Developer Note

Apple LaserWriter Pro 600 and 630 Printers

Developer Note APPLE CONFIDENTIAL

March 1, 1993
Developer Technical Publications

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014
408-996-1010

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

If you discover physical defects in the manual or in the media on which a software product is distributed, APDA will replace the media or manual at no charge to you provided you return the item to be replaced with proof of purchase to APDA.

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Figures and Tables vii

Preface About This Developer Note ix

What This Note Contains ix
Conventions Used in This Note x
Other Reference Material x
For More Information xi

Chapter 1 Overview of LaserWriter Pro Hardware 1

Features of the LaserWriter Pro Printer 2
Communication 3
  Serial Communications 4
    Serial Port for RS-232C Devices 5
    LocalTalk/RS422 Serial Connector 6
  Centronics Parallel Connector 6
  Apple Ethernet Adapter for EtherTalk (630 Only) 8
  SCSI Connector (630 Only) 8
Internal SCSI Connector 10
Mode Switch 12
Status Lights 16
DRAM Expansion 17

Chapter 2 Overview of LaserWriter Pro Software 19

Adobe PostScript Programming Language 20
LaserWriter Pro Driver 20
LaserWriter Pro Utility Program 21
Startup Page 21
Configuration Page 21
Page Types 24
LaserJet IIP Emulation 25
  Selecting Emulation 25
  Setting the Server to Emulation Mode 25
  Selecting From Within a PostScript Language Program 25
  Using a Serial Connection 26
  LaserJet IIP Emulation Differences 26
Mode Selection Parameters 26

Device Setup 30
Page Device Parameters 30
Page Device Parameter Summary 31
Product Strings 34
Install Procedure 34
PhotoGrade Parameters 34
FinePrint Parameters 36
Interpreter Parameters 37
User Parameters 37
System Parameters 38
Device Parameters 41
File System 41
Communication Channels 44
Engine Device 48
Resource Categories 48
Emulator Parameters 54
Compatibility Operators 55
Setting System Parameters 57
buildtime 57
byteorder 57
checkpassword 58
defaultmultipurposetraysize 58
defaultpapertray 58
defaulttimeouts 58
dostartpage 58
dosysstart 59
emulate 59
pagecount 59
papersize 59
papertray 60
printername 60
product 60
ramsize 60
realformat 60
revision 61
setdefaultmultipurposepapertraysize 61
setdefaulttimeouts 61
setdostartpage 61
setdosysstart 62
setpapertray 62
setprintername 62
Setting Page Device Parameters 63
margins 63
pagestackorder 63
setmargins 63
setpagestackorder 63

Setting User Parameters 64
jobname 64
jobtimeout 64
setjobtimeout 64
waittimeout 64
Setting Device Parameters 65
appletalktype 65
diskonline 65
diskstatus 65
initializedisk 65
hardwareiomode 66
manualfeed 66
manualfeedtimeout 66
sethardwareiomode 67
setsoftwareiomode 67
setuserdiskpercent 67
softwareiomode 67
userdiskpercent 68
Setting Serial Communication Parameters 68
sccbatch 68
sccinteractive 68
setsccbatch 69
setsccinteractive 69
SCC Operator Encoding 69
Page Size Compatibility Operators 71
Paper Tray Compatibility Operators 72

Chapter 4 TrueType Fonts 73

TrueType Font Format 74
TrueType Code 75
Patch 75
TrueType Font Definition 76
Device Operation 77
Class A Devices 77
Class B Devices 78
Class C Devices 78
Downloading TrueType Fonts to Disk 78
TrueType Font Dictionary Entries 80

Figures and Tables

Chapter 1 Overview of LaserWriter Pro Hardware 1

Figure 1-1 LaserWriter Pro side panel connectors 3
Figure 1-2 9-pin serial port connector for RS-232C device 5
Figure 1-3 8-pin mini-DIN connector for LocalTalk and RS-242 devices 6
Figure 1-4 Centronics parallel connector 6
Figure 1-5 14-pin Ethernet connector 8
Figure 1-6 30-pin SCSI connector pin designations 9
Figure 1-7 Internal SCSI connector 10
Figure 1-8 Status lights 16
Figure 1-9 Expansion slots for SIMM cards 17

Table 1-1 LaserWriter Pro features 2
Table 1-2 Signal descriptions for 9-pin serial port 5
Table 1-3 Signal descriptions for LocalTalk and RS422 serial port 6
Table 1-4 Signal descriptions for Centronics parallel port 7
Table 1-5 Signal descriptions for Ethernet connector 8
Table 1-6 Signal descriptions for 30-pin SCSI connector 9
Table 1-7 Signal descriptions for internal SCSI connector 11
Table 1-8 Switch settings and default parameter values 12
Table 1-9 Status light messages 16
Table 1-10 DRAM expansion configurations 17

Chapter 2 Overview of LaserWriter Pro Software 19

Figure 2-1 Layout of configuration page 22

Table 2-1 Configuration page functions 23
Table 2-2 Paper tray selection operators 24

Chapter 3 LaserWriter Pro Software 29

Table 3-1 Page device parameters 31
Table 3-2 Paper sizes and memory usage 33
Table 3-3 Paper tray slot numbers and input sources 33
Table 3-4 Product string values 34
Table 3-5 PhotoGrade device parameters 35
Table 3-6 Frequently used entries in the `PreRenderingEnhanceDetails` dictionary 35
Table 3-7 PhotoGrade calibration parameters 36
Table 3-8 FinePrint device parameters 36
Table 3-9 User parameters 37
Table 3-10 System parameters 38
Table 3-11 Parameters common to all FileSystem devices 42

Table 3-12 Parameter set relationships 45
Table 3-13 %Serial_NVx% communication parameter sets 46
Table 3-14 %SerialB_NVx% communication parameter sets 46
Table 3-15 %LocalTalk_NVx% communication parameter sets 47
Table 3-16 %EthernetTalk_NVx% communication parameter sets 47
Table 3-17 %Parallel_NVx% communication parameter sets 47
Table 3-18 %Engine% communication parameter sets 48
Table 3-19 Regular resource categories 49
Table 3-20 Resources with implicit instances 51
Table 3-21 Resources used in defining new resource categories 53
Table 3-22 Emulator parameters 54
Table 3-23 Compatibility operators 56
Table 3-24 SCC compatibility operators options byte values 70
Table 3-25 Optionsbyte-to-device parameters 70
Table 3-26 Device parameters-to-options conversion 71
Table 3-27 Paper size compatibility operators 72
Table 3-28 Paper tray compatibility operators 72

Chapter 4 TrueType Fonts 73

Table 4-1 Section code comments 80
Table 4-2 Type 42 key-value pairs common to all PostScript font dictionaries 81
Table 4-3 Entries for Type 1 specific font dictionaries 82
Table 4-4 Font dictionary entries specific to Type 42 fonts 83
Table 4-5 Optional entries for FontInfo dictionary 83

About This Developer Note

The LaserWriter Pro 600 and the LaserWriter Pro 630 are new members of Apple's LaserWriter printer family. This developer note describes the features and capabilities of the printers, and it is intended for use by software and hardware developers.

To use this note, you need to understand the Adobe™ PostScript™ Level 2 programming language. You should also be familiar with the computer for which you intend to develop software.

You do not need to use this note if you are simply running packaged programs for your Apple computer. However, the note is useful if you are writing or modifying a program that is used with the LaserWriter Pro printer.

Your owner's guide provides instructions for connecting the printer to your computer, inserting paper, and performing other routine operating tasks. This note does not provide that type of information.

This preface describes the contents of the note, gives visual cues and conventions used in the note, and lists other books to which you can refer.

What This Note Contains

This note is made up of four chapters.

- Chapter 1, "Overview of LaserWriter Pro Hardware," describes the hardware features of the LaserWriter Pro printer, including the built-in communications ports and interfaces. It also shows how to select printer modes using the rotary mode-selection switch.
- Chapter 2, "Overview of LaserWriter Pro Software," provides general information about the Adobe PCL (Printer Control Language), LaserWriter Pro driver, the utility program, starting and configuration pages, page types, LaserJet IIP emulation, and modifying mode selection parameters.
- Chapter 3, "LaserWriter Pro Software," describes the software parameters, resource categories, and PostScript Level 2 operators that enable you to set up and configure the LaserWriter Pro printer.
- Chapter 4, "TrueType Fonts," describes the TrueType downloadable PostScript font format for the LaserWriter Pro printer.

Conventions Used in This Note

Note
This visual cue is used throughout the book to draw attention to information of general interest. ◆

IMPORTANT
This type of note contains information that is essential for an understanding of the main text. ▲

A special font, `Courier`, is used for characters that you type, or for lines of program code. `It looks like this`.

Other Reference Material

This developer note assumes that you are familiar with printer technology, and know how to operate and program Apple LaserWriter printers. It supplements the following developer notes:

- *LaserWriter IIf and LaserWriter IIg Printers*, R0230LL/A, 030-2460-A.
- *Personal LaserWriter NTR Printer*, R0258LL/A, 030-2664-A

Additional information is available in the following publications:

- The owner's guide that is shipped with every LaserWriter printer explains how to set up the printer in the standard configuration. The guide gives basic operating information on how to load toner cartridges, load the paper tray, set up the configuration switch for your communications environment, set up an external hard disk for fonts. It also provides basic troubleshooting information.
- The *LaserWriter Reference*, published by Addison-Wesley, describes the capabilities of the LaserWriter Plus, the LaserWriter IINT, and the LaserWriter IINTX printers. It also includes information that is not in this note about fonts and about communicating with LaserWriter printers over the serial channels.
- The *PostScript Language Reference Manual*, second edition, published by Addison-Wesley, is required if you plan to write programs in the PostScript Level 2 programming language. The edition referenced throughout this note is the Second Edition. There is also a supplement to this manual, referred to as the *PostScript Language Level 2 Supplement*, or simply the *Supplement.* This book is available from Adobe Systems, Inc.
- The *PostScript Language Tutorial and Cookbook*, published by Addison-Wesley, provides a basic introduction to the PostScript programming language. It also includes sample PostScript programs that help you quickly understand how the PostScript programming language works.

- The *PostScript Language Program Design*, published by Addison-Wesley, is written for programmers who want to take advantage of the PostScript program language to design efficient PostScript programs and printer devices.

For More Information

APDA is Apple's worldwide source for over three hundred development tools, technical resources, training products, and information for anyone interested in developing applications on Apple platforms. Customers receive the quarterly *APDA Tools Catalog* featuring all current versions of Apple development tools and the most popular third-party development tools. Ordering is easy; there are no membership fees, and application forms are not required for most of our products. APDA offers convenient payment and shipping options, including site licensing.

To order products or to request a complimentary copy of the *APDA Tools Catalog*, contact

APDA
Apple Computer, Inc.
P.O. Box 319
Buffalo, NY 14207-0319

Telephone	800-282-2732 (United States) 800-637-0029 (Canada) 716-871-6555 (International)
Fax	716-871-6511
AppleLink	APDA
America Online	APDA
CompuServe	76666,2405
Internet	APDA@applelink.apple.com

CHAPTER 1

Overview of LaserWriter Pro Hardware

The LaserWriter Pro 600 and 630 printers are the next-generation replacements for the LaserWriter II printer. They provide higher print quality, better performance, a wider range of interfaces, and improved paper handling, at a lower cost than the LaserWriter IIf or the LaserWriter IIg.

Both LaserWriter Pro printers can operate with either Macintosh computers, or with DOS-based IBM personal computers. Both printers are available in 110-volt and 220-volt versions.

This chapter describes:

- hardware features of the printers
- communications ports and interfaces
- mode selection using the rotary switch

Features of the LaserWriter Pro Printer

Table 1-1 lists functional features of the LaserWriter Pro printer. Information listed applies to both the LaserWriter Pro 600 and 630, unless otherwise indicated.

