for unbound font symbol sets and will return an error ("ERROR=NONE").

Status Response Error Codes

If the user requests out of range values in the command value fields, or if the entity is unsupported or does not exist, or if the request is inappropriate, the printer responds with an error.

If the entity type specified is out of range or unsupported, an invalid entity error will be returned. For example, if the Inquire Entity command contained an out of range value of 8 (**E$_C$*s8I**), the following error response would be generated.

```
PCL
INFO ENTITY
ERROR=INVALID ENTITY
```

If the entity type is valid but the location (either the type, the unit, or the combination) is invalid; or, the specified device is not installed, an invalid location error will be returned. For example, if you requested a status for a cartridge type but identified an out of range location unit of 9 (*s4t9U), the following error response would be generated:

```
PCL
INFO FONTS
ERROR=INVALID LOCATION
```

If the entity type and location are valid but there are no entities of the specified type in that location; or, if the type is inappropriate for the specified entity (that is, internal user-defined pattern or currently selected macro) then an error response will be generated. For example, if the user requests the downloaded symbol sets and there are no downloaded unbound fonts (multiple symbol set maps per font), the following error response would be generated:

```
PCL
INFO SYMBOLSETS
ERROR=NONE
```

The status response for some requests can be fairly large (such as for fonts). In processing status responses, if the printer runs out of memory, an internal error will be returned, as shown below.

```
PCL
INFO SYMBOLSETS
ERROR=NONE
```

The error conditions described above are the only conditions for which an error response is generated. If the user makes a syntax error in the escape sequence, sends a command which the printer cannot interpret, the printer will ignore the command and no error response will be given.

Free Space Command

The Free Space command returns the amount of available internal (user) memory.

$^E{}_C$*s1M

Default= n/a
Range= 1

The user can affect the amount of memory available in the printer by using either the Flush All Pages command or the Printer Reset command. If these commands are sent before the Free Space command the resulting available memory returned by this response can be larger.

Making More Memory Available

Sending a Flush All Pages command prior to the Free Space command causes the printer to stop receiving data while it processes and prints that which is currently in memory. This allows the user to identify how much memory is currently available without page data (refer to the Flush All Pages command described on the following pages for additional information).

A printer reset can be used to clear some data from memory. Performing a printer reset causes the printer to delete all temporary entities (fonts, macros, user-defined patterns). In addition, it deletes the font cache.

Font Cache

To print characters from a scalable font the printer converts the scalable character outlines into sized, bitmaps. Initially, when a scalable font is selected, no bitmap characters are created. Bitmaps are created on a character by character basis, as they are needed for printing. As more and more pages are printed using more fonts, these bitmaps consume more and more memory. The bitmaps for a font used on the first page of a job can remain in memory until the end of a job. (If there is insufficient room for page processing the printer will automatically delete the font cache bitmap characters not used for the current page.)

Status Responses

A free space status response returns two values:

```
TOTAL=
LARGEST=
```

TOTAL= identifies the total available user memory (in bytes).
LARGEST= identifies the largest continuous block of available memory (in bytes).

An example response is shown below:

```
PCL
INFO MEMORY
TOTAL=100000
LARGEST=25000
```

The above example indicates that the printer has 100,000 bytes of available memory with the largest continuous block being 25,000 bytes.

Status Response Error Codes

If the Free Memory command value field is out of range (not 1) then the memory status response will return an invalid unit error. For example, if the Free Space command with a value of 2 ($^{E}_{C}$***s2M** were sent, the following error would be returned:

```
PCL
INFO MEMORY
ERROR=INVALID UNIT
```

Flush All Pages

The Flush All Pages command suspends processing of the input data stream until all pages currently in the printer have been printed.

$^{E}_{C}$**&r#F**

# = 0 - Flush all complete pages
= 1 - Flush all pages

Default = 0
Range = 0 - 1

A value of 0 indicates that only complete pages are to be processed. If a partial page exists, it will not be processed. A value of 1 indicates that all page data including the partial page will be processed.

For example if the printer contains two complete pages (page "A" and page "B") and one partial page (page "C"), and receives a flush all complete pages (value field 0), it will eject pages "A" and "B" and retain page "C." If the printer received the flush all pages (value field 1) it will process and eject pages "A," "B," and "C."

The printer resumes receiving (processing) data when the last page is processed and ejected from the paper path.

Echo Command

The Echo command echoes its value field (in ASCII format) back to the host.

$^{E}_{C}$***s# X**

# = Echo value (ASCII)
Default = 0
Range = -32767 to 32767

If multiple users are requesting status, it can be difficult to distinguish one user's status response from another. The Echo command provides the means to label status responses. Since the value the user selects for the value field is returned, this command can be used as a user identification mark or "place holder." By sending the Echo command, with some specific value field number, at the beginning and end (or the beginning or end only) of a status request, enables the user to delineate their status response data with their own echo identification number.

* Status readback requests are processed in the order they are received.
* Status readback responses are returned to the printer port from which they were received.

Echo Response

The Echo command returns the following response:

PCL
ECHO <value field>

where <value field> is the Echo command value field value that the user selected (within the range -32767 to 32767).

For example, if the Echo command, $^{E}{}_{C}$***s-999X** was sent, the status response would be:

PCL
ECHO -999

Printer Job Language

Hewlett-Packard's Printer Job Language (PJL) was first introduced with the LaserJet IIISi printer to provide an effective and reliable mechanism for selecting a printer language from a multilingual peripheral device. PJL continues to offer this level of functionality and has been enhanced to provide an even higher level of device control.

Syntax Rules

The following symbols and conventions will be used in this section:

Symbol	Value	Description
<>		Indicates syntactic category
::=		"To be written as" symbol
|		"Pipe" separates choices
{}$_{1}$		Choose 1 of the enclosed items
{}$_{0+}$		Repeat the enclosed items 0 or more times
{}$_{1+}$		Repeat the enclosed items 1 or more times
{}$_{opt}$		Optional items
<bs>	ASCII 8	Backspace character
<ht>	ASCII 9	Horizontal tab
<lf>	ASCII 10	Linefeed
<ff>	ASCII 12	Formfeed character
<cr>	ASCII 13	Carriage return character
<esc>	ASCII 27	Escape character
<sp>	ASCII 32	Space character
<double quote>	ASCII 34	Double quote character
<decimal point>	ASCII 46	Decimal point (period) character
<ws>	{<sp>|<ht>}$_{1+}$	White space
<eol>	{<ws>}$_{opt}$ {<cr>}$_{opt}$ <lf>	End of line
<symbol>	! # $ % & ' ( ) * + - . / : ; < = > ? @ [ ] \ ^ _ ` { } |	
<digit>	ASCII 48-57	Number characters
<number token>	{+/-}$_{opt}$ {<digit>}$_{1+}$ {<decimal point> {<digit>}$_{0+}$}$_{opt}$	
<string character>	ASCII 65-90 97-122 48-57 127-255 <white space> <symbol>	Characters allowed in strings
<string>	<double quote> {<string character>}$_{0+}$ <double quote>	Character string
<string token>	<string>	A minimum of 80 characters are significant

Kernel Commands

The PJL kernel commands are those commands which are common to all Hewlett-Packard peripherals supporting Printer Job Language.

UEL/SPJL

$^{E}{}_{C}$%-12345X

When the Universal Exit Language/Start of PJL command is received, it causes the current printer language to terminate and return control to PJL.

In addition to causing the current printer language to shut down and return control to PJL, this command performs a reset. If the printer is in PCL, a PCL printer reset (**$^{E}{}_{C}$E**) is performed; if the printer is in PostScript, a PostScript End-of-Job (Cntl-D) command is performed.

NOTE ☞

All jobs, both PCL and PostScript, should start and end with the UEL/SPJL command.

The UEL/SPJL command must be immediately followed by the "@PJL" PJL command prefix. If any characters or control codes (such as <CR or <LF>) other than the PJL command prefix are received, the printer will enable the default language (if one has been selected on the printer's control panel) or will attempt to auto-select a language to process the incoming data.

Enter Language

@PJL ENTER LANGUAGE = {PCL | POSTSCRIPT}$_1$<eol>

The Enter Language command causes the printer to enable the specified printer language

If the printer does not receive the Enter Language command requesting a specific language, the printer will either process the job in the default language if one has been selected or will attempt to determine the appropriate language for the print job by sampling the data stream. For the best possible performance and highest level of reliability, the language to be used for a print job should be explicitly requested.

Comment

@PJL COMMENT <Words> <eol>

The Comment command allows a line of information to be added as a comment. Comments are ignored by the PJL parser.

Job Control Commands

Job commands provide control over specific aspects of printing. For example, the job command supports the printing of specific pages from a print job. Job commands also allow feedback regarding a particular job's progress when using status read back.

Job Command

@PJL JOB {NAME=<job name> | START=<first page> | END=<last page> }$_{0+}$ <eol>

The job command is used to inform the printer of the start of a print job and to synchronize the job and page status information. When the printer receives this command it will reset the page count associated with the unsolicited page status and return a job status message if job status is enabled. Resetting the page count associated with unsolicited page status only affects future pages. Pages that have already been processed, but not yet printed, are not affected. <first page> and <last page> are of type <number token>.

Job Command Options

The following command options are available for use with the PJL JOB command. You may use one or more of the following options.

Option	**Description**
NAME	Allows the print job to be tagged with a name. <job name> is of type <string token>.
START	Used to indicate the first page that will be printed. If omitted, the printer will start printing at the beginning of the job. <first page> is of type 32-bit signed (1 to 2,147,483,647).
END	Used to indicate that last page of the job to print. If omitted, the printer will print to the end of the job. <last page> is of type 32-bit signed (1 to 2,147,483,647).
PASSWORD	Sets the default password for PJL security. Ranges are { 0 | ... | 65535}. PJL security is disabled if the value is zero.