Table 1-1 LaserWriter Pro features

Features	Specifications
Printing speed	8 pages per minute
First print time	Less than 18 seconds (full page)
Warm-up time	2 minutes (worst case)
Imaging	300/600 dpi (software switchable) Fine print and photograde available at 300 dpi
Processor	Motorola MC68EC030, 25 MHz
High-capacity paper handling	250-sheet cassette standard 500-sheet cassette optional 75-envelope feeder optional 100-sheet multipurpose tray standard
ROM	4 MB
DRAM	8 MB Expandable to 16 MB, 20 MB, or 32 MB
Simultaneously active interface ports	LaserWriter Pro 600 3 ports: RS-232 Serial, LocalTalk/RS-422, Centronics LaserWriter Pro 630 5 ports: RS-232 Serial, LocalTalk/RS-422, SCSI, Centronics, Ethernet
Communication rates	Up to 57,000 baud (RS-232)

Table 1-1 LaserWriter Pro features (continued)

Features	Specifications
Fonts	SuperPlus font set: 64 fonts TrueType font scaler Internal and external SCSI ports for hard-disk font storage on the LaserWriter Pro 630
PDL (Page Description Language)	PostScript™ Level-II, version 2011, release 130
Emulation	Built-in LaserJet IIP emulator. Compatible with PCL (Printer Control Language), Level 4

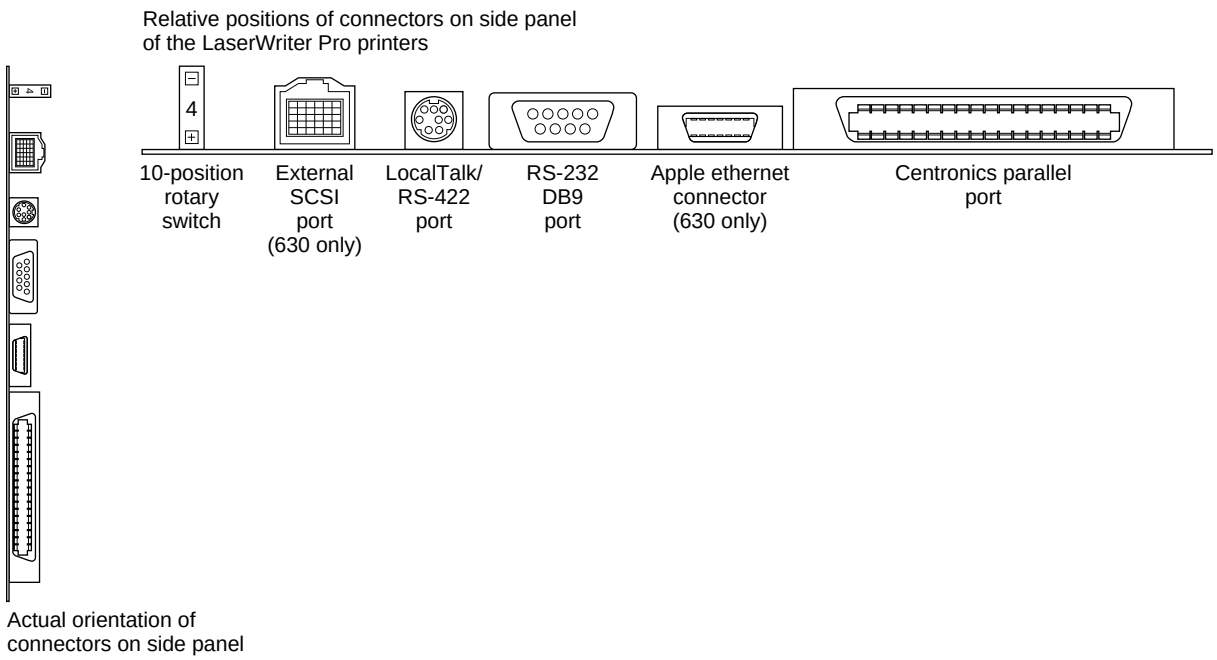
Communication

The LaserWriter Pro printer supports a variety of communication ports. Figure 1-1 shows the relative positions of these ports on the side panel of the printers. It also shows the actual vertical orientation of the connectors.

Note

The LaserWriter Pro 600 does not support all ports. ◆

Figure 1-1 LaserWriter Pro side panel connectors



The controller of the LaserWriter Pro 600 supports host communication via the following ports:

- Serial RS-422/LocalTalk port
- RS-232 9-pin serial port
- Centronics parallel port

The LaserWriter Pro 630 controller supports host communication via the following ports:

- External SCSI port for a local hard disk drive to store fonts.
- Serial RS-422/LocalTalk port
- RS-232 9-pin serial port
- EtherTalk implemented by the Apple Ethernet adapter
- Centronics parallel port

In addition to the external ports, each printer has an internal SCSI connector that supports an optional internal 2.5-inch hard disk drive used for storing fonts.

All ports are always active. The LaserWriter Pro controller supports simultaneous communication and arbitrates between the ports, accepting only one channel at a time. Simultaneous communication over all I/O channels makes it easier for the LaserWriter Pro to work in multicomputer environments.

Macintosh computers work over the LocalTalk or Ethernet channels, while IBM DOS or other networked machines can use LocalTalk or EtherTalk. Individual IBM DOS machines or servers can print directly through the Centronics parallel channel or through the RS-232 srial channel.

For information on how to configure the communication ports to suit a particular configuration, refer to the section “Mode Switch,” later in this chapter.

Serial Communications

The LaserWriter Pro printer provides two serial port connectors: one for RS-232C devices, and one for LocalTalk and RS-242 devices.

Serial Port for RS-232C Devices

A 9-pin sub-mini-DIN connector supports communication with RS-232C devices. Figure 1-2 shows the connector pin designations, and Table 1-2 lists the pin functions for the 9-pin connector.

Figure 1-2 9-pin serial port connector for RS-232C device

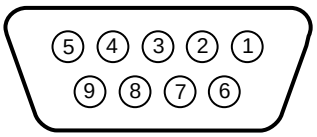


Table 1-2 Signal descriptions for 9-pin serial port

Pin number	Signal name	Description
1	DCD	Data carrier detect. If printer is used with modem, shows that the carrier signal is present.
2	/RXD	Receive data (inverted)
3	/TXD	Transmit data (inverted)
4	DTR	Data terminal ready. Send data when this signal is asserted.
5	GND	Ground
6	DSR	Data set ready. Indicates the modem is ready. Not used with the LaserWriter Pro printer.
7	RTS	Request to send. Flow control signal.
8	CTS	Clear to send. Flow control signal.
9	RING	Ring signal

LocalTalk/RS422 Serial Connector

An 8-pin mini-DIN supports LocalTalk and RS-242 devices. Figure 1-3 shows the connector pin designations for the 8-pin connector, and Table 1-3 lists the pin functions.

Figure 1-3 8-pin mini-DIN connector for LocalTalk and RS-242 devices

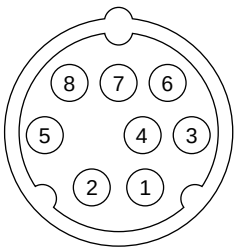


Table 1-3 Signal descriptions for LocalTalk and RS422 serial port

Pin number	Signal name	Description
1	HSKo	Handshake signal, output
2	HSKi	Handshake signal, input
3	/TXD	Transmit data (inverted)
4	GND	Signal ground
5	/RXD	Receive data (inverted)
6	TXD	Transmit data
7	GPi	General-purpose input
8	RXD	Receive data

Centronics Parallel Connector

The LaserWriter Pro 630 provides a 36-pin connector for communication with a standard Centronics parallel interface. Figure 1-4 shows the pin designations for this connector and Table 1-4 lists the signal descriptions.

Figure 1-4 Centronics parallel connector

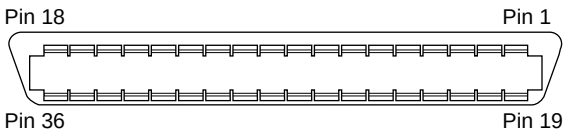


Table 1-4 Signal descriptions for Centronics parallel port

Pin number	Signal name	Description
1	/DATA STROBE	Strobe for input data
2	DATA 1	Data input bit 1
3	DATA 2	Data input bit 2
4	DATA 3	Data input bit 3
5	DATA 4	Data input bit 4
6	DATA 5	Data input bit 5
7	DATA 6	Data input bit 6
8	DATA 7	Data input bit 7
9	DATA 8	Data input bit 8
10	/ACKNOWLEDGE	Handshaking output signal; acknowledges receipt of data
11	BUSY	Busy output signal; indicates that a /DATA STROBE has been received, but that /ACKNOWLEDGE has not yet been given
12	PAPER ERROR	Output status signal. Indicates a paper error
13	SELECT OUT	Daisy-chained printer select signal; not used in the LaserWriter Pro printer, which is always selected
14	AUTO FEED	Indicates that paper should be fed automatically at the end of each line; not used in PostScript printers
15	SELECT IN	Daisy-chained printer select signal. Not used in the LaserWriter Pro printer, which is always selected.
16	SIGNAL GROUND	Signal ground
17	CHASSIS GROUND	Chassis ground
18	Not used	Not used
19-30	SIGNAL GROUND	Signal ground
31	/PRIME	Reset signal. Host asserts it to cancel the current job on this port.
32	/FAULT	Fault signal. Asserted if there is any printer problem that prvents printing, such as a paper jam, or out of paper error.
33-36	Not used	Not used

Apple Ethernet Adapter for EtherTalk (630 Only)

The LaserWriter Pro 630 controller supports AppleTalk over the Ethernet network. That communications link is referred to as EtherTalk. A 14-pin connector provides the interface to the Ethernet through Apple's AUI interface, which allows you to connect to ThinNet, 10-Base-T, and ThickNet cabling systems, using the appropriate Apple Ethernet adapter. Figure 1-5 shows the 14-pin connector, and Table 1-5 lists the signal descriptions.

Figure 1-5 14-pin Ethernet connector

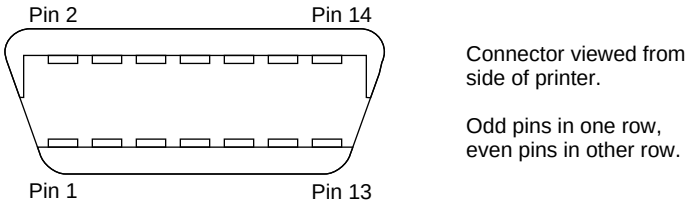


Table 1-5 Signal descriptions for Ethernet connector

Pin number	Signal name	Description
1	AAUI5V	+5 V power
2	RX+EXT	Receive data positive
3	RX EXT	Receive data negative
4	GND	Ground
5	CD+EXT	Carrier detect positive
6	CD EXT	Carrier detect negative
7, 8	AAUI5V	+5V power
9	TX+EXT	Transmit data positive
10	TX EXT	Transmit data negative
11	GND	Ground
12, 13	Open	Open line
14	AAU15V	+5 V power

SCSI Connector (630 Only)

The LaserWriter Pro 630 provides connections for up to seven hard-disk drives through a 30-pin, square SCSI connector. These disk drives provide nonvolatile storage for PostScript Level 2 resources, such as fonts. Figure 1-6 shows the pin designations for the connector, and Table 1-6 lists the signal descriptions.

Note
SCSI ID #6 cannot be used for any of these SCSI devices, since it is reserved for the LaserWriter Pro printer. ◆

Figure 1-6 30-pin SCSI connector pin designations

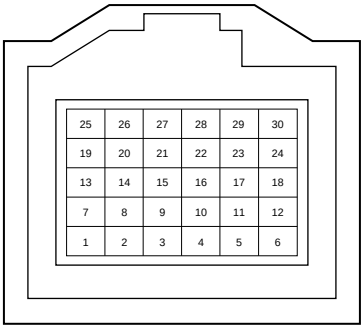


Table 1-6 Signal descriptions for 30-pin SCSI connector

Pin number	Signal name	Description
1	Open	Open line
2	/DB0	Bit 0 of SCSI data bus
3, 8, 10, 12, 13, 15, 17, 20, 22, 24	GND	Ground
4	/DB1	Bit 1 of SCSI data bus
5	Open	Open line
6	/DB2	Bit 2 of SCSI data bus
7	/DB3	Bit 3 of SCSI data bus
9	/ACK	Handshake signal; acknowledges a request for data transfer
11	/DB4	Bit 4 of SCSI data bus
14	/DB5	Bit 5 of SCSI data bus
16	/DB6	Bit 6 of SCSI data bus
18	/DB7	Bit 7 of SCSI data bus
19	/DBP	Parity bit of SCSI data bus
21	/REQ	Request for a data transfer
23	/BSY	When active (low) indicates that the SCSI bus is busy
25	/ATN	When active (low) indicates an attention condition

continued

Table 1-6 Signal descriptions for 30-pin SCSI connector (continued)

Pin number	Signal name	Description
26	/C/D	When active (low) indicates that data is on the SCSI bus; when high indicates that control signals are on the bus
27	/RST	SCSI bus reset
28	/MSG	When active (low) indicates the message phase is active
29	/SEL	Selects between target and initiator
30	/I/O	Controls the direction of data movement; when low data is output; when high data is input

Internal SCSI Connector

The LaserWriter Pro printer provides an internal 40-pin SCSI connector. It is mounted on the surface of the printer's printed circuit board, above the rotary switch. This connector enables you to install an internal 2.5-inch hard-disk drive to store fonts, and other PostScript Level 2 resources. Figure 1-7 shows the relative position of the connector, and the connector pins. Table 1-7 lists the signal assignments for the connector.