EOJ Command

@PJL EOJ {NAME=<job name>}$_{opt}$ <eol>

The EOJ Command is used to indicate to the printer when the print job has completed.

Option	Description
NAME	Allows the print job to be tagged with a name. <job name> is of type <string token>.

NOTE ☞ For every JOB command issued, there must be a corresponding EOJ command.

Environment Commands

By allowing PJL to set default values along with the control panel and the data stream for the switched personality, the priority of defaults can become confusing. The control panel values have the lowest priority and will only be used in the event that neither the PJL nor the data stream specifies a default. PJL has the next highest priority and can only be overridden by the personality data stream value, which has the highest priority.

Reset Command

@PJL RESET

Resets the PJL current variables to their default values (stored in NVRAM), as would happen if the printer were power cycled. This command does not affect the unsolicited status settings.

Initialize Command

```
@PJL INITIALIZE
```

Resets the PJL variables, current and default, to their factory default settings. This effectively initializes the printer parameters to their cold start values, except I/O configuration values. This command does not affect the unsolicited status settings.

Setting Environment Variables

Current Variables - @PJL SET <variable> = <value>
Default Variables - @PJL DEFAULT <variable> = <value>

When setting the environment variables, the user must decide whether to change the current environment variables or the default environment variables. The difference is that changes to the current environment values affect only the current job. Changes to the default environment variables are written to NVRAM (this has the same effect as making a front panel change and changes the front panel display) and affect future jobs.

NOTE As a general rule applications should not change the default environment variables.

The following environment variables may be set through PJL. It is the responsibility of each personality to use these values if appropriate to the personality as the initial default value. The actual list of environmental values is printer specific; each printer may need to add or subtract from this list to properly support its feature set.

Printing Menu

The following list specifies the set of LaserJet 4Si printer specific environment variables and values corresponding to the printing menu on the front panel. These variables are reset to their factory default values by the @PJL INITIALIZE command.

Variable	Values	Description
BIND	LONG/SHORT	Sets the default relationship of the font and back pages of a duplexed page.
CPLOCK	ON/OFF	Sets the default control panel lockout state. The factory default is OFF.
COPIES	1 - 999	Sets the default number of copies for each page of the job. The factory default is 1. Integer values less than 1 will be snapped to 1. Integer values greater than 999 will be snapped to 999. Floating point values will be truncated; if the resulting integer is out of range, the integer value will be snapped to the appropriate limit.
DUPLEX	ON/OFF	Sets the default duplex mode.
FORMLINES	5 - 128	Sets the default number of lines per page. The factory default is 60.
JOBOFFSET	ON/OFF	Sets the defualt job offset state.
MANUALFEED	ON/OFF	Sets the manual feed default mode for the job. The factory default is OFF.
OUTBIN	UPPER LOWER	Sets the defualt output tray. The Upper Bin is the face down bin. The Lower Bin is the face up bin.
ORIENTATION	PORTRAIT LANDSCAPE	Sets the default page orientation for the job. The factory default value is PORTRAIT.
PAPER	LETTER, LEGAL, EXECUTIVE, A4, COM10, MONARCH, C5, DL, B5	Sets the default paper size for the job. The factory default is LETTER.
RET	ON/OFF	Sets the default configuration for resolution enhancement hardware.

Job Menu

The following list specifies the set of LaserJet 4Si specific environment variables corresponding to the job menu on the front panel. These variables are reset to their factory defaults by the @PJL INITIALIZE command.

Variable	Values	Description
PAGEPROTECT	OFF, LETTER, LEGAL, A4	Sets the Default page protection configuration. The factory default is OFF.
PERSONALITY	AUTO PCL POSTSCRIPT	Sets the default personality for implicit switching.
RESOLUTION	600/300	Sets the default resolution for the job. The factory default value is 600. Integer values less than 300, will be snapped to 300. Integer values greater than 600 will be snapped to 600. Integer values between 300 and 449, inclusive, will be snapped to 300. Integer values between 450 and 599, inclusive, will be snapped to 600. Floating point values will be truncated; if the resulting integer is not 300 or 600, the integer value will be snapped to the appropriate limit as described above.
TIMEOUT	5 - 300	Sets the default timeout for IO in seconds

NOTE ☞ These variables can be set to any legal value at any time, regardless of the current amount of free memory or currently set resolution. When a job is sent, if there is not enough memory to run with the current resolution and page protection configuration, the system will temporarily override the resolution and/or page protect values in order to allow the job to run.

Setting Personality Dependent Variables

Current Variables - @PJL SET LPARM: <personality> <variable> =<value>
Default Variables - @PJL DEFAULT LPARM: <personality> <variable> =<value>

When setting the personality dependent variables, the user must decide whether to change the current personality dependent variables or the default personality dependent variables. The difference is changes to the current variables affect only the current job. Changes to the default personality variables affect future jobs. Valid values for <personality> are dependent on the installed personalities in the device. These values include PCL and POSTSCRIPT.

PCL Menu

The following list specifies the current set of PCL personality specific variables which LaserJet 4Si supports.

Variable	**Value(s)**
FONTSOURCE	I - Internal M1 M2 M3 M4 - ROM SIMM C - Cartridge S - Soft
FONTNUMBER	0, 1, 2 ... *n*
PITCH	0.44, 0.45, ... 99.99
PTSIZE	4.00, 4.25, ... 999.75
FORMLINES	5, 6, ... 128
SYMSET	ROMAN8, ISOL1, ISOL2, ISOL5, PC8, PC8DN, PC850, PC852, PC8TK, WINL1, WINL2, WINL5, DESKTOP, PSTEXT, VNINTL, VNUS, MSPUBL, MATH8, PSMATH, VNMATH, PIFONT, LEGAL, ISO4, ISO6, ISO11, ISO15, ISO17, ISO21, ISO60, ISO69, VNDING, PSDING, DING1, DING2, DING3

These variables are reset to their factory defaults by the @PJL INITIALIZE command.

Font Source Variable

FONTSOURCE = { I | M*n* | C | S }$_1$

Sets the default font source for the job. The factory default value depends on the font sources currently installed. If a cartridge with a default marked font is installed, the default value is C. Otherwise, if a ROM SIMM with a default marked font is installed, the default font source is the ROM SIMM. Lower numbered units have higher priority over higher numbered units, so M1 is the higher priority ROM SIMM. Otherwise the default is I.

The valid values for this variable depend on the currently installed font base. Only locations which contain one or more fonts are legal values. If a font source is specified which does not at the time contain any fonts, then the value is ignored, and a 7-001 semantic error is produced.

Certain events cause the value of this variable to be automatically changed. If the currently set font source is C and the cartridge is removed from the printer, then the value of this variable is automatically changed to the factory default. If the currently set font source is S and all soft fonts are deleted, then the value of this variable is automatically changed to the factory default value. Also, if the currently set font source is S, the currently set font number is the highest-numbered soft font, and any soft font is deleted, then the value of this variable is automatically changed to the factory default value.

Font Number Variable

FONTNUMBER = {0 | 1 | ... | *n*}$_1$

Sets the default font number for the job. The factory default value is 0.
The valid range for this variable depends on the currently set font source. If the font source contains a default marked font, the values start at 0; otherwise, the values start at 1. The upper limit depends on the number of installed fonts in the font source.

Integer values outside the valid range will be ignored. Floating point values will be truncated; if the resulting integer is out of range, the value will be ignored. If the value is ignored, then a 25-014 error will be produced.

This variable is tied to both the FONTSOURCE and SYMSET variables. If the value of the FONTSOURCE variable changes, then the FONTNUMBER variable is automatically changed to the lowest numbered font in the new font source (either 0 or 1, depending on whether the new font source contains a default marked font). If the value of the SYMSET variable changes, then the FONTNUMBER variable is automatically changed to the lowest numbered font in the current font source (either 0 or 1).

Also, if the currently set font source is S, the currently set font number is the highest-numbered soft font, and a soft font is deleted, then the FONTSOURCE variable is changed to its factory default value, and the value of the -FONTNUMBER variable is changed to the lowest numbered font in the new font source (either 0 or 1).

Pitch Variable

PITCH = {0.44 | 0.45 | ... | 99.99}$_1$

Sets the default pitch of the default font, if the default font is a fixed pitch scalable font in units of characters per inch. The factory default is 10.00.
The pitch value can be specified to two decimal places. Any additional digits beyond two decimal points will be truncated. If the value is less than 0.44, the value will be snapped to 0.44. If the value is greater than 99.99, the value will be snapped to 99.99.

Point Size Variable

PTSIZE = {4.00 | 4.25 | ... | 999.75}$_1$

Sets the default height of the default font, if the default font is a proportional scalable font, in units of points. The factory default is 12.00.

The point size value is significant only to a quarter of a point, If the value is not a multiple of 0.25, the value will be rounded down to the nearest multiple of 0.25. If the value is less than 4.00, the value will be snapped to 4.00. If the value is greater than 999.75, the value will be snapped to 999.75.

Form Lines Variables

FORMLINES = {5 | ... | 128}$_1$

Sets the default number of lines per page for the job. The factory default is 60.

Integer values less than 5 will be snapped to 5. Integer values greater than 128 will be snapped to 128. Floating point values will be truncated; if the resulting integer is out of range, the integer value will be snapped to the appropriate limit.

This variable is tied to the PAPER and ORIENTATION variables. If the value of either of those variables is changed, then the FORMLINES variable is automatically updated to maintain the same line spacing.

Symbol Set Variable

SYMSET = {ROMAN8 | ISOL1 | ISOL2 | ISOL5 |PC8 | PC8DN |PC850 | PC852 | PC8TK | WINL1 | WINL2 | WINL5 |DESKTOP | PSTEXT | VNINTL | VNUS | MSPUBL | MATH8 | PSMATH |VNMATH | PIFONT | LEGAL | ISO4 | ISO6 | ISO11 | ISO15 | ISO17 |ISO21 | ISO60 | ISO69 | VNDING | PSDING | DING1 | DING2 | DING3}$_1$

Sets the default symbol set for the job. The factory default value is ROMAN8.

Status Readback Commands

PJL supports printer to host status readback. It is not inconsistent for the individual personalities to provide personality dependent status readback if - appropriate, and even to duplicate some of the PJL status readback functionality.

All PJL status readback requests will generate an appropriate response. All PJL status readback responses from the printer will start with @PJL and will be terminated with <ff><F0>. By using <ff>to terminate the response, it is possible to have multiple line responses.

Inquire Variables

The INQUIRE and DINQUIRE commands are used to inquire either the current or the default value for a variable that may be set through PJL.

The general case syntax for use of the inquire variables is:

@PJL {INQUIRE | DINQUIRE}$_1$ {LPARM: <personality>}$_{opt}$ <variable><eol>
The general case syntax for the returned information is:
@PJL {INQUIRE |DINQUIRE}$_1$ {LPARM: <personality>}$_{opt}$ <variable><cr><lf>
<value><cr><lf>
<ff>

If the printer does not support the <variable> or the <personality> or the <variable> is missing, the response will be:

@PJL {INQUIRE |DINQUIRE}$_1$ {LPARM: <personality>}$_{opt}$ <variable> <cr><lf>
"?"<cr><lf>
<ff>

Inquire Environment Current Variables

@PJL INQUIRE {LPARM : <personality>}$_{opt}$ <variable><eol>

Example: You have used the @PJL SET COPIES = 5 command and wish to verify it by using status readback.

The command request will have the form:

@PJL INQUIRE COPIES<eol>

The returned information should be of the form:

@PJL INQUIRE COPIES<cr><lf>
5<cr><lf>
<ff>

Inquire Environment Default Variables

@PJL DINQUIRE <variable><eol>

If you had used the @PJL DEFAULT RESOLUTION = 600 command and wished to verify it by using status readback, the status readback command syntax would be:

@PJL DINQUIRE RESOLUTION<eol>

The returned information would be:

@PJL DINQUIRE RESOLUTION<cr><lf>
600<cr><lf>
<ff>

Inquire Personality Dependent Current Variables

Command syntax:

@PJL INQUIRE LPARM: <personality> <variable> <eol>

Syntax of printer response:

```
@PJL INQUIRE LPARM: <personality> <variable> <cr><lf>
<value><cr><lf>
<ff>
```

Inquire Personality Dependent Default Variables

Command syntax:

```
@PJL DINQUIRE LPARM: <personality> <variable><eol>
```

Syntax of printer response:

```
@PJL DINQUIRE LPARM: <personality> <variable> <cr><lf>
<value><cr><lf>
<ff>
```

Product Specific Information

Besides the previously mentioned environment variables which may be set programmatically, there exists a set of read only environment information variables.

Identification

This command causes the printer to return a string identifying the device. To maintain forward compatibility with future printers, applications should not use the ID to identify printer features. The more detailed CONFIG should be used.

Command syntax:

```
@PJL INFO ID <eol>
```

Syntax of printer response:

```
@PJL INFO ID <cr><lf>
<string><cr><lf>
<ff>
```

Configuration

Command syntax:

```
@PJL INFO CONFIG <eol>
```

Syntax of printer response:

```
@PJL INFO CONFIG <cr><lf>
{<feature>{=<value> | <option number>}opt <cr><lf>
{<ht><returned option>{ <attribute}opt <cr><lf>}0+}1+
<ff>
```

Memory

Command syntax:

@PJL INFO MEMORY <eol>

Syntax of printer response:

@PJL INFO MEMORY <cr><lf>
{<feature>=<value><cr><lf>}$_{1+}$
<ff>

The INFO MEMORY command will return a response similar to the following.

INFO MEMORY
<F0>TOTAL=1000000
LARGEST=250000

The actual numbers will depend on how much memory is available and the size of the largest free block.

The TOTAL number will not include any memory allocated for strips or frame buffers. This implies that the amount of memory available to a personality will be less than this value, and the LARGEST number returned may not be the largest free block when a personality is invoked.

Status

Command syntax:

@PJL INFO STATUS <eol>

Syntax of printer response:

@PJL INFO STATUS <cr><lf>
CODE=<category><status code> <cr><lf>
{<feature>=<value><cr><lf>}$_{1+}$
<ff>

If the printer supports the status variable, the format of the returned information will be:

@PJL INFO STAT =
<F0>CODE=10000
DISPLAY = "00 READY"
ONLINE = TRUE

Variables

This command causes the printer to return a series of lines listing the environmental and personality dependent variables. The listing will also contain the possible values that can be set and the current settings, (i.e. not the default settings).

Command syntax:

```
@PJL INFO VARIABLES <eol>
```

Syntax of printer response:

@PJL INFO VARIABLES <cr><lf>
@PJL INFO CONFIG <cr><lf>
{<variable>{=<value>}$_{opt}${ <option number>}$_{opt}$ <cr><lf>
{<ht><returned option>}$_{0+}$}$_{1+}$ <cr><lf>
<ff>

Ustatus

This command causes the printer to respond with a series of lines listing the types of unsolicited status supported by the printer. The listing will also contain the possible values that can be set and the current setting.

Command syntax:

```
@PJL INFO USTATUS <eol>
```

Syntax of printer response:

@PJL INFO USTATUS <cr><lf>
{<variable>{=<value>}$_{opt}${ <option number>}$_{opt}$ <cr><lf>
{<ht><returned option>}$_{0+}$}$_{1+}$ <cr><lf>
<ff>

A sample of how the returned data would be formatted follows:

```
@PJL INFO USTATUS <cr><lf>
<F0>DEVICE=OFF [3 ENUMERATED] <cr><lf>
<ht>OFF <cr><lf>
<ht>ON <cr><lf>
<ht>VERBOSE <cr><lf>
JOB=OFF [2 ENUMERATED] <cr><lf>
<ht>OFF <cr><lf>
<ht>ON <cr><lf>
PAGE=OFF [2 ENUMERATED] <cr><lf>
<ht>OFF <cr><lf>
<ht>ON <cr><lf>
TIMED=0 [2 RANGE] <cr><lf>
<ht>5 <cr><lf>
<ht>120 <cr><lf>
<ff>
```

The current setting for each type of unsolicited status will be specified in the response.

ECHO

PJL will include an echo command. The echo command will cause PJL to return the comment section, when received, back to the host computer. The echo command can be used to synchronize the printer with the host computer.

@PJL ECHO{ <words>}*opt* <eol>

The <words> are ignored by the PJL parser and a minimum of the first 80 characters are returned to the host as:

@PJL ECHO { <words>}*opt* <cr><lf>
<ff>

Unsolicited Status

The previous status request are all of the solicited type. For each solicited request, the printer will return an appropriate response when it processes the request. Unsolicited status is enabled or disabled when the USTATUS command is processed. The actual status message will be sent when the status that was enabled changes. For example, setting unsolicited device status to ON will cause an unsolicited status message to be sent each time the device status changes. Example events that change the device status includes doors opening, paper jamming, and running out of paper.

The different types of unsolicited status from the printer may be enabled/disabled as required by the host with the following command:

@PJL USTATUS <variable>=<value> <eol>

Where <variable> and <value> are specified below.

The following sections specify the types of unsolicited status supported by the LaserJet 4Si printer.

Device Variable

When set to ON the printer will send unsolicited status whenever the status of the device changes. When set to verbose the device will send unsolicited status for all categories of errors.

To turn off device status the command would appear as:

@PJL USTATUS DEVICE=OFF <eol>

To turn on device status, the command would appear as:

@PJL USTATUS DEVICE=ON <eol>
or
@PJL USTATUS DEVICE=VERBOSE <eol>

The returned data will be formatted as:

```
@PJL USTATUS DEVICE <cr><lf>
CODE=<category><satus code> <cr><lf>
{DISPLAY=<string><cr><lf>}opt
{ONLINE=<boolean><cr><lf>}opt
<ff>
```

where <category> and <status code> together are a five digit value of the form *ccsss* where *cc* - digits of <category> and *sss* - digit of <status code>).

A sample of how the return data may be formatted is:

```
@PJL USTATUS DEVICE <cr><lf>
CODE=40001 <cr><lf>
DISPLAY="12 PRINTER OPEN" <cr><lf>
<ff>
```

The following table lists the possible values for the two digit category values and their associated descriptions.