Figure 1-7 Internal SCSI connector

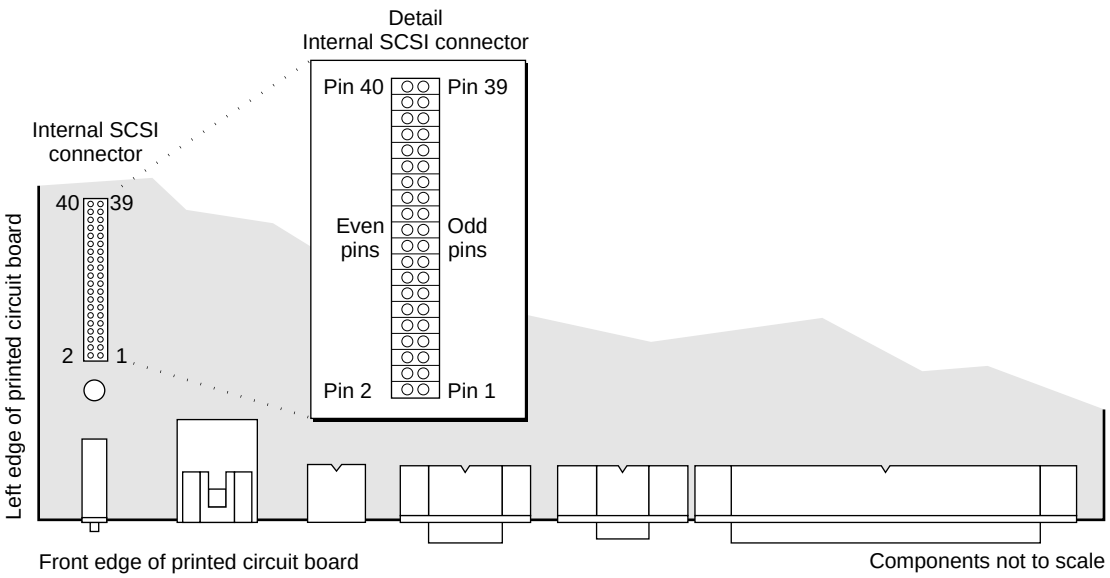


Table 1-7 Signal descriptions for internal SCSI connector

Pin number	Signal name	Description
1, 2, 24	+5V	+5 V power supply
3, 4, 5, 7, 9, 11, 13, 15, 19, 21, 23, 27, 31, 35	GND	Ground
6	/DB0	Bit 0 of SCSI data bus
8	/DB1	Bit 1 of SCSI data bus
10	/DB2	Bit 2 of SCSI data bus
12	/DB3	Bit 3 of SCSI data bus
14	/DB4	Bit 0 of SCSI data bus
16	/DB5	Bit 0 of SCSI data bus
17	Open	Open line.
18	/DB6	Bit 0 of SCSI data bus
20	/DB7	Bit 0 of SCSI data bus
22	/DB8	Parity bit of SCSI data bus
25	/ATN	When active (low) indicates an attention condition
26	/BSY	When active (low) indicates that the SCSI bus is busy
28	/SCSIACK	Handshake signal; acknowledges a request for data transfer
29	/RST	SCSI bus reset
30	/MSG	When active (low) indicates the message phase is active
32	/SEL	Selects between target and initiator
34	/C/D	When active (low) indicates that data is on the SCSI bus; when high indicates that control signals are on the bus
36	/REQ	Request for a data transfer
37, 38	MGND	Memory ground
39, 40	Motor5V	+5 V power supply for motor

Mode Switch

The LaserWriter Pro has a ten-position rotary wheel that allows you to set the printer to work in different communication environments. The first six positions (0–5) allow you to select a set of fixed parameters for each of the communication channels. You cannot change these sets of parameters using the PostScript operators. You may change the remaining sets of parameters (6–9). See Chapters 2 and 3 for further information. Table 1-8 shows the types of connections and the default parameter values for each switch setting.

Table 1-8 Switch settings and default parameter values

Switch setting	Port name	Type of connection and default parameter values	Mode
0	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript
1	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: DTR Protocol: raw	HP PCL 4
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: raw	HP PCL 4
2	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 19200 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript

Table 1-8 Switch settings and default parameter values (continued)

Switch setting	Port name	Type of connection and default parameter values	Mode
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript
3	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: DTR Protocol: raw	Not arbitrated for input.
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics. Protocol: raw	HP PCL 4
4	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 1200 baud rate Data bits: 7. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript
5	8-pin/RS-422	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	9 pin/RS-232	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript

continued

Table 1-8 Switch settings and default parameter values (continued)

Switch setting	Port name	Type of connection and default parameter values	Mode
6	8-pin/RS-422	Serial 19200 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	9 pin/RS-232	Serial 19200 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: normal	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript
7	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 1200 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: ETX/ACK Protocol: normal	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: normal	PostScript
8	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 1200 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: DTRF Protocol: raw	HP PCL 4
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics Protocol: raw	HP PCL 4

Table 1-8 Switch settings and default parameter values (continued)

Switch setting	Port name	Type of connection and default parameter values	Mode
9	8-pin/RS-422	LocalTalk	PostScript
	9 pin/RS-232	Serial 9600 baud rate Data bits: 8. No parity Stop bits: 1 Flow control: XON/XOFF Protocol: BSP	PostScript
	14-pin Apple Ethernet adapter	EtherTalk	PostScript
	36-pin Parallel	Centronics. Protocol: BSP	PostScript

You can find out the number of the current switch position by:

- Using the PostScript Level 2 system parameter `PrinterMode`.
- Choosing Configure Communication from the Utilities menu of the LaserWriter Pro utility program.
- Looking at the switch on the back of the printer.

IMPORTANT

You should change mode switch settings only between jobs. Changing the mode switch during operation immediately affects the printing in progress. The PostScript language interrupt aborts the job, and the printer looks for a new job with the communication parameters designated by the new switch settings, which become active after two seconds. If the host computer continues to run the job that was in progress, the data it sends to the printer may cause unpredictable results.

In addition, if you change the switch settings on a LaserWriter Pro that is connected to an AppleTalk network, other users on the network may not be aware of the new communication parameters, and this could also cause unpredictable results. ▲

Status Lights

The LaserWriter Pro has four colored lights on the left side of the printer. These lights indicate what the printer is doing. Figure 1-8 shows a view of the status light symbols, and Table 1-8 describes the functions of the lights.

Figure 1-8 Status lights

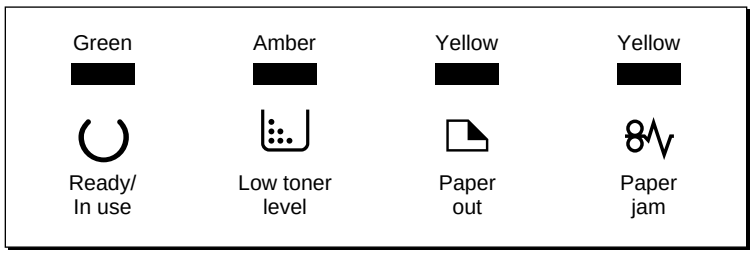


Table 1-9 Status light messages

Light	Light's state	Printer's state
Ready / In Use Green	On	The printer is ready to use.
	Off	Printer cannot print because there is an error condition, or the printer cover is open.
	Flashing	Indicates one of several conditions: the printer is warming up, it is printing a startup page, or it is processing data for the next print job.
Low Toner Level Amber	On	Toner is low or needs to be redistributed.
	Off	Toner level is all right.
	Flashing	Toner cartridge is installed incorrectly.
Paper Out Yellow	On	Paper tray is empty, or has been removed from the printer.
	Off	There is an adequate supply of paper in the paper tray.
	Flashing	The printer is in manual-feed mode, and is ready for the next sheet of paper.
Paper Jam Yellow	On	There is a paper jam.
	Off	Paper is feeding correctly through the printer.
	Flashing	Printer requires service.

NOTE If both the Paper Out and Paper Jam lights flash, or if they both stay on, the printer requires service.

DRAM Expansion

You may expand the LaserWriter Pro printer's DRAM capacity using SIMM (single in-line memory module) cards, which are plugged in to the slots on the surface of the printer's main circuit board. As shown in Figure 1-9, there are two slots for SIMM cards. Slot 1 is the slot closer to the edge of the circuit board.

Note

There is no silk-screened identification of these slots. ◆

To expand DRAM capacity, you must use one of the configurations listed in Table 1-10.

Figure 1-9 Expansion slots for SIMM cards

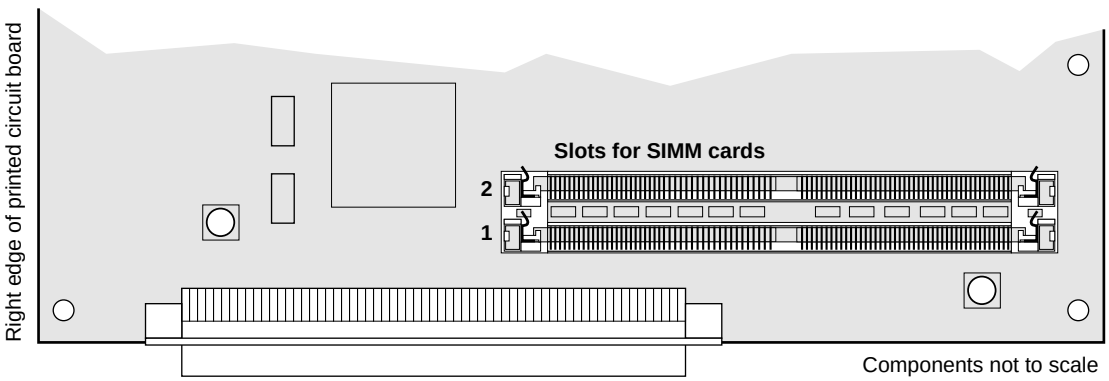


Table 1-10 DRAM expansion configurations

Total expansion	SIMM card capacity	SIMM card type	Number of SIMM cards used	Slots used
8 MB	4 MB	Single-sided	2	1 and 2
8 MB	8 MB	Double-sided	1	1
16 MB	16 MB	Single-sided	1	1
20 MB	16 MB	Single-sided	1	1
	4 MB	Single-sided	1	2
32 MB	16 MB	Single-sided	1	1
	16 MB	Single-sided	1	2
32 MB	32 MB	Double-sided	1	1

CHAPTER 2

Overview of LaserWriter Pro Software

This chapter provides an overview of the following software features:

- Adobe PostScript Programming Language
- LaserWriter Pro driver
- LaserWriter Pro utility program
- Startup and configuration pages
- Page types
- LaserJet IIP emulation
- Mode selection

Adobe PostScript Programming Language

The LaserWriter Pro printer uses Adobe PostScript Version 2010.130. This version of the PostScript language has features and capabilities that might not be present in other PostScript output devices. This developer note describes only the supplementary PostScript language features of the LaserWriter Pro. You should use the note in conjunction with the *PostScript Language Reference Manual* (published by Addison Wesley), and the *PostScript Language Supplement for Version 2010*.

Chapter 3 of this note, "LaserWriter Pro Software," provides detailed information about the features specific to the LaserWriter Pro.

LaserWriter Pro Driver

The LaserWriter Pro driver and Print Manager provide a general printer interface to the LaserWriter Pro printer. The interface should meet the needs of most Macintosh applications.

The driver

- provides full support for PostScript level 2
- allows the printer to switch between different quality levels
- supports multiple bins, a multipurpose paper tray, and an envelope feeder
- enables the printer to report paper size in the standard and optional cassettes to the user
- presents printer jam status if reported back by the printer
- supports optical density control through the video interface
- supports both TrueType and Type 1
- is compatible with version 7.X of the Macintosh LaserWriter driver

LaserWriter Pro Utility Program

The LaserWriter Pro utility program shipped with each LaserWriter Pro printer allows you to control and configure the printer. It is similar to the utility for the LaserWriter IIg. Upgrades to the LaserWriter Pro Utility program allow you to:

- select HW Resolution, 300 or 600.
- turn Automatic Tray Switching on and off.

Startup Page

When the LaserWriter Pro printer is powered up, it normally prints a start up page that contains information about the printer's hardware configuration. The page includes the name and model of the printer, the various communications channel parameters, the number of fonts in ROM, and the total amount of installed RAM.

You can prevent the startup page from being printed using any of the following methods:

- choose Set Startup Page from the Utilities menu in the LaserWriter Utility program.
- set the `DoStartPage` system parameter of the PostScript Level 2 operator, `setsystemparams`, to `false`
- set the PostScript Level 1 compatibility operator `dostartpage` in `statusdict` to `false`

Configuration Page

The configuration page, shown in Figure 2-1 on page 22, describes the current communications parameters and other values stored in the printer's nonvolatile memory. You can print the configuration page by choosing Print Configurations Page from the Utilities menu of the LaserWriter Pro Utility program. Table 2-1 on page 23 summarizes the functions of each block in the configuration page.