Category	**Description**
10	Indicates an informational message only an error condition does not exist. (e.g. device was put online and is ready or device was put offline)
20	Indicates a PJL parser error the entire PJL command was ignored. (e.g. invalid or unsupported language in ENTER command or PJL syntax error)
25	Indicates a PJL parser warning a portion of the PJL command was ignored. (e.g. unsupported option or value out of range)
27	Indicates a semantic error all or part of the PJL command was ignored. (e.g. number of EOJ commands exceeds number of JOB commands)
30	Indicates that an auto-continuable condition has occurred. If no operator action is taken the device will auto-continue if AUTO CONT=ON. If AUTO CONT=OFF the printer will wait until an operator intervenes. (e.g. 21 PRINT OVERRUN or 20 MEMORY OUT)
35	Indicates operator intervention may be required. The printer will stay on-line and continue to operate possibly with reduced functionality. Data may have been lost. (e.g. e.g. 21 PRINT OVERRUN or 20 MEMORY OUT)
40	Indicates that operator intervention is required. Printing will not continue until the condition is resolved. (e.g. paper jam or paper mount request)
50	Indicates a hardware error has occurred. (e.g. 79 SERVICE)

The following tables list many of the possible <category><status code> values which can occur along with the associated printer display string.

Category/Status Code	**Display String**
10001	00READY [online]
10002	00READY [offline]
10001	[Pers. ready mesg][online]
10002	[Pers. ready mesg][offline]
10001	[PJL RDYMSG][online]
10002	[PJL RDYMSG][offline]
10003	02 WARMING UP
10004	04 SELF TEST
10004	05 SELF TEST
10004	06 PRINTING TEST
10004	06 FONT PRINTOUT
10005	07 RESET
10005	09 MENU RESET
10005	09 ACTIVE I/O
10005	09 ALL I/O
10004	15 ENGINE TEST
10006	16 TONER LOW

The following list specifies the set of display strings and the corresponding PJL error codes for device attendance messages:

Category/Status Code	Display String
	10 RESET TO SAVE
40001	12 PRINTER OPEN
40002	13 PAPER JAM
40010	14 NO EP CART
40013	FE CARTRIDGE
40013	FC [Cartridge slot]
40013	FR [Cartridge slot]
40013	FI [Cartridge slot]
40014	MF FEED [Paper size]
40014	ME FEED [Envelope size]
40015	[Tray] LOAD [Paper size]
40015	[Tray] LOAD [Envelope size]
40009	[Tray] TRAY EMPTY

The following list specifies the set of display strings and the corresponding PJL error codes for error messages:

Category/Status Code	Display String
30002	20 MEM OVERFLOW
30001	21 PRINT OVERRUN
40017	22 I/O CONFIG ERR
40018	40 ERROR
*	41 ERROR
*	42 OPT INTERFACE
*	43 OPT INTERFACE
*	68 READY/SERVICE

* These values not currently available

The following list specifies the set of display strings and the corresponding PJL status codes for service messages:

Category/Status Code	Display String
40008	50 SERVICE
*	51 ERROR
*	52 ERROR
*	53 [Unit] ERRORUNIT
*	55 ERROR
*	61 SERVICE
*	62 SERVICE
*	63 SERVICE
*	64 SERVICE
*	65 SERVICE
*	67 SERVICE
*	68 ERROR
*	68 SERVICE
*	69 SERVICE
*	70 ERROR
*	71 ERROR
*	72 SERVICE
*	79 SERVICE(XXXX)

* These values not currently available

The following list specifies the set of PJL parser errors and warnings. These status codes do not have an associated display string.

Category/Status Code	Description
30001	Status buffer overflow
30002	Syntax error
30003	Unsupported/invalid option

Job Variable

When the unsolicited job status variable is set to ON, an unsolicited status message is sent each time a JOB or an EOJ command is encountered. The command to turn ON unsolicited job status would appear as:

```
@PJL USTATUS JOB=ON <eol>
```

The returned data for the start of job will be formatted as:

```
@PJL USTATUS JOB <cr><lf>
START <cr><lf>
{NAME=<job name><cr><lf>}opt
<ff>
```

where the NAME is included if the JOB command included the NAME option. The device returns the unsolicited JOB status when the JOB command is parsed.

The returned data for the end of job will be formatted as:

```
@PJL USTATUS JOB <cr><lf>
END <cr><lf>
{NAME=<job name><cr><lf>}opt
{PAGES=<page count><cr><lf>}opt
<ff>
```

where the NAME is included if the EOJ command included the NAME option.

PAGES is included if the device supports the unsolicited page status variable. The device returns the unsolicited EOJ status when the last page of a job is available to the user.

Since the start of job unsolicited status is synchronized with the parsing of the job command, and the end of job unsolicited status is synchronized with the availability of the last page of the print job, it is possible for the device to send the unsolicited job status information in an order different than the data stream order of the JOB and EOJ commands.

Page Variable

When the unsolicited page status variable is set to ON, an unsolicited status message is sent each time a page has reached the output stacker/tray.
The command to turn ON unsolicited page status is:

```
@PJL USTATUS PAGE=ON <eol>
```

The format of the returned status is:

```
@PJL USTATUS PAGE <cr><lf>
<page number><cr><lf>
<ff>
```

where <page number> indicates the number of pages formatted (includes silent-run pages) since the last JOB command. The number of pages is incremented by one for each simplex page formatted. <page number> has the range of a signed 32-bit integer (1 - 2,147,483,647).

Timed Variable

When the unsolicited time status variable is set to ON, an unsolicited status message is sent at the specified time interval. The unsolicited status will be similar to the response sent when the host sends the INFO STATUS command. The timed feature allows "automatic polling" of the device without the host intervention and without exiting the current language back to PJL. The time interval supported, in seconds, is device dependent but all devices will support a time of 0 (zero) to turn off the feature. To turn on timed status the command would appear as:

```
@PJL USTATUS TIMED=<seconds><eol>
```

The returned information will be formatted as:

```
@PJL USTATUS TIMED <cr><lf>
CODE=<category><satus code>
{DISPLAY=<string><cr><lf>}opt
{ONLINE=<boolean><cr><lf>}opt
<ff>
```

Device Attendance Commands

An additional feature that may be desirable in a shared printing environment is device attendance messages. These messages, sent by the host, will cause a message to be displayed at the control panel and take the printer offline. The operator will then respond to the message and put the printer back online to resume printing.

Simple attendance

This request will cause the printer to display the message and go offline. The operator can respond to the message and put the printer back online.

```
@PJL OPMSG DISPLAY= <message><eol>
```

where <message> is a <string>. Each implementation needs to handle the case where the <string> does not contain any visible characters or the <string> is longer than the printer's display (16 characters for LaserJet 4Si).

Status Attendance

If status readback has been added, it may be desirable to distinguish how the operator put the printer back online. For example, the operator may have pressed the online key (attendance complete) or the continue key (override attendance).

```
@PJL STMSG DISPLAY = <message> <eol>
```

where <message> is a <string>. Returned data will have the following syntax:

```
@PJL STMSG DISPLAY=<message> <cr><lf>
<key><cr><lf>
<ff>
```

where <key>::={ ONLINE | CONTINUE | RESET }[1]

No Attendance

This request will not cause the printer to go offline, it simply replaces the ready message with the indicated message.

```
@PJL RDYMSG DISPLAY=<message><eol>
```

where <message> is a <string>. A null ready message (the message contains no characters) will cause the printer to display the printer's internal ready message.

Sample PJL Job Stream

The following is a sample job stream using the previously described PJL.

```
@PJL JOB NAME="EXAMPLE"
@PJL INITIALIZE
@PJL DEFAULT COPIES=1
@PJL SET PAGEPROTECT=LETTER
@PJL SET LPARM:PCL SYMSET=USASCII
@PJL ENTER LANGUAGE=PCL
PCL data stream
<PCL EOJ>
<ESC>%-12345X@PJL RESET (sets PROTECT back to OFF)
@PJL ENTER LANGUAGE=POSTSCRIPT
PostScript data stream
<PostScript EOJ>
<ESC>%-12345X@PJL RESET
@PJL EOJ NAME="END OF EXAMPLE"
```

Resource Saving

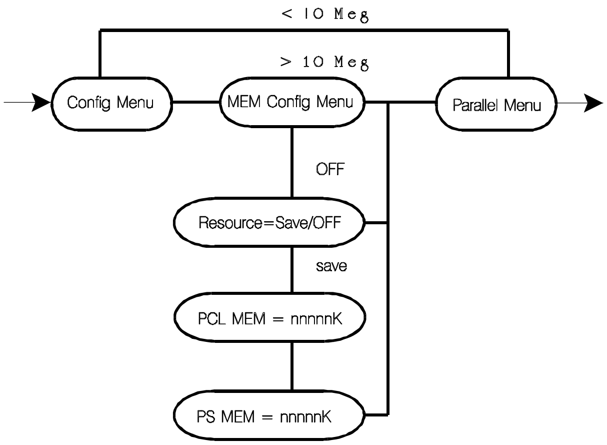
LaserJet 4Si supports resource saving of each personalities' permanent resources by setting aside printer memory dedicated to holding personality specific resources. The amount of memory set aside can be different for each installed personality. Some personalities may have memory set aside for resource saving without requiring all personalities to do so. Anytime the amount of memory dedicated to a specific personality is changed all personalities will lose any and all resources that had previously been saved including unprocessed print jobs.

Memory Requirements for Resource Saving

In the printer there are many features, besides resource saving, that require some minimum amount of memory to work properly. Page protection requires a certain amount of memory to be set aside based on the page size selected. The larger the page size the more memory is needed. 600 DPI printing requires more memory resources than 300 DPI. Duplex printing in PostScript requires more memory than simplex printing. It is currently felt that all of these features resource saving is the lowest priority. Therefore in the LaserJet 4Si resource saving will not be available until the memory requirements of all the other features are satisfied. The system will reserve enough memory to allow 600 DPI duplex (if installed) printing on legal paper and any remaining memory will be available for resource saving. It is currently felt that this will require at least 10 MB if the duplex unit is not installed and 16 MB if the duplexer is installed. This will leave about 2MB for resource saving.