Figure 2-1 Layout of configuration page

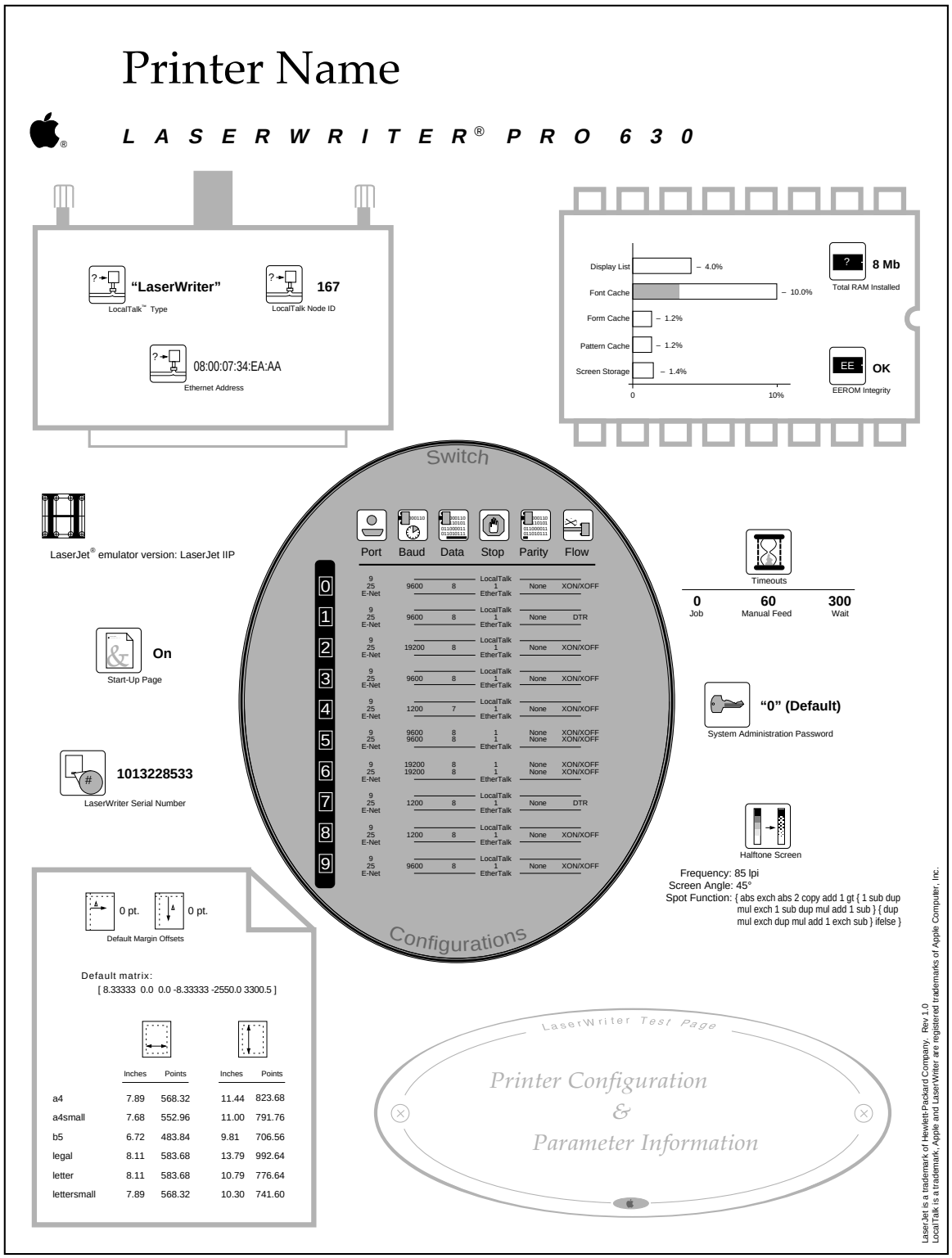


Table 2-1 Configuration page functions

Block	Description
Printer name	Shows the printer's name, as specified by the `printername` operator. Below the name are the Apple logo and the model name, LaserWriter Pro 600 or LaserWriter Pro 630.
Communication parameters	Indicates which communication protocol the LaserWriter Pro is using: ■ LocalTalk type ■ LocalTalk Node ID ■ Ethernet Address
Memory parameters	Shows how much DRAM is installed, the condition of the EEROM integrity, and a bar graph of memory allocation in the five following categories: ■ display list ■ font cache ■ form cache ■ pattern cache ■ screen storage
Emulation parameters	Shows the following information: ■ available emulators and their version numbers, for example LaserJet emulator version: Laser Jet IIP ■ startup page setting, On or Off ■ LaserWriter serial number
Switch configurations	Shows the parameter values for each of the different switch settings, as listed in Chapter 1, Table 1-8.
Miscellaneous parameters	Shows the values of the following parameters, in seconds: ■ job timeout ■ manual feed timeout ■ wait timeout It also displays ■ the system administrator's password; if no password has been assigned, it displays "0" Default ■ halftone screen information: frequency, screen angle, spot function
Page parameters	Shows the following page information: ■ default margin offsets, in points ■ default matrix values of the transformation matrix, that transforms user coordinate space into device space ■ page sizes, the width and length, in points and inches, of the image areas of the different page types: a4, a4small, b5, legal, letter, lettersmall

Page Types

The page types for the LaserWriter Pro are the same as those described in Chapter 4 of the *LaserWriter Reference*.

At the beginning of each job, the server selects the default paper tray, as assigned by the `defaultpapertray` operator. (See "Compatibility Operators for Setting System Parameters," in Chapter 3.) If the default is the main cassette, the server can detect its size and install the appropriate image region. If the default is the multipurpose tray, the server uses the image region most recently installed by means of the `setdefaultmultipurposepapertraysize` operator.

When the multipurpose tray is selected in this way, or by using the `setpapertray operator`, it is treated like the main cassette. Several sheets of paper may be stacked in it, and it feeds continuously until it is empty, at which time the paper-out light comes on. If a job requires a particular paper size, it should invoke one of the paper tray selection operators listed in Table 2-2 before it generates an image. That paper tray selection stays in effect for the duration of the job. The server restores the default paper tray selection when that job is finished.

Table 2-2 Paper tray selection operators

Operator	Description
a4tray	Selects the paper tray containing A4-size paper and set the page type to either `a4` or `a4small`, depending on the value of pagetype. This operator raises the PostScript language error `rangecheck` if no paper tray contains A4-size paper.
b5tray	Selects the paper tray containing B5-size paper and sets the page type to `b5`. This operator raises the PostScript language error `rangecheck` if no paper tray contains B5-size paper.
legaltray	Selects the paper tray containing legal-size paper and sets the page type to `legal`. This operator raises the PostScript language error `rangecheck` if no paper tray contains legal-size paper.
lettertray	Selects the paper tray containing letter-size paper and sets the page type to either `letter` or `lettersmall`, depending on the value of *pagetype*. This operator raises the PostScript language error `rangecheck` if no paper tray contains letter-size paper.

LaserJet IIP Emulation

The LaserWriter Pro printer has a built-in Hewlett Packard LaserJet IIP emulator, which is compatible with Hewlett-Packard PCL (Printer Control Language), Level 4. Chapter 3 of this developer note, "LaserWriter Pro Software," describes the features of the emulator. Chapter 3 of the *LaserWriter Reference* provides more information on the subject.

Selecting Emulation

You may invoke the emulation in one of three ways. Each method is described briefly below. Before beginning the emulation, the `emulate` procedure erases the current page and initializes the graphics state. It also clears the operand stack and the dictionary stack.

Running an emulator consumes some PostScript virtual memory. If `emulate` returns normally with no interrupt that virtual memory is reclaimed. If `emulate` is invoked when there is too little virtual memory, a virtual memory error, `VMerror`, occurs. You should not call the `emulate` procedure when the printer is in interactive mode.

Setting the Server to Emulation Mode

You may set the server to an emulation mode, as described in the *LaserWriter Reference*. This method has the advantage of making the communication protocol the same as that of the printer being emulated. However, it has the disadvantage that the channel must be closed and reopened when you switch modes using the mode switch. When the channel is closed, all buffered data is flushed out, and any data sent from the host before the channel is reopened is lost. This method is not appropriate if the host computer frequently switches between sending PostScript language programs and receiving emulation input.

Selecting From Within a PostScript Language Program

You may invoke the emulation from within a PostScript language program. This method means the host computer can switch back and forth between PostScript language programs and emulation inputs, and both types of data can be in the printer's input buffer at the same time without data loss.

A PostScript language program can invoke the emulator using a `statusdict` procedure called `emulate`. This procedure takes a file and an emulator name from the operand stack. The file is the input source for the emulation. The emulator name operand selects the emulation to be invoked. To select the Hewlett-Packard LaserJet IIP emulator, you must use the name `/LaserJet IIP`. To maintain compatibility with older printers, the name /hpcl will also select the LaserJet IIP emulator. The following example shows how to select the LaserJet emulator.

```
currentfile /LaserJetIIP statusdict /emulate get exec
```

The emulate procedure returns

- at the end of the job
- when a Control-D is encountered in the input source. In this case, that Control-D marks the end of the job
- when <ESC><DEL>-0 is encountered in the input source. In this case, the PostScript interpreter ejects the current page and continues executing whatever was on the execution stack before it executed the `emulate` procedure. A Control-D must still be sent to terminate the job.

Using a Serial Connection

You may invoke the emulation in binary mode over a serial connection, such as an asynchronous RS-232 or RS-422 connector, or during an AppleTalk PAP (Printer Access Protocol) session.

To invoke the emulation in this way, the rotary switch should be in one of the following positions:

- position 6, 7, 8, or 9, to select the binary protocol on the 25-pin connector
- any position where AppleTalk is selected for the Ethernet or mini DIN-8 port

Note
You cannot use the Normal serial protocol for PostScript language programs because several of the control characters have a special meaning. ◆

LaserJet IIP Emulation Differences

The Hewlett-Packard LaserJet IIP emulation has a number of features that the LaserJet IIP and LaserWriter implement in different ways. For a complete description of LaserJet Plus emulation, see “Using the Hewlett-Packard LaserJet Plus Emulator,” in Chapter 3 of the *LaserWriter Reference*.

Mode Selection Parameters

The LaserWriter Pro printer operates in a variety of communications environments. You may select the type of connection (LocalTalk, serial, EtherTalk, parallel) the parameter values (baud rate, and so on), and the mode (PostScript or HP PCL 4), using the ten-position rotary switch on the side panel of the printer. The parameters associated with switch positions 0–5 are fixed and you cannot change them. You will find information about the mode selection rotary switch in Chapter 1, under “Mode Switch.”

You may change the parameters for switch positions 6–9 in one of the following ways:

- By using the appropriate device parameter for the PostScript Level 2 `setdevparams` operator. Parameters and semantics for setdevparams are described in Chapter 3, under "PostScript Level 2 for the LaserWriter Pro."

Note

This is the recommended procedure. ◆

- By using the PostScript Level 2 `setsccbatch` operator. Information about `setsccbatch` is provided in Chapter 3, under "Compatibility Operators for Setting SCC Parameters."

CHAPTER 3

LaserWriter Pro Software

This chapter describes the software parameters that enable you to set up and configure the LaserWriter Pro printer. They include:

- page device parameters
- interpreter parameters
- resource categories
- emulator parameters
- compatibility operators
- page size compatibility operators
- paper tray compatibility operators

The chapter also explains how to set system, page, user, device, and serial communication parameters; and how to encode the SCC operator.

Device Setup

The `setpagedevice` operator is used in PostScript page descriptions to specify processing requirements and select optional printer features. The `setpagedevice` operator can also be used to specify default device setup or configuration parameters that may be used when the page description does not specify the parameters.

The `currentpagedevice` operator is used to get the current accumulated values and the adjusted state of the page device. The parameters for the `setpagedevice` operator are cumulative, in that each new call to `setpagedevice` does not reset the state in total, but modifies it. In addition, on each call to `setpagedevice` the resulting accumulated page device state is processed to enable the printer to accomplish the required results. This may result in further modification of the page device state.

For more information about how the `setpagedevice` operator is used to specify the processing requirements of a document, refer to Section 4.11 of the *PostScript Language Reference Manual*.

Page Device Parameters

This section describes the LaserWriter Pro page device parameters. It covers the following topics:

- page device parameters present in the LaserWriter Pro printer; the semantics for all parameters appear in the *PostScript Language Reference Manual*, and the *PostScript Language Reference Manual Supplement*, for version 2010.
- PhotoGrade parameters
- FinePrint parameters

Page Device Parameter Summary

Table 3-1 lists the page device parameters, lists their defaults, and provides additional technical information.