Resource savings and The Control Panel

Resource saving is in the Memory Configuration Menu. The Memory Configuration menu is after the **CONFIG MENU** and before the **PARALLEL MENU**. The Memory Configuration menu will only appear if there are at least two personalities and at least 10 MB of memory (16 MB if the optional duplex unit is installed) in the printer. The Memory Configuration menu will be identified by the title **"MEMORY CONFIG MENU"**.



The first **MEMORY CONFIG MENU** item that will appear on the display is the item used to enable/disable resource saving. This menu item will be **"RESOURCES=SAVE/OFF"**. If this item is set to **"RESOURCES=OFF*"** no other **MEMORY CONFIG MENU** items will appear when the *ITEM Key* is pressed. If resource saving is enabled by setting **"RESOURCES=ON*",** then additional **MEMORY CONFIG MENU** items will appear that allow the user to configure the printer memory by setting aside memory for each installed personality to use as resource save space. These menu items will be in the form of **"XXXX MEM=10000K"** where **XXXX** is four spaces reserved for the personality name. The most common names being PCL and PS (PostScript). The numbers 10000K indicate the number of Kilobytes of memory that is to be reserved for the personality. This number must be an even multiple of 100Kbytes and will be in the range of 0 to 99900K (**NOTE:** 99900K is well beyond the maximum memory that can be installed in the printer). Each time the user presses the *+/- keys* the **"MEM=XXXX"** will increment/decrement by 100K with the following exceptions.

The first exception is caused by the fact that there may be an absolute minimum memory that must be available to support resource saving for some personalities. For instance it is possible that PostScript will require that all of their VM must be saved requiring a minimum of 700K plus space for fonts. In this case the **"PS MEM="** menu item may change between **"PS MEM=0K"** and **"PS MEM=800K"** for instance when the *+/- key* is pressed. The second exception has to do with attempting to exceed the maximum memory size available to this specific personality for resource saving (See NOTE #1 below). If the maximum memory size available to PCL for resource saving is 10000K and the menu display currently reads **"PCL MEM=10000K"** then pressing the *+ key* will cause the display to change to **"PCL MEM=0K"**. Conversely pressing the *- key* from **"PCL MEM=0K"** will cause the display to change to **"PCL MEM=10000K"**

NOTE #1: The maximum memory size available to this specific personality for resource saving is affected by the total memory installed in the printer and the memory currently reserved by other personalities for resource saving.

Resource Saving Related Error Messages

There are two actions the user can take that may cause error conditions in resource saving. The first is changing the configuration of the printer by adding or removing personality or DRAM SIMMs. Changing the printer configuration in this manner could cause the current MEMORY CONFIGURATION settings to become invalid or unachievable. The following sub-sections explain how LaserJet 4Si's resource saving will react to changes in the printer's configuration after the configuration changes are made and the printer is powered on. The view taken in defining LaserJet 4Si's reaction to configuration changes is to "View current configuration as permanent". The implication of this view is that if the configuration changes (i.e. personality or memory is removed), LaserJet 4Si is turned ON and then OFF, and the configuration is then changed backed to the original configuration (i.e. the removed resource is re-installed) when LaserJet 4Si is then turned back ON, the printer's resource saving memory settings may be different that the resource saving memory settings before the resource was removed. Two new clearable warning messages have been added to indicate changes in configuration. The first clearable warning message is:

WM CHK MEM CNFIG

This message indicates that there is still sufficient memory for resource saving but the current setting cannot be satisfied or that some personalities save area has been changed by the printer. The second clearable warning is:

WM MEM CNFIG N/A

This warning indicates that because of configuration changes that resource saving is now not available.

Enabling/Disabling Resource Saving

The paragraphs below describes how the printer functions and which messages are displayed when **RESOURCES** is turned **SAVE** or **OFF**.

Menu Change: Resource Saving Enabled
Each time the Resource Saving is enabled, LaserJet 4Si will verify there is enough installed memory to satisfy every personality's save area requirements. If there is insufficient installed memory to satisfy every personality's save area requirements, then the printer will honor one or more of the personality save area settings until the available memory is no longer sufficient to satisfy any remaining personality's requirement. All remaining personalities' save area size settings will be set to zero. There is no priority or defined algorithm for deciding which personalities get assigned memory and which personalities have their save area sizes set to zero. Any order LaserJet 4Si uses to set personalities' save area sizes to zero is acceptable. If the printer sets any personality's save area size to zero the **"CHK MEM CNFIG"** clearable warning message will be displayed.

Menu Change: Resource Saving Disabled
When the user selects **"RESOURCES=OFF*"** no other MEMORY CONFIGURATION manager related action's need to be performed except re-configuring LaserJet 4Si's memory.

Adding/Removing Options

The paragraphs below describes how the printer functions and which messages are displayed when options are added or removed.

<u>Power On: First Time</u>
When LaserJet 4Si detects that NVRAM has never been initialized it will set **"RESOURCES=OFF*"** and set each installed personalities' save area size to the minimum save area size required by the personality regardless of the amount of installed memory.

<u>Power On: Additional Memory Installed</u>
Installing additional memory above 10MB does not cause LaserJet 4Si to modify any settings associated with MEMORY CONFIGURATION or to display any MEMORY CONFIGURATION related error messages.

<u>Power On: Memory Removed</u>
Removing memory does not cause LaserJet 4Si to modify any MEMORY CONFIGURATION setting or display any error messages if the MEMORY CONFIGURATION is disabled. If the MEMORY CONFIGURATION is enabled and there is now less than 10 Mbytes of memory installed, then LaserJet 4Si will set **"RESOURCES=OFF*"** and display the **"MEM CNFIG N/A"** clearable warning message to indicate that the **MEMORY CONFIGURATION** is no longer available.

During power on, if more than 10 Mbytes of memory is installed and resource saving is enabled, LaserJet 4Si will verify there is enough installed memory to satisfy every personalities' save area setting. If there is insufficient installed memory to satisfy every personality's save area requirements, then the printer will honor one or more of the personality save area settings until the available memory is no longer sufficient to satisfy any remaining personality's requirement. All remaining personalities' save area sizes settings will be set to zero. There is no priority or defined algorithm for deciding which personalities get assigned memory and which personalities have their save area sizes set to zero. Any order LaserJet 4Si uses to set personalities' save area sizes to zero is acceptable. If the printer sets any personality's save area size to zero the **"CHK MEM CNFIG"** clearable warning message will be displayed.

Power On: Additional Personalities Installed

If LaserJet 4Si detects an additional personality has been installed, then it will add a menu item for that personality and set its memory size as follows. If the resource saving is disabled, or if the resource saving is enabled and there is enough memory to satisfy the new personality's minimum save area size requirement, the printer will set the new personality's save area size to the minimum size required. If there is not sufficient memory to satisfy the new personality's minimum save area size requirement, then the printer will set the new personality's save area size to zero. If the printer sets the personality's save area size to zero the **"CHK MEM CNFIG"** clearable warning message will be displayed.

Power On: Personality Removed

If LaserJet 4Si detects that a personality has been removed, the printer will remove the associated menu item. If the resource saving is enabled and the removed personality's save area size was greater than zero, then LaserJet 4Si will display the **"CHECK MEM MGT"** clearable warning message.

Power On: Duplex Option Added

If LaserJet 4Si detects that the duplex option has been added to a printer that also has PostScript, it will act the same as if memory has been removed. This is because adding duplex requires more memory for PostScript printing, leaving less memory available for resource saving, thus making it functionally equivalent to removing memory.

Power On: Duplex Option Removed

If LaserJet 4Si detects that the duplex option has been removed it will act the same as if memory has been added. This is because removing duplex requires less memory for PostScript printing, leaving more memory available for resource saving, thus making it functionally equivalent to adding memory.

Memory Overflow

When a memory overflow occurs the user needs to know whether the runtime memory pool was out of memory or whether the personality resource save area was out of memory. Also, the user would be interested in which personality was running when the error occurred. The following Error Messages are used:

"24 JOB MEM FULL" - Run time memory pool full.

"25 PCL MEM FULL", **"25 PS MEM FULL"**, or **"25 XXXX MEM FULL"** - Personality resource save area full (XXXX will be the personality name).

Resource Saving and the Self Test Configuration Page

The Self Test Configuration page will also contain resource saving information. One place containing this information is the **MENU** section of the configuration page. If the printer contains at least two personalities and at least 10 MB of memory then the **"RESOURCES=SAVE/OFF"** item will appear as a menu item under the heading **"MEM CONFIG MENU"** between the **CONFIG MENU** and the **PARALLEL MENU**. If **"RESOURCES=SAVE"** then the current settings for each personality will also appear right below it . Another area that shows resource saving information is the printer configuration section. This is the section that has the formatter number, RAM size, and so on. This section will contain lines that indicates whether or not there is sufficient memory for resource saving and if there is sufficient memory it will show the total memory available for resource saving and one additional line for each installed personality that indicates how much memory is set aside for that personality and what percentage of that memory is currently being used by the personality. Examples of these lines are shown below:

```
Resource Saving : 1300K Allocated of 2600K Available
  PCL Memory : Allocated 200K, 120K Used
  PS Memory : Allocated 1100K, 330K Used
```

The 1300K allocated at the end of the first line indicates that of the 2600k available for resource saving that only 1300K has actually been assigned to personalities.

If there is not sufficient memory for resource saving the lines will read:

Insufficient Memory Available for **EXT MEM MGT**
Additional Memory Needed for **RESOURCE SAVING: 5K**

NOTE : If only one personality is installed nothing related to MEMORY CONFIGURATION will appear on the self test configuration page.