Table 3-1 Page device parameters

Key	Type	Default	Comments
`BeginPage`	*procedure*	{pop}	See Note 1.
`EndPage`	*procedure*	{exch pop 2 ne}	See Note 2.
`ExitJamRecovery`	*boolean*	`false`	Value is constant and persistent across power cycles. See Note 3.
`HWResolution`	*array*	[600 600]	Controls the resolution of the output. Legal values are [300 300] and [600 600]. This key is treated as a request. The default policy is 2, signifying user interaction. In this case, the interaction is implicit and results in lower resolution. This key in the `pagedevice` directory still shows the requested resolution. The resolution actually achieved may be obtained by using the `HWResolution` procedure in the `ActualValues ProcSet`. Value is constant.
`ImagingBox`	*array or null*	`null`	In some configurations, this may be set to a default to reduce the imaging area allowed to less than a full page. See Note 2.
`InputAttributes`	*dictionary*	Depends on configuration	The values x and y depend upon which paper tray is installed. See Table 3-2. The multipurpose tray is always present. However, if the optional cassette tray assembly is installed but missing, the corresponding entry in the `InputAttributes` dictionary is set to null. This can only happen when the printer is turned on and the tray is not installed. Table 3-3 lists the slot numbers and corresponding input sources. If a job is sent to the printer and the tray is removed, the PostScript interpreter assumes a tray of the same size will be installed and sets the attributes accordingly. If a different tray is installed, the attributes change to reflect the characteristics of the new tray. There are values of matching tolerance for the `PageSize` parameter. See `PageSize` later in this table. See Note 2.

continued

Table 3-1 Page device parameters (continued)

Key	Type	Default	Comments
`Install`	*procedure*	`/DefaultHalftone` See the section "Install Procedure," later in this chapter, for more information.	This device parameter is associated with the PhotoGrade feature. See Note 2.
`ManualFeed`	*boolean*	`false`	See Note 2.
`ManualFeed Timeout`	*integer*	60	See Note 3.
`Margins`	*array*	[0 0]	Value is persistent across power cycles. See Note 2.
`MediaColor`	*string or null*	`null`	See Note 2.
`MediaType`	*string or null*	`null`	See Note 2.
`MediaWeight`	*number or null*	`null`	See Note 2.
`NumCopies`	*integer or null*	`null`	See Note 2.
`OutputFaceUp`	*boolean*	`false`	Value is persistent across power cycles. See Note 2.
`OutputPage`	*boolean*	`true`	See Note 2.
`PageSize`	*array*	Depends on configuration	Matching tolerance: 5-default user space units. Landscape mode ([762 612]) is also valid. See Table 3-2 for further information.
`Policies`	*dictionary*	`<</PolicyNotFound 1 /PageSize 0 /HWResolution 2 /PolicyReport{pop}>>`	See Note 2.
`PostRendering Enhance`	*boolean*	`true`	Value is persistent across power cycles. See Note 3.
`PostRendering EnhanceDetails`	*dictionary*	See Tables 3-2, 3-5, 3-6, and 3-8	
`PreRendering Enhance`	*boolean*	`true`	Value is persistent across power cycles. See Note 3.

continued

Table 3-1 Page device parameters (continued)

Key	Type	Default	Comments
`PreRendering EnhanceDetails`	*dictionary*	See Tables 3-2, 3-5, 3-6, and 3-8	See Note 3.
`TraySwitch`	*boolean*	`false`	If `true`, automatic tray switching is provided. In this case, if a tray runs out of media (paper), another tray will be selected that satisfies the following requirements: ■ Input attributes, with the exception of MatchAll key, match exactly ■ The tray contains paper ■ Neither tray involved is an envelope feeder Value is persistent across cycles.

NOTE 1. See Section 4.11.6 of the *PostScript Language Reference Manual.*
NOTE 2. See Section 4.11.3 of the *PostScript Language Reference Manual*.
NOTE 3. See Chapter 2 of the *PostScript Language Reference Manual Supplement*.

Table 3-1 lists all the page device parameters available with the LaserWriter Pro printer. Table 3-2 shows the different page sizes, and indicates the memory occupied by the page sizes at a resolution of 300dpi.

Table 3-2 Paper sizes and memory usage

		Memory occupied at 300dpi rendering enhancement	
Paper size	**Name**	off	on
[612 792]	Letter	983844	3935376
[612 1008]	Legal	1257344	5029376
[595 842]	A4	1015872	4063488
[516 729]	B5	741888	2967552

Table 3-3 Paper tray slot numbers and input sources

Slot number	Input source
0	Main cassette
1	Multipurpose tray
2	500-sheet cassette
3	Envelope feeder

Product Strings

Table 3-4 lists values assigned to the product strings associated with the LaserWriter Pro printer.

Table 3-4 Product string values

String name	Type	Value
`languagelevel`	*integer*	2
`product`	*string*	LaserWriter Pro
`version`	*string*	2010.130

Install Procedure

This section provides sample code for the install procedure.

```
{ currentpagedevice dup /PreRenderingEnhanceDetails get
   /ActualPreRenderingEnhance get
     {/PreRenderingEnhance Details get
        /DefaultHalftone get /Halftone findresource
      } { pop << /SpotFunction { abs exch abs 2 copy add 1 gt
          {1 sub dup mul exch 1 sub dup mul add 1 sub }
          {dup mul exch dup mul add 1 exch sub }
           ifelse } /Halftonetype 1/Frequency /ActualValues
           /ProcSet findresource
           /HWResolution get exec 0 get 300 eq {60} {85} ifelse
           /Angle 45
         >>
   } ifelse sethalftone
   {} settransfer false setstrokeadjust
<< 300 /DefaultColorRendering300 600 /DefaultColorRendering600 >>/
ActualValues /ProcSet findresource /HWResolution get exec
   0 get get /ColorRendering findresource setcolorrendering
}
```

PhotoGrade Parameters

Three `pagedevice` parameters contain special information to support the PhotoGrade feature of the LaserWriter Pro printer. They are `Install`, `PreRenderingEnhance`, and `PreRenderingEnhanceDetails`. Table 3-5 defines these parameters.

The PhotoGrade feature improves gray rendering. It does this by controlling special hardware in the printer that converts 4-bit pixels in the frame buffer to pseudo-gray pixels on paper. The feature is available only in the 300 x 300 dpi resolution mode of LaserWriter Pro printer, and only when sufficient memory (8 MB) is available.

Table 3-5 PhotoGrade device parameters

Device parameter	Description
`Install`	Performs graphics state setup for the page device. If `PreRenderingEnhance` is `true` and the request is satisfied, `Install` sets the halftone in the graphics state to the dictionary identified by `DefaultHalftone` in `PreRenderingEnhanceDetails`. Otherwise, a standard halftone dictionary is used, with a frequency of 60 at 300 dpi, and 85 at 600 dpi.
`PreRenderingEnhance`	Indicates whether or not you have requested PhotoGrade. Page descriptions can change this value within `save/restore` boundaries.
`PreRenderingEnhanceDetails`	Contains information specific to PhotoGrade technology. See Tables 3-6 and 3-7.

To use this capability, you must allocate a 4-bit-per-pixel frame buffer instead of the usual 1-bit-per-pixel buffer.

Note

Allocating this amount of memory to the frame buffer places an additional computational burden on the host computer. It may also cause some degradation in performance, especially if you are rendering images with the `image` or `imagemask` operators. ◆

The PhotoGrade technology is invoked before PostScript language objects are rendered to the frame buffer. This is why it is considered pre-rendering enhancement.

The `PreRenderingEnhance` entry in the page device dictionary is treated as a request, not an assertion. The rendering enhancement request is considered in conjunction with the page size request, the value of `HWResolution`, and the amount of memory installed. See Table 3-2.

To make the best use of the special hardware required by PhotoGrade, you should select a different halftone screen frequency. To do this, the default `Install` procedure used by the `setpagedevice` operator reads the value of `ActualPreRenderingEnhance`. If it is `false`, the halftone dictionary specified by `Install` is used as an argument to the `sethalftone` operator. If it is `true`, the name of a `Halftone` resource instance is obtained from the `PreRenderingEnhanceDetails` dictionary and that name is used to fetch a `Halftone` resource instance. The *PostScript Language Reference Manual* provides more information on available halftone resources. Table 3-6 shows frequently used entries in the `PreRenderingEnhanceDetails` dictionary.

Table 3-6 Frequently used entries in the `PreRenderingEnhanceDetails` dictionary

Key	Type	Default	Description
`DefaultHalftone`	*name*	106 x 45	Name of a valid `Halftone` resource instance used by `Install`
`Type`	*integer*	1	Indicates where PhotoGrade information is found and how it is represented

There are additional entries in the `PreRenderingEnhanceDetails` dictionary that you should change only with an application program like LaserWriter Utility, which is supplied with the printer. These parameters compensate for such factors as atmospheric humidity and toner life. Table 3-7 summarizes these parameters.

Table 3-7 PhotoGrade calibration parameters

Key	Type	Semantics
S1	String	60-byte string of encoded values
S2	String	96-byte string of encoded values
S3	String	96-byte string of encoded values
S4	String	256-byte string of encoded values

NOTE The value of each of these keys is persistent across power cycles.

FinePrint Parameters

The FinePrint feature increases the apparent resolution of the printer by adjusting the dots placed by the laser. Special hardware acts on the frame buffer, which is produced by rendering a PostScript language program, to produce smoother edges. The FinePrint feature is dependent upon resolution, and is only available in the 300 x 300 resolution mode of the LaserWriter Pro printer.

This feature is considered a post-rendering enhancement. Unlike PhotoGrade, the anti-aliasing hardware controlled by `PostRenderingEnhance` does not incur any memory or computational penalty.

Two `pagedevice` parameters in the LaserWriter Pro printer contains special information to support the FinePrint features of the printer. These parameters are `PostRenderingEnhance` and `PostRenderingEnhanceDetails`. See Table 3-8.

Table 3-8 FinePrint device parameters

Device parameter	Description
`PostRenderingEnhance`	Indicates whether or not you have requested FinePrint. You can only do this when the printer is in 300 x 300 resolution mode. Page descriptions can change this value within `save/restore` boundaries.
`PostRenderingEnhanceDetails`	Contains two entries: `Type` and `ActualPostRenderingEnhance`. You should consider this a read-only dictionary, the main purpose of which is to indicate whether or not a request for FinePrint succeeded.

NOTE See Chapter 2 of the *PostScript Language Reference Manual Supplement*.

Interpreter Parameters

Certain parameters control the operation and behavior of the PostScript interpreter. Many of them are connected with memory allocation and other specific-purpose resources. For instance, interpreter parameters control the maximum amount of memory allocated to virtual memory, font cache, and halftone screens.

The LaserWriter Pro printer is configured initially with interpreter parameter values appropriate for most applications. However, using a PostScript language program, you can alter the interpreter parameters to favor certain applications, or to adapt the printer to special requirements. There are three classes of interpreter parameters: user, system, and device.

Each class has a PostScript language operator to read the parameter values and an operator to set parameter values. There are six resulting operators: `currentuserparams`, `setuserparams`, `currentsystemparams`, `setsystemparams`, `currentdevparams`, and `setdevparams`.

You will find information on parameter semantics in the *PostScript Language Reference Manual*, and the *PostScript Language Reference Manual Supplement*.

User Parameters

Within reasonable limits, you can change user parameters without a special authorization or password, using any PostScript Language program. User parameters establish temporary policies on issues such as size limits, and inserting new items into caches.

The `setuserparams` operator sets user parameters, and the `currentuserparams` operator reads their current values. Unless otherwise indicated, all user parameters are subject to `save` and `restore` boundaries. `Restore` resets all user parameters to their values at the time of the matching `save`. The initial value of the user parameters when the printer is turned on for the first time depends upon the product. Table 3-9 lists the user parameters present in the LaserWriter Pro printer.

Table 3-9 User parameters

Key	Type	Default	Details
`JobName`	*string*	()	≤ 32 characters
`JobTimeout`	*integer*	0	≥ 0
`MaxDictStack`	*integer*	530	≥ 0
`MaxExecStack`	*integer*	10015	≥ 0
`MaxFontItem`	*integer*	12500	≥ 0
`MaxFormItem`	*integer*	100000	≥ 0

continued

Table 3-9 User parameters (continued)

Key	Type	Default	Details
`MaxLocalVM`	*integer*	2147483647	≥ 0
`MaxOpStack`	*integer*	10000	≥ 0
`MaxPatternItem`	*integer*	20000	≥ 0
`MaxScreenItem`	*integer*	Varies	≥ 0. Initial value is 3000 bytes per MB of installed RAM, with a maximum of 12000 bytes.
`MaxUPathItem`	*integer*	5000	≥ 0
`MinFontCompress`	*integer*	1250	≥ 0
`VMReclaim`	*integer*	0	0, -1, -2
`VMThreshold`	*integer*	40000	≥ 0
`WaitTimeout`	*integer*	0	≥ 0

NOTE Refer to the *PostScript Language Reference Manual*, and to the *PostScript Language Reference Manual Supplement*, for further information.

System Parameters

System parameters alter the overall configuration of the printer. For certain parameters, as shown in Table 3-10, changes persist, even when you turn the printer off and on again. Other parameters return to default values when the printer is turned off. You can set system parameters using the `setsystemparams` operator and read them using the `currentsystemsparams` operator. You must use a password to change system parameters. System parameters are not subject to `save` and `restore`, and their values persist across jobs. Table 3-10 lists the system parameters present in LaserWriter Pro printer.