Resource Saving and PJL

PJL data stream commands can be used to view the current configuration and utilization if **"RESOURCES=SAVE"**. The PJL INFO CONFIG command contains the configuration and utilization information found on the Self Test Configuration page in the following format:

RESOURCE SAVING [4 TABLE]

Memory	Amount	Used
System	2600k	1300k
PCL	200k	120k
Postscript	1100k	330k

PostScript Level 2

The LaserJet 4Si printer will support Adobe PostScript Level 2 via SIMM upgrade. Hewlett-Packard encourages developers to support the following in their PostScript printer drivers.

- ♣ HP LaserJet "4Si" Printer in your products menu.
- ♣ Language switching using the @PJL ENTER LANGUAGE command (refer to Chapter 3).
- ♣ Printer specific features implemented in PostScript Level 2 (media, envelope and tray selections).
- ♣ Resolution selection using @PJL SET RESOLUTION command (refer to Chapter 3).
- ♣ Tagged Binary Communications Protocol (TBCP) as a user selectable option. TBCP will improve printing performance.
- ♣ Compression filters, binary token encoding and binary object sequence encoding.

For description information on the implementation of specific details of the external interfaces to the Level 2 PostScript interpreter for Hewlett-Packard printers, refer to the "LaserJet 4Si PostScript Developers Guide". For information on TBCP and improved support for device-specific features, refer to later sections in this document. For information on all other features, refer to the PostScript Language Reference Manual (Second Edition) and Adobe's PostScript Language Software Development Kit. A good place to start in the kit is Supporting Level 2 Functionality, Technical Note #5110.

HP recommends that for optimum performance when sending binary data to a HP PostScript Level 2 printer that developers use TBCP, as discussed later in this document. Compression filters may be used with TBCP. The use of compression filters may enhance or degrade performance overall. The effect of compression filters is dependent upon such factors as what I/O is used to communicate with the printer, what the model of the printer is, or even which processor board is present in any particular printer. When sending binary data where portability between printers is an issue, such as sending binary data to a file which may be used on non HP printers, we recommend that ASCII85 encoding be used for PostScript Level 2 printers and Hexadecimal Encoding be used for PostScript Level 1 printers.

Compatibility with PostScript Level 1

In general, PostScript Level 2 is compatible with Level 1 because Level 2 emulates Level 1 operators. However, there are some differences. The following sections cover differences between the current PostScript Level 1 and Level 2 implementations that developers should be aware of.

Monochrome or Color Device

Only Level 1 devices with color extensions have color operators like *setcmykcolor* and setcolorscreen. All Level 2 devices, on the other hand, even monochrome devices, have color operators. There are some applications which determine whether or not a device supports color by checking for the existence of a color operator (i.e., *setcmykcolor* known). When such an application prints to a monochrome Level 2 device, halftone images tend to look blotchy or dark.

This is because the application sets up halftone screens with the *setcolorscreen* operator using four colors, while the printer ignores all but the black screen. The correct way to determine whether a device is monochrome or color is as follows:

```
statusdict /processcolors known
{statusdict /processcolors get exec} {1}
ifelse
```

This returns the number of process colors (1, 3 or 4) in the device. The application can then set up halftone screens accordingly.

Another way to determine whether a device is monochrome or color is to use the *DefaultColorModel and *ColorDevice keys in the device's PostScript Printer Description (PPD) file (See the section Supporting HP LaserJet Device-Specific Features for more information on PPD files).

Duplex

For duplex documents, calling a device setup Level 1 operator between pages will cause the previous page to be simplexed if it was printed on the first side. Device setup Level 1 operators include all Level 1 operators equivalent to the setpagedevice parameters documented in Adobe's PostScript Language Addendum for the Hewlett-Packard LaserJet III Family of Printers.

The LaserJet 4Si printer supports duplex printing.

Statusdict Compatibility Operators

In Level 1, all statusdict operators that set defaults generate an invalidaccess error if they are executed in an encapsulated job. In Level 2, no error is generated and the operators execute as if they are in an unencapsulated job.

Page Type Operators

letter, legal, executivepage, a4, b5, com10envelope, monarchenvelope, -c5envelope, dlenvelope - In Level 2, all page type operators have a side effect in that they initiate a tray search for a tray with the correct size paper to match the requested image size. This can result in paper being pulled from a different tray than the default or requested tray. If the desired tray size is not found, the next larger available medium is chosen but no adjustment is done to the image size to make it fit. For example, it is now more difficult to create a letter size image on legal sized paper. Also, all page type operators reside in userdict rather than statusdict.

setdefaultduplexmode, setdefaultpapertray, setdefaulttimeouts, -setdefaulttrayswitch, setdefaulttumble and setdostartpage - In Level 1 these operators did not take effect until the end of the job when executed in an unencapsulated job. In Level 2 they take effect immediately.

setdefaultduplexmode, setduplexmode - In Level 1, attempting to enable duplex on a printer that does not support duplex results in an error. In Level 2 no error is generated and the job is simplexed. Also, tumble operators are ignored.

setdefaulttimeouts, setjobtimeout - Larger Persistent values can be set using these operators in Level 2 than in Level 1. manualfeedtimeout has increased from an 8-bit value to a 16-bit value, jobtimeout has increased from an 8-bit value to a 12-bit value, and waittimeout has increased from an 8-bit value to a 12-bit value.

setdoidlefonts, doidlefonts, idlefonts - Level 2 has its own mechanism for font caching, so these operators do not affect anything. setdoidlefonts will just pop the argument off the stack, doidlefonts will always return false, and idlefonts will just place a mark on the stack.

setduplexmode, setpapertray, settrayswitch, settumble - In Level 1, if these operators are executed in an unencapsulated job, they take effect for the current job only. In Level 2 they change the default.

setprintername/printername - In Level 1, if the Serial, Parallel, XIO or MIO communications channels are active when setprintername is executed in an unencapsulated job, the string must be "Hewlett-Packard LaserJet #x" where "x" represents any printable character except ":" or "@", and the default is "Hewlett-Packard LaserJet #0". In Level 2, the string must be "HP LaserJet" plus up to 6 additional characters, and the default is "HP LaserJet."

setsccbatch/sccbatch: In Level 1 setsccbatch allows the developer to set the baud rate and appear to set the parity value for a particular channel, and the sccbatch operator returns the new baud rate and the parity value that the user had requested. In the printer hardware, the new baud rate actually takes effect but the parity remains unchanged. In Level 2, setsccbatch accepts values as in Level 1, but does not affect the hardware settings in any way, and sccbatch returns the actual hardware settings, rather than the values input by the user.

cexec - cexec is an undocumented operator in Level 1. In Level 2 it is not supported and executing it will result in an undefined error (PS ERROR 18).

checkpassword - In Level 1 checkpassword returns true if the password argument matches the Level 1 system password. In Level 2 checkpassword returns true if the password argument matches either of the Level 2 passwords.

cshow - In Level 1, cshow is an undocumented operator that takes a single string argument, and centers it about the current point with respect to the y-axis. In Level 2, cshow is a documented operator that takes both a procedure and a string argument, and has to do with character mapping, primarily for composite fonts. Executing cshow with only a string argument in Level 2 results in a stackunderflow error (PS ERROR 14).

gsave/grestore - In Level 1, device parameters like tray selection and duplexing that get set with operators, such as setpapertray and setduplexmode are not included in the graphics state that gets saved and restored. In Level 2, all of these device parameters are stored in the page device dictionary, even if they are set via Level 1 operators, and the page device dictionary is included in the graphics state. Refer to Adobe's *PostScript Language Addendum for the Hewlett-Packard LaserJet III Family of Printers* for a complete list of page device parameters.

manualfeed - This a Level 1 operator that is initially set to false. If this Boolean is set to true, manual feed is enabled, regardless of the value of the ManualFeed entry in the page device.

manualfeedtimeout - This is the only Level 1 operator in statusdict that is not initialized to some default value, so it should never be read without being set first. If defined, this value overrides the value of ManualFeedTimeout specified in the current page device dictionary. This is the number of seconds the printer will wait for a page to be manually feed 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.

papersize - In Level 1 and Level 2 this is an undocumented operator that varies from printer to printer. It's operation may not be as compatible with DEC environments in Level 2 as it is in Level 1.

papertype - In Level 1 this is an undocumented operator. In Level 2 it is no longer present and executing it results in an undefined error (PS ERROR 18).

product - product is now found in systemdict as well as statusdict.
setpassword - This operator is not supported in Level 2. There was no reasonable way to map this into the dual password architecture of Level 2. Executing it results in an undefined error (PS ERROR 18).

solicited status - The list of solicited (ctrl T) status returned in Level 2 does not include printing, as it did in Level 1.

Device Specific Features

When customers purchase an HP LaserJet 4Si printer, they expect to be able to access its features from their applications. The following is a checklist of basic printer features for which users expect support:

HP LaserJet 4Si PostScript Level 2

____ Paper Sources
    ____ Auto Select *
    ____ Upper Tray
    ____ Lower Tray
    ____ Envelope Feeder
    ____ Manual Feed

____ Paper Sizes
    ____ Letter (8 1/2 x 11 in)
    ____ Legal (8 1/2 x 14 in)
    ____ Executive (7 1/4 x 10 1/2 in)
    ____ A4 (210 x 297 mm)
    ____ JIS B5 (182 x 257 mm) **
    ____ Envelope #10 (4 1/8 x 9 1/2 in)
    ____ Envelope Monarch (3 7/8 x 7 1/2 in)
    ____ Envelope DL (110 x 220 mm)

____ Orientation
    ____ Portrait
    ____ Landscape
    ____ Rotated Landscape

____ Resolution Enhancement
    ____ ON
    ____ OFF

____ Output Bin

    ____ Upper Bin
    ____ Lower Bin (rear)

____ Duplex printing

    ____ ON with Long edge binding
    ____ ON with Short edge binding
    ____ OFF

* Auto Select refers to the paper source being selected automatically for the user, based on the paper size requested.
** There is no B5 paper tray for these printers, so B5 paper is primarily for manual feed. When manually feeding B5 paper, the A4 settings on the paper guide should be used.