Table 3-10 System parameters

Key	Type	Default	Details
`BuildTime`	*integer*	N/A	Read only. Identifies date ROM was created.
`ByteOrder`	*boolean*	`false`	Read only.
`CurDisplayList`	*integer*	`0`	Read only. Identifies amount of RAM currently occupied by the display list.
`CurFontCache`	*integer*	`0`	Read only. Identifies amount of RAM currently occupied by the font cache.
`CurFormCache`	*integer*	`0`	Read only. Identifies amount of RAM currently occupied by the form cache.

continued

Table 3-10 System parameters (continued)

Key	Type	Default	Details
`CurOutlineCache`	*integer*	`0`	Read only. Identifies the amount of RAM currently occupied by the outline cache.
`CurPatternCache`	*integer*	`0`	Read only. Identifies the amount of RAM currently occupied by the pattern cache.
`CurScreenStorage`	*integer*	`0`	Read only. Identifies the amount of RAM currently occupied by screen storage.
`CurSourceList`	*integer*	`0`	Read only. ≥0.
`CurUPathCache`	*integer*	`0`	Read only. ≥0.
`DoStartPage`	*boolean*	`true`	May be `true` or `false`. Value is persistent across power cycles.
`FactoryDefaults`	*boolean*	`false`	May be `true` or `false`. Value is persistent across power cycles for all parameters except `PageCount` and `serialnumber`, which are reset after power down.
`FatalErrorAddress`	*integer*	`0`	Hardware address of the last call to the fatal error handler.
`GenericResourceDir`	*string*	`(Resource/)`	Any valid file system prefix.
`JobTimeout`	*integer*	`0`	≥ 0. Value is persistent across power cycles.
`MaxDisplayList`	*integer*	Varies	≥0. Initial value is 4% of installed RAM. This number is recomputed when the RAM configuration changes.
`MaxFontCache`	*integer*	Varies	≥ 0. Initial value is based on the amount of RAM installed. It is 167,772 bytes for 4 MB RAM. Otherwise, it is 10% of installed RAM. This number is recomputed when the RAM configuration changes. Value is persistent across power cycles.
`MaxFormCache`	*integer*	`100000`	≥ 0
`MaxOutlineCache`	*integer*	`655536`	≥ 0
`MaxPatternCache`	*integer*	`100000`	≥ 0

continued

Table 3-10 System parameters (continued)

Key	Type	Default	Details
`MaxRasterMemory`	*integer*	`0`	≥ 0. The value indicates the largest amount of memory that may be allocated to the frame buffer. A value of `0` indicates that enough memory should be reserved for the largest achievable frame buffer, which is a 300 dpi legal page with 4 MB RAM, or a 600 dpi legal page with 8 MB RAM. The implementation ignores values that are too small, and guarantees that an A4small, lettersmall, or B5 size frame buffer can be allocated. Value is persistent across power cycles.
`MaxScreenStorage`	*integer*	Varies	≥ 0. Initial value is 30,000 bytes per MB of RAM installed, up to a maximum of 120,000 bytes. This number is recomputed when the RAM configuration changes. Value is persistent across power cycles.
`MaxSourceList`	*integer*	`24576`	≥ 0
`MaxUPathCache`	*integer*	`300000`	≥ 0
`PageCount`	*integer*	`0`	Read only. ≥ 0. Indicates how many pages have been successfully delivered.
`PrinterMode`	*integer*	Depends on configuration	Read only. Returns the value of the rotary switch setting. The range is `0-9`.
`PrinterName`	*string*	`String`	Any string ≥ 32 characters, : and @ are not allowed.Value is persistent across power cycles.
`RamSize`	*integer*	Varies	Read only. ≥ 0. Indicates the amount of RAM installed.
`RealFormat`	*string*	`(IEEE)`	Read only. IEEE.
`Revision`	*integer*	`1`	Read only. Indicates ROM revision number.
`StartJobPassword`	*string*	`null`	Any string ≥ 32 characters is not allowed. Value is persistent across power cycles.
`StartupMode`	*integer*	`1`	If `0`, do nothing. If `1`, then find the file Sys/Start (using `Search Order`) and execute it. Value is persistent across power cycles.
`SystemParamsPassword`	*string*	`null`	Any string ≥ 32 characters is not allowed. Value is persistent across power cycles.

continued

Table 3-10 System parameters (continued)

Key	Type	Default	Details
`StartJobPassword`	*string*	`null`	Any string ≥ 32 characters is not allowed. Value is persistent across power cycles.
`ValidNV`	*boolean*	`true`	Refer to the *Program Language Reference Manual Supplement*.
`WaitTimeout`	*integer*	`60`	≥ 0. Value is persistent across power cycles.

Device Parameters

You may set device parameters using the `setdevparams` operator, and you may read them using `currentdevparams`. Like system parameters, device parameters require a password, are global to the PostScript environment, and have the same persistence characteristics. Some device parameters can be stored in nonvolatile memory.

Device parameters are different from both system and user parameters in that device parameters may be interdependent. This means that the legality of a given parameter may depend on the value of another parameter.

Device parameters fall into sets that correspond to a particular device, `such as %Serial%, %disk2%`, and so on. Even if two products have the same device, the parameters in the set might be different because the hardware support for that device is different.

File System

The file system supported by the LaserWriter Pro printer is described in the *PostScript Language Reference Manual*. The following restrictions apply to the file system:

- Filenames cannot end with a slash (/), or contain adjacent slashes (//).
- Filenames may not contain the colon character (:)
- There can be no more than 31 non-slash characters between each slash character, and the total number of characters in a filename may not exceed 255.
- The printer itself uses SCSI ID 6, so you may not use this ID for any other device.
- The LaserWriter Pro file system requires that you close a file before deleting it. Otherwise, an `invalidfileaccess` error occurs.
- When you reformat a hard disk attached to the printer, you should turn off the printer, and then turn it on again.

Note

Several parameters are in units of pages. A page is exactly 1024 bytes. ◆

Table 3-11 lists the parameters common to all FileSystem devices.

Table 3-11 Parameters common to all FileSystem devices

Key	Type	Details
`Free`	*integer*	Read only. Indicates the amount of free space (in pages for disks, and bytes for cartridges) on the media device. Legal values: Any non-negative integer Errors: None
`HasNames`	*boolean*	Read only. Indicates whether the device supports files. If `false`, the device is a single entity of data. Legal values: `true, false` Errors: None
`InitializeAction`	*integer*	Specifies an action for initializing the device. The following values are valid for disks: `0` indicates no action, and it is the value returned when the parameters is read. `1` indicates that the current file system (if there is one) should be deleted, and a new one of the size specified `Logical Size` be created. `2` reformats the entire medium before creating a new file system of the size specified by `LogicalSize`. `3` or greater has the same effect as the value `2`, and it also carries out product-dependent actions, which typically consist of reformatting the disk and running integrity tests before creating the file system. Some devices can have additional parameters that serve as arguments to `InitializeAction`. The following values are valid for cartridges: `0` indicates no action, and it is the value returned when the parameters is read. `1` reformats the entire medium before creating a new files system of the size specified by PhysicalSize. Legal values: Any non-negative integer Errors: None
`LogicalSize`	*integer*	When set, specifies the size of the file system to be created. It is used as an argument to the action carried out by `InitializeAction`. If `LogicalSize` is `0`, `InitializeAction` uses a default size that is normally the size of the entire device. See also `InitializeAction`. When queried, this parameter indicates the current size of the file system on the device (in pages for disks, bytes for cartridges). A value of `0` indicates that there is no valid file system.

continued

Table 3-11 Parameters common to all FileSystem devices (continued)

Key	Type	Details
`LogicalSize` *(continued)*		If `LogicalSize` is set with a certain value and the device is then reformatted, a query of `LogicalSize` should return the value that was set. However, if the parameter is queried at any time before the device is reformatted, it may return a different value from the one that was set, because it will return the current size of the device. Legal values: Any non-negative integer Errors: `rangecheck, typecheck`
`Mounted`	*boolean*	If this parameter is set to `true`, the system attempts to mount the device. If it is set to `false`, the system attempts to dismount the device. Depending on the type of device, mounting a device makes it known to the system, and makes it at least possible to read it. A device will not mount successfully if it does not contain a valid file system. When queried, the return value indicates whether the device is currently mounted. You can obtain the result of an attempted mount by querying `Mounted` immediately after setting it. Legal values: Any non-negative integer Error: `typecheck`
`PhysicalSize`	*integer*	Read only. Indicates the size (in pages for disks, and bytes for cartridges) of the media. Legal values: Any non-negative integer Errors: None
`Removable`	*boolean*	Read only. Indicates whether the media of the device can be removed. Legal values: `true,false` Errors: None
`Searchable`	*boolean*	Read only. Indicates whether the device participates in searches during operation that require a device, but for which no device has been specified. Legal values: `true, false` Errors: None
`SearchOrder`	*integer*	Read only. Indicates the priority at which the device participates when searching for a file during operations in which no device has been specified. This parameter is ignored if the `Searchable` parameter is `false`. Legal values: Any non-negative integer Errors: None

continued

Table 3-11 Parameters common to all FileSystem devices (continued)

Key	Type	Details
`Type`	*name*	Read only. This constant always returns a value of `FileSystem`. Legal values: `FileSystem` Errors: None
`Writeable`	*boolean*	Read only. Indicates whether you can currently write to the device. Usually this boolean is `true` only if the media is physically able to be written to, and is not write protected. Legal values: `true, false` Errors: None

Communication Channels

The LaserWriter Pro 600 printer has three communication ports: an RS-422 serial port, an RS-232 serial port, and a Centronics 25-pin parallel port. These ports implement the following communication channels:

- `%LocalTalk%` using the RS-422 serial port in LocalTalk mode
- `%Serial%`, and `%SerialB%` using the RS-422 serial port in LocalTalk mode
- RS-422 protocols using the RS-422 port
- RS-232 protocols using the RS-232 port
- `%Parallel%` using the Centronics parallel port

The LaserWriter Pro 630 printer has five communication ports: an RS-422 serial port, an RS-232 serial port, a Centronics 25-pin parallel port, a 14-pin Ethernet connector, and an external 30-pin SCSI connector. These ports implement the following communication channels:

- `%LocalTalk%` using the RS-422 serial port in LocalTalk mode
- `%Serial%`, and `%SerialB%` using the RS-422 serial port in LocalTalk mode
- RS-422 protocols using the RS-422 port
- RS-232 protocols using the RS-232 port
- `%Parallel%` using the Centronics parallel port
- `%EtherTalk%` using the Ethernet connector
- the external SCSI port supports a local hard disk drive to store fonts

Each channel has three related parameter sets: nonvolatile, pending, RAM.

- with nonvolatile parameter sets, the set names contain the suffix `_NV`. For example, the nonvolatile parameters associated with `%SerialB%` are in a parameter set called `%SerialB%_NV`. This is a read/write set, and it allows you to make changes that persist across system restarts.

- with pending parameter sets, the set names contain the suffix `_Pending.` This is a read-only set, and it provides information about the settings that will be in use at the beginning of the next job on the corresponding channel, assuming that there have been no previous parameter changes.
- with RAM parameter sets, the set names have no suffix. This is a read/write set, and it allows you to make changes to the printer that last only until the printer is rebooted.

Note

The `hardwareiomode` and `sethardwareiomode` compatibility operators are provided for PostScript Level 1 applications compatibility only. Since Level 1 implementations of PostScript are backed up in EEROM, these implementations are simulated in Level 2 using the nonvolatile parameters sets. You may therefore access them only from the nonvolatile parameters switch setting on the LaserWriter Pro printer (switch position 6). ◆

Note

Since `sethardwareiomode` provides backward compatibility and is not supported on previously existing EtherTalk implementations, it does not support the `%EtherTalk_NV%` parameter set. ◆

Table 3-12 summarizes the relationship between the sets.

Table 3-12 Parameter set relationships

Changes	Nonvolatile set	Pending set	RAM set
Write-through operations on the RAM set	Not affected	Contents affected	—
Write-through operations on the nonvolatile set	—	Not affected	Contents affected
Write-through operations on the pending set	Contents affected	—	Contents affected if the mode switch is in the position that corresponds to the nonvolatile set being changed

Tables 3-13 through 3-17 list the factory default values of the parameter sets. The values of the RAM and pending sets are determined by the values of the nonvolatile set. The *PostScript Language Reference Manual Supplement* describes the semantics for each parameter.

Table 3-13 `%Serial_NVx%` communication parameter sets

Key	Type	%Serial_NV% default	%Serial_NV2% default	%Serial_NV3% default	%Serial_NV4% default
`Baud`	*integer*	`19200`	`1200`	`1200`	`9600`
`DataBits`	*integer*	`8`	`8`	`8`	`8`
`Enabled`	*boolean*	`true`	`true`	`true`	`true`
`FlowControl`	*name*	`XonXoff`	`EtxAck`	`Dtr`	`XonXoff`
`Interpreter`	*name*	`PostScript`	`PostScript`	`LaserJet IIP`	`PostScript`
`On`	*boolean*	`true`	`true`	`true`	`true`
`Parity`	*name*	None	None	None	None
`Protocol`	*name*	`Normal`	`Normal`	`Raw`	`Binary`
`StopBits`	*integer*	`1`	`1`	`1`	`1`

NOTE LaserWriter Pro printer does not support `CheckParity`.

Table 3-14 `%SerialB_NVx%` communication parameter sets

Key	Type	%Serial_NV% default	%Serial_NV2% default	%Serial_NV3% default	%Serial_NV4% default
`Baud`	*integer*	`19200`	`9600`	`9600`	`9600`
`DataBits`	*integer*	`8`	`8`	`8`	`8`
`Enabled`	*boolean*	`true`	`false`	`false`	`false`
`FlowControl`	*name*	`XonXoff`	`XonXoff`	`XonXoff`	`XonXoff`
`Interpreter`	*name*	`PostScript`	`PostScript`	`PostScript`	`PostScript`
`On`	*boolean*	`true`	`false`	`false`	`false`
`Parity`	*name*	None	None	None	None
`Protocol`	*name*	`Normal`	`Normal`	`Normal`	`Normal`
`StopBits`	*integer*	`1`	`1`	`1`	`1`

NOTE LaserWriter Pro printer does not support `CheckParity`.

Table 3-15 %LocalTalk_NVx% communication parameter sets

Key	Type	%LocalTalk_NV% default	%LocalTalk_NV2% %LocalTalk_NV3% %LocalTalk_NV4% defaults
LocalTalkType	*string*	LaserWriter	LaserWriter
Enabled	*boolean*	false	true
Interpreter	*name*	PostScript	PostScript
On	*boolean*	false	true

NOTE LocalTalkType parameters refer to the same nonvolatile storage. Changing the LocalTalk_NV LocalTalkType parameter changes the LocalTalkType parameter of all _NVx sets. The _Pending and RAM sets are not affected.

Table 3-16 %EthernetTalk_NVx% communication parameter sets

Key	Type	Default - all cases	Details
EtherTalkType	*string*	LaserWriter	Parameters refer to the same nonvolatile storage. Changing this parameter changes EtherTalkType for all _NVx sets.
EtherTalkZone	*string*	8	≤ 31 characters. The characters : @ are not allowed.
EthernetAddress	*string*	unique	A read-only string of the form XX:XX:XX:XX:XX:XX, that is the Ethernet address of the printer.
Enabled	*boolean*	true	
Interpreter	*name*	PostScript	
On	*boolean*	true	

Table 3-17 %Parallel_NVx% communication parameter sets

Key	Type	Parallel_NV defaults	Parallel_NV2 defaults	Parallel_NV3 defaults	Parallel_NV4 defaults
Interpreter	*name*	PostScript	PostScript	LaserJet IIP	PostScript
On	*boolean*	true	true	true	true
Enabled	*boolean*	true	true	true	true
Protocol	*name*	Normal	Normal	Raw	Binary
OutputDevice	*string*	(%Serial%)	(%Serial%)	(%Serial%)	(%Serial%)

NOTE OutputDevice specifies which communications device is to be used for stdout and stderr. Legal values are %Serial%, %SerialB%, and so on.

Engine Device

The `%Engine%` device contains parameters that control the print engine itself. The LaserWriter Pro's `%Engine%` device contains the parameters listed in Table 3-18.

Table 3-18 `%Engine%` communication parameter sets

Key	Type	Default	Details
`Darkness`	*real*	`0.5`	`Darkness` controls the amount of toner applied to the paper. A value of 0.0 signifies the minimum darkness, and 1.0 signfies the maximum darkness. Values outside this range are not legal. The LaserWriter Pro printer supports 16 levels of darkness, so this parameter is quantized into 16 steps. This is done by taking the integer portion of `Darkness*15`. Therefore, a value of 0.0 is not distinguishable from 0.05, but it is distinguishable from 0.1. Changes in the `Darkness` parameter are not sent to the engine until there are no pages in the paper path, either feeding or being copied.
`Type`	*name*	`Parameters`	—

Resource Categories

In Level 2, PostScript objects, such as fonts, patterns, and filters can be managed as open-ended collections of resources grouped into categories. A resource is requested by resource category and name. If the resource does not reside in virtual memory, the resource management mechanism loads it from an external source, such as a disk, a ROM cartridge, or a network file server. The *PostScript Language Reference Manual* discusses named resources in detail.

Tables 3-19 through 3-21 list the factory-installed categories and resource instances. There are several groups of resources:

- New resources that are part of the regular resource may be added by the customer. These include such items as font and pattern resources (Table 3-19).
- Categories of implicit resources (Table 3-20) represent built-in capabilities of the interpreter. For example, the `FontType` category indicates that the interpreter understands the font formats for font types 0, 1, 3, 4, 5, and 42.
- Some resources define new resource categories (Table 3-21).

Table 3-19 Regular resource categories

Category name	Instances	Details
ColorRendering	DefaultColorRendering300 DefaultColorRendering600	
ColorSpace	No instances defined	
Encoding	ISOLatin1Encoding StandardEncoding	
Font	AvantGarde-Book AvantGarde-BookOblique AvantGarde-Demi AvantGarde-DemiOblique Bookman-Demi Bookman-DemiItalic Bookman-Light Bookman-LightItalic Courier Courier-Bold Courier-BoldOblique Courier-Oblique Emulatorfont Helvetica Helvetica-Bold Helvetica-BoldOblique Helvetica-Narrow Helvetica-Narrow-Bold Helvetica-Narrow-BoldOblique Helvetica-Narrow-Oblique Helvetica-Oblique NewCenturySchlbk-Bold NewCenturySchlbk-BoldItalic NewCenturySchlbk-Italic NewCenturySchlbk-Roman Palatino-Bold Palatino-BoldItalic Palatino-Italic Palatino-Roman Symbol Times-Bold Times-BoldItalic Times-Italic Times-Roman ZapfChancery-MediumItalic ZapfDingbats	
Form	No instances defined	

continued

Table 3-19 Regular resource categories (continued)

Category name	Instances	Details

`Halftone`	`150x0`
	`106x45`
	`85x56`
	`75x0`
	`53x45`
	`80x45`
		The instances listed name Type 1 halftone dictionaries, where the name is interpreted as `FrequencyxAngle`.
		`Frequency` and `Angle` are the corresponding keys in the Type 1 halftone dictionary. See the *PostScript Language Reference Manual* for further information.

`Pattern`	No instances defined	

`ProcSet`	`DiagnosticProcs`
	`SamplePages`
	`ActualValues`
		`DiagnosticProcs` contains two operators, `EngineHours` and `SendCommand`. `EngineHours` pushes an integer that estimates the number of hours the printer has been turned on since it left the factory. `SendCommand` takes an integer argument, sends it to the printer engine, and returns the integer result of that command.
		`SamplePages` contains two operators, `StartPage` and `TestPage`. `StartPage` rasterizes the start page and prints it. It takes no arguments and returns no results. `TestPage` prints out the configuration page.
		`ActualValues` contains one operator, `HWResolution`, which returns the resolution currently being used.

Table 3-20 Resources with implicit instances

Category name	Instances	Details
ColorRendering	1	
ColorSpaceFamily	CIEBasedA CIEBasedABC DeviceCMYK DeviceGray DeviceRGB Indexed Pattern Separation	
Emulator	LaserJetIIP SystemPatch hpcl	See Table 3-22. SystemPatch provides the mechanism for patching printer software that is specific to the LaserWriter Pro printer.
Filter	ASCII85Decode ASCII85Encode ASCIIHexDecode ASCIIHexEncode CCITTFaxDecode CCITTFaxEncode DCTEDecode DCTEEncode LZWDecode LZWEncode NullEncode RunLengthDecode RunLengthEncode SubFileDecode	
FMapType	2, 3, 4, 5, 6, 7, 8	
FontType	0, 1, 2, 4, 5, 42	These instances indicate the number of fonts the LaserWriter Pro printer is capable of rasterizing.
FormType	1	
HalftoneType	1, 2, 3, 4, 5	

continued

Table 3-20 Resources with implicit instances (continued)

Category name	Instances	Details
IODevice	%EtherTalk% %EtherTalk_NV% %EtherTalk_NV2% %EtherTalk_NV3% %EtherTalk_NV4% %EtherTalk_Pending%	The EtherTalk I/ODevices are not present in the LaserWriter Pro 600 printer.
	%LocalTalk% %LocalTalk_NV% %LocalTalk_NV2% %LocalTalk_NV3% %LocalTalk_NV4% %LocalTalk_Pending%	
	%Parallel% %Parallel_NV% %Parallel_NV2% %Parallel_NV3% %Parallel_NV4% %Parallel_Pending%	
	%Serial% %Serial_NV% %Serial_NV2% %Serial_NV3% %Serial_NV4% %Serial_Pending%	
	%SerialB% %SerialB_NV% %SerialB_NV2% %SerialB_NV3% %SerialB_NV4% %SerialB_Pending%	
	%Disk0% %Disk1% %Disk2% %Disk3% %Disk4% %Disk5% %Disk7%	The %diskx% IODevices are present only when a SCSI disk with the corresponding SCSI ID is attached to the printer. Note that %disk6% is reserved for the LaserWriter Pro printer.
	%Engine%	
PatternType	1	

Table 3-21 Resources used in defining new resource categories

Category name	Instances

`Category`	`Category`
	`ColorRendering`
	`ColorRenderingType`
	`ColorSpace`
	`ColorSpaceFamily`
	`Emulator`
	`Encoding`
	`FilterFMapType`
	`Font`
	`FontType`
	`Form`
	`FormType`
	`Generic`
	`Halftone`
	`HalftoneType`
	`ImageType`
	`IODevice`
	`Pattern`
	`PatternType`
	`ProcSet`

`Generic`	No instances defined

Emulator Parameters

Table 3-22 lists the emulator parameters specific to the LaserWriter Pro printer. The parameters apply to the Hewlett-Packard LaserJet IIP emulator. The emulator parameters specify the characteristics of fonts used in the LaserJet IIP emulation.

Table 3-22 Emulator parameters

Key	Type	Semantics
`FontFixed`	*boolean*	If `true`, this parameter requests a fixed pitch font, such as Courier. If `false`, it requests a proportional spaced font, such as Times®.
`FontHeight`	*real*	Specifies the desired font height in points. For fixed-pitch fonts, the pitch takes precedence over the height. For instance, if you specify Courier in 10 pitch, with 10-point height, you will get 12-point height, since that is the height of 10-pitch Courier.
`FontItalic`	*boolean*	If `true`, requests an italic or oblique font.
`FontPitch`	*real*	This parameter is used only if `FontFixed` is true. In this case, it is a real number specifying the number of characters per inch.
`FontSymbolSet`	*integer*	This parameter is used in mapping from 7-bit or 8-bit numbers to glyphs that appear on the page. The value of this parameter is the number associated with this field in a downloaded font, for example, `Roman8=277`.
`FontTypeface`	*integer*	The value of this parameter corresponds to the number that the LaserJet IIP has assigned to a particular font. The emulator uses this mapping and ignores fonts with other names.

Font	LaserJet IIP value
`Courier`	3
`Helvetica`	4
`Times`	5
`Letter Gothic`	6
`Prestige Elite`	8
`Orator`	10
`Optima`	17
`Garamond`	18
`Cooper Black`	19
`New Century Schoolbook`	23
`University Roman`	24

continued

Table 3-22 Emulator parameters (continued)

Key	Type	Semantics
`FontWeight`	*integer*	Specifies the weight or boldness of the desired font. As an option, PostScript contains a weight parameter in the `FontInfo` dictionary. The weight name is a string that maps to a font weight, as shown below: Weight name — LaserJet IIP value `/Thin` -3 `/Light` -3 `/Roman` 0 `/Medium` 0 `/Book` 0 `/Regular` 0 `/Demi` 2 `/Bold` 3 `/Heavy` 4 `/Black` 4 `/UltraBold` 4 `/ExtraBold` 4
`Landscape`	*boolean*	If `true`, the initial orientation of the page will be landscape rather than portrait.
`LinesPerInch`	*integer*	Specifies the default value for the vertical motion index. This determines the spacing between lines, and thus the number of lines per page.
`ManualFeed`	*boolean*	See the *PostScript Language Reference Manual*.
`MaxLJMemory`	*integer*	There are LaserJet IIP PCL escape sequences that permanently download fonts and macros. With the flexibility of PostScript Level 2 memory allocation, the LaserJet IIP emulator permanently acquires memory at the expense of other PostScript needs, such as virtual memory and font cache.

Compatibility Operators

The PostScript language has undergone a number of significant extensions. The language is designed to be a universal standard for device-independent page descriptions, but each PostScript language implementation supports features and capabilities particular to that implementation. Appendix D, "Compatibility Strategies," in the *PostScript Language Reference Manual*, presents guidelines for taking advantage of language extensions, while maintaining compatibility with PostScript interpreters.

Level 1 implementations provide a collection of device control and system parameter configuration operators and procedures, most of which are defined in the dictionary `statusdict`. The contents of `statusdict` are product dependent, although an attempt has been made to maintain a consistent specification for common features. It is the dictionary for product-specific operators and other definitions.

Device control and configuration of system parameters in PostScript Level 2 are accomplished in a standard way in the language through the device setup and interpreter parameter operators.

Level 1 PostScript language driver software frequently depends on `statusdict` operators that were present in PostScript Level 1 products. To maintain compatibility with these products, a collection of `statusdict` operators and procedures is included in each Level 2 implementation. Most of these functions are implemented as PostScript language procedures that call `setpagedevice` or appropriate Level 2 operators with appropriate arguments.