Another feature that users very often request is the ability to print a job using paper from more than one source. The most basic approach is a letterhead approach, where the first page comes from one source, and the rest of the job from another. This capability is provided in the MS Windows and Macintosh drivers that HP is providing for the PostScript Level 2 SIMM. The best approach, however, would be to allow the user to select a paper source on a page-by-page basis.

Improved Support for Device-Specific Features in Level 2

In Level 1, device control operators are defined in a special dictionary named statusdict. The contents of statusdict are product dependent, although an attempt was made to maintain some consistency. In Level 2, there is a new page device setup operator called setpagedevice. This operator provides a standard framework for specifying the requirements of a page description and for controlling both standard and optional features of a device. It uses a standard syntax to request features supported by all devices, such as selecting a page size, and features supported only by some devices, such as duplex printing. Setpagedevice also provides a standard mechanism for determining what to do when a page description makes requests the device can not fill. Setpagedevice makes it easier for developers to support multiple devices with one driver, and makes print jobs with device-specific feature support more portable.

For an explanation of how setpagedevice works, see the PostScript Language Reference Manual (Second Edition), especially Section 4.11. For a list of setpagedevice parameters (like PageSize and Duplex) supported by PostScript Level 2, see Adobe's PostScript Language Addendum for the Hewlett-Packard LaserJet III Family of Printers. For an explanation of a particular parameter, the addendum may refer you to the PostScript Language Reference Manual (Second Edition), or to Adobe's PostScript Language Supplement for Version 2011, which documents parameters that are new since the reference manual was published.

HP and Adobe recommend that developers use the new Level 2 operators, instead of the Level 1 compatibility operators, for supporting device-specific features. HP and Adobe also recommend that developers not implement a combination of Level 1 and Level 2 operators. Note that because Level 2 is generally compatible with Level 1, device feature support implemented with Level 1 operators will still work with the PostScript Level 1, except for a few exceptions.

See the sections *Compatibility with PostScript Level 1* in this document for information on the exceptions.

Strategies for Supporting Device-Specific Features

Adobe has created PostScript Printer Description (PPD) files to facilitate the support of device-specific features. PPD files are text files containing information about printers in a consistent format that can be read by software. The files contain many different kinds of device-specific information that may be used by printer driver software for decision-making purposes or to invoke particular features of the printer. There is one PPD file per device, so there are PPDs for each LaserJet printer . PPD files reside on the host computer and/or print spooler.

PPD files contain information about the printer like available fonts, the PostScript Level, and the PostScript version of the interpreter. They also contain information about device features. For each device feature, some of the information which the PPD file provides is the name of the feature (i.e., PageSize), all possible options for the feature (i.e., Letter, Executive, etc.), and the PostScript code necessary to invoke the feature with the selected option. The application/driver is to use the information in the PPD file to build a user interface with all available features and options. Then the application/driver is to invoke the features and options selected by the user by extracting the appropriate PostScript code from the PPD file and placing it in the output file.

When it comes to placing the PostScript code in the output file, Adobe recommends surrounding the code with comments that conform to Adobe's Document Structuring Conventions (DSC). The DSC comments identify the code in an output file which invokes device-specific features. This allows the output file to be redirected from one device to another by a spooler or other post-processing software. As an output file is routed across a network, a spooler can extract device-specific code by parsing for the associated DSC comments. The spooler can then parse the PPD file for the new device, extract new device-specific code, and insert the new device-specific code before sending the output file to the new device.

The following is an excerpt of a PostScript output file that follows the DSC. Note that the lines between %%BeginFeature and %%EndFeature requesting duplexing were extracted from a PPD file:

```
%!PS-Adobe-3.0
%%Title: test.ps
%%LanguageLevel: 2
%%EndComments
%%EndProlog
%%BeginSetup
%%BeginFeature: *Duplex DuplexNoTumble
1 dict dup /Duplex true put setpagedevice
1 dict dup /Tumble false put setpagedevice
%%EndFeature
...
%%EndSetup
%%Page: one 1
```

```
...
%%Trailer
```

An additional technique can be used to make PostScript files with device-specific features even more portable. An example of this technique is as follows:

```
mark {
1 dict dup /Duplex true put setpagedevice
1 dict dup /Tumble false put setpagedevice
} stopped cleartomark
```

If this code were sent to a device that did not support setpagedevice, that is, a Level 1 printer, the code between the mark { and } stopped cleartomark would simply be ignored. Without the mark { and } stopped cleartomark, the code would cause an error and the job would terminate.

Detailed information on PPD files and the DSC can be found in PostScript Language Reference Manual (Second Edition):

- Document Structuring Conventions - Version 3.0, Appendix G

Adobe's PostScript Language Software Development Kit:

- Supporting Device Features, Technical Note #5117
- Cooperative Printing Guidelines, Technical Note #5030
- PostScript Printer Description Files Specification, Versions 3.x and 4.x
- PPD File Parser: Application Programmer's Interface, Technical Note #5127

Two PPD files for each printer mode are provided in the LaserJet 4Si Developer Kit -- one complying with Version 3. 0 of the PPD Specification, and the other complying with Version 4.0b3. The Version 4.0b3 PPD is a preliminary version of the 4.X PPD which is still being developed. While the Version 4.0b3 PPD illustrates the direction that PPD's are following, the specifics of the PPD are not yet frozen. Therefore this PPD should not be used to develop code which is to be released.

Duplexing and setpagedevice

For duplex documents, calling setpagedevice (or a Level 1 device setup operator which is equivalent to a setpagedevice call) between pages will cause the previous page to be simplexed if it was printed on the first side.

Persistent Parameter Changes

Device parameters set with setpagedevice fall into the following categories:

- Volatile - New value is retained only for the duration of the job.
- Semi-persistent - New value is retained across job boundaries, but not power cycles, 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.

- ♣ Constant - Value never changes.
- ♣ Persistent/Read-only - Value changes by internal mechanisms only and is retained across job boundaries and power cycles.

HP discourages persistent parameter changes. For Level 2 this means that developers should change persistent parameters in an encapsulated job rather than in an unencapsulated (by means of the startjob operator) job. For Level 1 this means that developers should use an operator like setduplexmode in an encapsulated job instead of using an operator like setdefaultduplexmode in an unencapsulated (by means of the exitserver operator) job. There are a couple of reasons for avoiding persistent parameter changes. First, changing a persistent parameter requires a change to the printer's NVRAM, and NVRAM has a finite life. Second, it's hard to find a logical reason for doing so. One job should not change a setting that will affect subsequent jobs.

TBCP

The HP's implementation of PostScript Level 2 supports a new protocol for communication between the printer and a host computer over a serial or parallel channel. This new protocol is called Tagged Binary Communications Protocol (TBCP). TBCP creates a "transparent" communication channel, in that it allows any 8-bit binary value (0 - 255) to be treated as data. When TBCP is not used, the printer is in Adobe Standard Protocol (ASP) mode, which was designed for data in the printable ASCII range, and a few special control characters outside that range. TBCP was created to allow any 8-bit binary value to be treated as data, while allowing a few of the values to function as special control characters as well.

One use for which TBCP is clearly advantageous is raster image data. With ASP, 8-bit binary raster image data has to be converted into printable ASCII data. This is normally done with ASCII Hex encoding, which doubles the size of binary data. TBCP also adds a little overhead to binary data, but about 3% rather than 100%. The size of raster image data can be further reduced by using compression filters with TBCP. Compression filters are a new Level 2 feature. Two other new Level 2 features, binary token encoding and binary object sequence encoding, involve the substitution of 8-bit binary data for printable ASCII data. To reduce transmission and execution overhead, and as such require TBCP to send the binary data over a serial or parallel channel.

HP recommends that all developers support TBCP because it reduces data transmission time and storage requirements. However, it only brings these benefits if binary data is sent. HP also recommends that developers support binary token encoding and binary object sequences. For those developers whose applications print raster image data or download fonts, HP recommends sending raw binary data or binary data compressed with one of the new Level 2 compression filters. Because the decompression overhead associated with the compression filters can in many cases outweigh the benefits of the reduced size, HP recommends that developers do performance testing to determine the best solution for their applications.

NOTE ☞ TBCP is unnecessary over transparent channels (like LocalTalk) that support a packet protocol. Packet protocols are already able to handle binary data because data and control functions are completely separate.

TBCP vs. BCP

TBCP was created by Adobe and HP as an alternative to Adobe Binary Communications Protocol (BCP). The two protocols are similar but not compatible. HP's implementation of PostScript Level 2 supports TBCP, and not BCP, and errors will result if BCP is sent. Likewise, errors will result if TBCP is sent to a printer that only supports BCP.

A comparison between TBCP and BCP will highlight the differences in implementation and will be helpful in understanding the implementation of TBCP. First, both BCP and TBCP reside in the protocol layer which receives the data and is then passed on to the PostScript interpreter. In BCP, the binary data is enabled by the interpreter. In TBCP, the binary data is enabled in the I/O protocol layer, before the interpreter. In BCP, input data received immediately after the PostScript interpreter commands to enable or disable BCP may be parsed incorrectly at the I/O protocol layer. This is because of the latency that exists between the I/O protocol layer and PostScript interpreter layer. Therefore mixed ASP and BCP jobs will not always be handled correctly. Since TBCP controls the binary data protocol through the I/O protocol layer, it is always in sync with the incoming data. Thus PostScript jobs which mix the ASP and TBCP protocol will always be handled correctly. ASP jobs sent with BCP mode that rely on CR/LF folding (the conversion of a CR/LF into a LF) to take place, may not work correctly.

Implementing TBCP

To use TBCP, a PostScript job needs to be constructed as follows:

```
<CTRL - A>M
PostScript job with quoting
<ESC>%-12345X
```

where <CTRL-A> represents a byte with the hex value 0x01 and <ESC> represents a byte with the hex value 0x1B. The <ctrl-A>M enables TBCP. The <ESC>%-12345X disables TBCP and puts the printer back into ASP mode.
If you normally put a <CTRL-D> at the beginning of your jobs, it should come immediately after the <CTRL-A>M. As far as a <CTRL-D> at the end of a job, it can be put immediately before the <<esc>>%-12345X, but is optional. If it's not present, the <ESC>%-12345X performs the exact same function as a <CTRL-D> in addition to disabling TBCP.

The PostScript job that goes between the enabling and disabling sequences is likely to contain some 8-bit binary data values that have the same hex values as special control characters. The list of special control characters is as follows:

ASCII KeyBoard	ASCII Name	HEX Value	Type	Control Function
CTRL-A	SOH	01	Synchronous	Quote data character
CTRL-C	ETX	03	Asynchronous	Abort job and flush to end of file
CTRL-D	EOT	04	Synchronous	End of File marker
CTRL-E	ENQ	05	----	Reserved for future use

CTRL-Q	DC1	11	Asynchronous	XON in XON/XOFF flow control
CTRL-S	DC3	13	Asynchronous	XOFF in XON/XOFF flow control
CTRL-T	DC4	14	Asynchronous	Job status request
CTRL-[	ESC	1B	Synchronous	Start of end protocol sequence
CTRL-\	FS	1C	---	Reserved for future use

In order to differentiate data bytes from special control characters, any data byte that is the same as one of the above special control characters must be quoted. A data byte is quoted by replacing it with a two-byte sequence. The first byte is a <CRTL-A> (hex 0x01) and the second byte is the original byte XORed with the value hex 40. For example, to send the hex value 14 as data, you would send the two byte hex sequence 0x01 0x54 instead. This method of quoting guarantees that whenever any of the nine special control characters is received by the printer, the control function is intended, regardless of whether the preceding character is a <CTRL-A>. The generation and processing of asynchronous special control characters, may occur at a lower level than the generation and consumption of the data stream. In particular, on a host machine, the <CTRL->A quoting convention may be implemented by an application driver, while XON/XOFF processing is performed independently by a lower level communications layer.

Any data byte not equal to one of the nine special control characters is transmitted by simply sending the byte as is.

Below is a sample implementation of quoting in C:

```
#define UCHAR unsigned char

#define CTRLA                                                   (UCHAR)0x01
#define CTRLC                                                   (UCHAR)0x03
#define CTRLD                                                   (UCHAR)0x04
#define CTRLE                                                   (UCHAR)0x05
#define CTRLQ                                                   (UCHAR)0x11
#define CTRLS                                                   (UCHAR)0x13
#define CTRLT                                                   (UCHAR)0x14
#define CTRLBRKT                                                (UCHAR)0x1B
#define CTRLBKSL                                                (UCHAR)0x1C

/* QuoteByte quotes a character, if necessary, using the Adobe Tagged Binary Communications P
*/

void QuoteByte(c)
UCHAR c;
{
  switch (c)
  {
    case CTRLA:
    case CTRLC:
    case CTRLD:
    case CTRLE:
    case CTRLQ:
    case CTRLS:
    case CTRLT:
    case CTRLBKSL:
        outbyte(CTRLA);
        outbyte((UCHAR)(c ^ 0x40));
        break;
    default:
            outbyte(c);
    }
}
```

After a <CTRL-A> is received, the next character received must be an 'M', the result of XORing one of the special control characters with the value hex 40, or one of the asynchronous special control characters. Receipt of any other character is considered an error in the input. Between the <CTRL-A> and the XORed character, any number of asynchronous special control characters may appear. However, receipt of a synchronous special control character (<CTRL-D> or <CTRL-A> between a <CTRL-A> and the XORed character is considered an error. Note that the primary difference between synchronous and asynchronous special control characters is that synchronous special control characters are generated as part of the PostScript data stream and belong in particular locations within the PostScript data stream. Asynchronous special control characters, on the other hand, are typically generated by a lower level communications layer, and appear in random locations within the PostScript data stream.

If one of the special control characters arrives unquoted and it specifies no control function for the channel, the character is simply discarded. For example, if XON or XOFF is received over a parallel channel, it is discarded.

The characters <<CTRL-E>> and <<CTRL-\>> currently specify no control functions. They are included among the characters that must be quoted in case new control functions are added in the future.

For any data received by the I/O protocol layer, the following table shows what is received by the PostScript interpreter and what action results:

Received by the I/O Protocol Layer	**Received by the PostScript interpreter**	**Action**
<CTRL-C>	--	Abort Job
<CTRL_T>	--	Request Job Status
<CTRL-Q>	--	Flow Control
<CTRL-S>	--	Flow Control
<CTRL-D>	End of File Mark	--
<CTRL-E>	--	--
<CTRL-\>	--	--
<CTRL-[>%-12345X	End of File Mark	TBCP disabled
<CTRL-A>A	<CTRL-A>	--
<CTRL-A>C	<CTRL-C>	--
<CTRL-A>D	<CTRL-D>	--
<CTRL-A>E	<CTRL-E>	--
<CTRL-A>M	<CTRL-M>	--
<CTRL-A>Q	<CTRL-Q>	--
<CTRL-A>S	<CTRL-S>	--
<CTRL-A>T	<CTRL-T>	--
<CTRL-A>[	<CTRL-[>	--

<CTRL-A>\	<CTRL-\>	--
<CTRL-A><Any other char>	--	Error
<Any other char>	<char received>	--

Bidirectional Communication

TBCP is bidirectional. That is, if TBCP is being used over a bidirectional communication channel like serial, TBCP is used on any back channel communication going from the printer to the host.

If the printer receives a <CTRL-A>M from the host, the printer sends a <CTRL-A>M back to the host in front of the first data it sends to the host. If the printer never has any data to send back to the host, the <CTRL-A>M does not get sent. If the printer sends binary data over the back channel, it will quote any data values that are the same as one of the special control characters. At the end of the job, after the host has sent the <ESC>%-12345X to the printer, the printer will send <CTRL-D><ESC>%-12345X back to the host.

EPS Files and User Interface

Because TBCP is a device dependent feature, it should never be used in Encapsulated PostScript (EPS) files. In addition, HP and Adobe recommend allowing the user to enable or disable TBCP at the user interface level, because in using TBCP you gain performance but lose portability.

When presenting this option to the user, do not use the term TBCP. Instead allow the user to select binary communications for printing directly to the printer or Hexadecimal ASCII when printing to a file which may be sent to some other printer. The selection may be made in an area titled Performance Options.

PostScript Error Reporting

One class of errors that can be generated by the PostScript Level 2 are PostScript errors (PS ERROR xx). LaserJet 4Si reports PostScript errors differently from previous LaserJet printers using PostScript cartridges. First PostScript errors are no longer displayed on the printer's control panel. Also, *print errors* can be enabled from the printer's control panel.

AUTOCONTINUE=OFF will no longer display PS errors on the printer's control panel. This makes the LaserJet 4Si printer behave more like PostScript printers which do not typically have displays. If a PostScript job encounters a PostScript err, the job stream will be scanned for end of job, and the job will be aborted. No error will be displayed on the printer's control panel display. If PRT PS ERRS=ON or DoPrintErrors is True, then an error page will be printed.

The control panel selection of PRT PS ERRORS and system parameter DoPrintErrors controls whether or not the printer prints a page describing the PostScript error.

The contents of the page are as follows:

ERROR - This is the PostScript name for the error, i.e., undefined. Information on the error can be found in the PostScript Language Reference Manual.

OFFENDING COMMAND - This is the PostScript object that was being executed at the time of the error; e.g., moveto.

STACK - This is the contents of the PostScript operand stack at the time of the error.

The system parameter DoPrintErrors must be set to true or false by sending a PostScript job from the host to the printer. The factory default for DoPrintErrors is false.

A PostScript job that sets DoAutoContinue to false and DoPrintErrors to true would look like this:

```
true 0 startjob pop
</DoAutoContinue false
/DoPrintErrors true
>setsystemparams
false 0 startjob pop
```

PostScript Errors

The following is a list of PostScript errors that are returned by the LaserJet 4Si printer.

PostScript Error
dictfull
dictstackoverflow
dictstackoverflow
execstackoverflow
invalidaccess
invalidexit
invalidfileaccess
invalidfont
invalidrestore
ioerror
limitcheck
nocurrentpoint
rangecheck
stackoverflow
stackunderflow
syntaxerror
timeout
typecheck
undefined
undefinedfilename
undefinedresult
unmatchedmark
unregistered
VMerror*
quitexecuted*
Fatalerror*
configurationerror
interrupt
undefinedresource

Executive Mode

The PostScript Level 2 supports executive mode for interactive debugging.