Adobe Systems recommends that you do not use the `statusdict` operators in PostScript Level 2 drivers, since the presence or absence of the operators depends on the product. Instead, you should use Level 2 standard operators.

The compatibility operators present in the LaserWriter Pro printer appear in three dictionaries: `statusdict`, `userdict`, and `systemdict`. Table 3-23 lists these operators by dictionary group.

Table 3-23 Compatibility operators

`statusdict`

```
a4tray
appletalktype
b5tray
buildtime
byteorder
checkpassword
defaultmultipurposepagesize
defaultpapertray
defaulttimeouts
diskonline
diskstatus
dostartpage
dosysstart
emulate
hardwareiomode
initializedisk
jobname
jobtimeout
legaltray
lettertray
manualfeed
manualfeedtimeout
margins
pagecount
pagestackorder
papersize
papertray
printername
product
ramsize
revision
realformatrevision
sccbatch
sccinteractive
setdefaultimeouts
setdefaultpapertray
setdefaultmultipurposepagesize
setdostartpage
setdosysstart
sethardwareiomode
setjobtimeout
setmargins
setpagestackorder
setpapertray
setprintername
setsccinteractive
setsoftwareiomode
softwareiomode
setuserdiskpercent
userdiskpercent
waittimeout
```

NOTE `sccinteractive` and `setsccinteractive` do not perform an operation.

continued

Table 3-23 Compatibility operators (continued)

userdict

a4	letter
a4small	lettersmall
b5	note
legal	

systemdict

devdismount	devstatus
devmount	devforall
devformat	

Setting System Parameters

This section describes the compatibility operators that set Level 2 system parameters. Table 3-1 also provides information about the system parameters.

buildtime

Syntax – `buildtime` *int*

Definition This operator returns an integer with the same value as the system parameter `BuildTime`.

Error `stackoverflow`

byteorder

Syntax – `byteorder` *bool*

Definition This operator returns a Boolean value with the same value as the system parameter `ByteOrder`.

Error `stackoverflow`

checkpassword

Syntax *string* | *int* `checkpassword` *bool*

Definition This operator checks whether *string* or *int* (*int* is converted to a *string*) is a valid password for either `SystemParamsPassword`, or `StartJobPassword`. If valid, `true` is returned. Otherwise, `false` is returned. If either password is not set, then `true` is returned. A returned value of `true` indicates that *string* or *int* is a valid argument to `startjob` and `exitserver`.

Errors `stackunderflow, typecheck`

defaultmultipurposetraysize

Syntax – `defaultmultipurposetraysize` *name bool*

Definition This operator returns the *name* and *bool* parameters used with `setdefaultmultipurposetraysize` to set the default multipurpose tray size. The standard for this operator is `/lettertrue`.

Error `stackoverflow`

defaultpapertray

Syntax – `defaultpapertray` *tray*

Definition This operator returns the default paper tray number set by `setpapertray`.

Error `stackoverflow`

defaulttimeouts

Syntax – `defaulttimeouts` *job manualfeed wait*

Definition This operator returns the following:

- system parameter `JobTimeout` for *job*
- system parameter `WaitTimeout` for *wait*
- page device parameter `ManualFeedTimout` for *manualfeed*

Error `stackoverflow`

dostartpage

Syntax – `dostartpage` *bool*

Definition This operator returns the value of the system parameter `DoStartPage`.

Error `stackoverflow`

dosysstart

Syntax – `dosysstart` *bool*

Definition This operator returns `true` only if the value of the system parameter `StartupMode` is `1`.

Error `stackoverflow`

emulate

Syntax *file name* `emulate` –

Definition This operator invokes one of the emulators. The *file* parameter is used as an input source for the emulation. For the LaserWriter Pro printer the file should always be the file that is returned by the `currentfile` operator. The *name* parameter selects which emulator to invoke. The acceptable name is `/hpcl`.

If a serial input channel is used as the emulation source, the binary protocol must be selected. Otherwise, a `rangecheck` error occurs.

This procedure enables host computers to switch between PostScript language interpretation and the emulator.

Errors `rangecheck, stackunderflow, typecheck, VMerror`

pagecount

Syntax – `pagecount` *int*

Definition This operator returns the value of the system parameter `PageCount`.

Error `stackoverflow`

papersize

Syntax – `papersize` *name bool*

Definition This operator returns the name of the operator that selects a tray containing paper of the current size. For example, if the current paper size is letter, this operator returns the value `/lettertray`. The value of *bool* is `true` if the page feeds short edge first, `false` if the page feeds long edge first. For the LaserWriter Pro printer, the value of *bool* is always true.

If there are two paper trays installed, and you execute the operator returned by papersize at a later time, it may not choose the same tray, if both trays have the same size paper installed.

Error `stackoverflow`

papertray

Syntax – `papertray` *integer*

Definition This operator returns the paper tray numbers, as set by the `setpapertray` operator. The standard value for `papertray` is the value of the `defaultpapertray` operator.

Error `stackoverflow`

printername

Syntax *string* `printername` *substring*

Definition This operator stores the value of the system parameter `PrinterName` in *string*, and it returns a string object designating the *substring* actually used.

Errors `rangecheck, stackunderflow, typecheck`

product

Syntax – `product` *string*

Definition This operator returns a *string* with the same value as the string `product` in `systemdict`.

Error `stackoverflow`

ramsize

Syntax – `ramsize` *int*

Definition This operator returns an integer with the same value as the system parameter `RamSize`.

Error `stackoverflow`

realformat

Syntax `– realformat` *string*

Definition This operator returns a *string* with the same value as the system parameter `RealFormat`.

Error `stackoverflow`

revision

Syntax – `revision` *int*

Definition This operator returns a *string* with the same value as the system parameter `Revision`.

Error `stackoverflow`

setdefaultmultipurposepapertraysize

Syntax *name bool* `setdefaultmultipurposetraysize` –

Definition This operator tells the interpreter what paper size is installed in the multipurpose tray. Because the printer cannot sense this information, operators that need to know paper size refer to the value stored by this operator.

This operator must be executed outside the server loop.

The *name* operand is the name of one of the standard device setup procedures: `/letter, /legal, /a4,` or `/a5`. The procedures `/lettersmall` and `a4small` are not allowed. The value of the pagetype operator controls whether the page is small or not.

The *bool* parameter is included for compatibility with other PostScript printers. It specifies whether the paper is to be fed long edge first or short edge first. For all paper sizes on the LaserWriter Pro printer, the value of *bool* must be `true`, which means the short edge is fed first.

Errors `invalidaccess, rangecheck, stackunderflow, typecheck`

setdefaulttimeouts

Syntax *job manualfeed wait* `setdefaultimeouts` –

Definition This operator returns the following:

- system parameter `JobTimeout` for *job*
- system parameter `WaitTimeout` for *wait*
- page device parameter `ManualFeedTimout` for *manualfeed*

Errors `invalidaccess, rangecheck, stackunderflow, typecheck`

setdostartpage

Syntax *bool* `setdostartpage` –

Definition This operator sets the system parameter `DoStartPage` to the value of *bool*.

This operator must be executed outside the server loop.

Error `invalidaccess, rangecheck, stackunderflow, typecheck`

setdosysstart

Syntax *bool* `setdosyststart` –

Definition This operator sets the system parameter, `StartupMode`, according to the value of *bool*. `StartupMode` is set to `1` if *bool* is `true`, and it is set to 0 if *bool* is `false`.

Errors `invalidaccess, stackunderflow, typecheck`

setpapertray

Syntax *integer* `setpapertray` –

Definition This operator sets the paper tray from which paper will be fed, and it sets the image area according to the size of paper in the tray, and the value of the `pagetype` operator. The integer argument must be:

0 for the main cassette
1 for the multipurpose tray
2 for the 500-sheet cassette
3 for the envelope feeder

Because this operator installs a new image area, it should be invoked before any marks are placed on the current page. If this operator is executed while an outstanding printer error exists, the interpreter waits until the error has been cleared before completing the execution of this operator. This is also true of operators such as `lettertray` and `legaltray`, because they execute `setpapertray`

Errors `rangecheck, stackunderflow, typecheck`

setprintername

Syntax *string* `setprintername` –

Definition This operator sets the system parameter, `PrinterName` to the value of *string*.

Errors `invalidaccess, limitcheck, stackunderflow, typecheck`

Setting Page Device Parameters

This section describes compatibility operators that set Level 2 page device parameters. Table 3-1 provides additional information about these operators.

margins

Syntax – `margins` *top left*

Definition This operator returns the *x* and *y* components of the page device `Margins` parameter as *left* and *top*, respectively.

Error `stackoverflow`

pagestackorder

Syntax – `pagestackorder` *bool*

Definition This operator returns the logical complement of the page device `OutputFaceUp` Boolean parameter

Error `stackoverflow`

setmargins

Syntax *top left* `setmargins` –

Definition This operator sets the page device `Margins` parameter to [left top].

Errors `rangecheck, stackunderflow, typecheck`

setpagestackorder

Syntax *bool* `setpagestackorder` –

Definition This operator sets the page device `OutputFaceUp` parameter to the logical complement of *bool*. For example, if *bool* is `true`, `OutputFaceUp` is set to `false`.

Errors `stackunderflow, typecheck`

Setting User Parameters

This section describes the compatibility operators that set Level 2 user parameters. Table 3-9 provides additional information about these operators.

jobname

Syntax – `jobname` *string*

Definition This operator is a string with the same value as the user parameter `JobName`. If you redefine either `jobname` or the user parameter `JobName`, the other is redefined to the same value.

Error `stackoverflow`

jobtimeout

Syntax `–` `jobtimeout` *int*

Definition This operator returns the value of the user parameter `JobTimeout`.

Error `stackoverflow`

setjobtimeout

Syntax *int* `setjobtimeout` –

Definition This operator sets the user parameter `JobTimeout` to the value of *int*.

Error `stackoverflow`

waittimeout

Syntax – `waittimeout` *int*

Definition This operator is an integer with the same value as the user parameter `WaitTimeout`. Redefining either `waittimeout` or the user parameter `WaitTimeout` redefines the other to the same value.

Error `stackoverflow`

Setting Device Parameters

This section describes the compatibility operators that set Level 2 device parameters.

appletalktype

Syntax – `appletalktype` *string*

Definition This operator is a string with the same value as the `LocalTalkType` device parameter in the `%LocalTalk%` parameter set.

Error `stackoverflow`

diskonline

Syntax – `diskonline` *bool*

Definition This operator returns `true` only if the writable disk device is registered as present and operational. The disk need not have an initialized PostScript file system.

Error `stackoverflow`

diskstatus

Syntax – `diskstatus` *freetotal*

Definition This operator returns the *total* number of pages free for all writable disk devices. A page is 1024 characters.

Error `stackoverflow`

initializedisk

Syntax pages action `initializedisk -`

Definition This operator initializes each writable disk, setting the disk device parameters as follows:

- `LogicalSize` to the value of *pages*
- `Initialize action` to *action* + 1

Error `invalidaccess, ioerror, rangecheck, stackunderflow, typecheck`

hardwareiomode

Syntax – `hardwareiomode` *int*

Definition This operator returns *int*, which indicates a current communications channel for which the corresponding device parameter `Enabled` Boolean value is `true`. Because multiple channels may be enabled, the smallest *int* is returned. The interpretation of *int* is as follows:

```
0   %Serial%
1   %Parallel%
2   %LocalTalk%
3   %SerialB%
```

Error `stackoverflow`

manualfeed

Syntax – `manualfeed` *bool*

Definition This operator is a Boolean that works in conjunction with the page device parameter `ManualFeed` to determine whether a page is to be fed manually. If either `manualfeed` or `ManualFeed` is `true` at the time of a `showpage` or `copypage`, than that page will be fed manually. Otherwise, the page will not be fed manually. The `manualfeed` compatibility operator is present in `statusdict` only if the page device parameter `ManualFeed` is defined for the product. The initial value of `manualfeed` when the printer is powered up is `false`.

Error `stackoverflow`

manualfeedtimeout

Syntax – `manualfeedtimeout` *int*

Definition This operator is an integer that works in conjunction with the page device parameter `ManualFeedTimeout` to determine the manual feed timeout for any given page. By default, `manualfeed`timeout is not defined in `statusdict`. In that case, the value of the page device parameter `ManualFeed`Timeout is used to determine the timeout value. If a job has defined `manualfeed`timeout to be an integer value in `statusdict`, then this value will be used instead of `ManualFeed`Timeout for the timeout value.

Error `stackoverflow`

sethardwareiomode

Syntax *int* `sethardwareiomode –`

Definition This operator opens specified channels for communication and closes other channels. The variable *int* specifies which communication channel(s) should be opened by setting the `On` and `Enabled` device parameters to true. All other channels will be explicitly closed by setting the `On` and `Enabled` parameters to `false`. The interpretation of *int* is as follows:

`0` Open `%Serial%` and `%SerialB%.` Close all others.
`1` Open `%Parallel%.` Close all others.
`2` Open `%LocalTalk%`. Close all others.
`3` Open `%Serial` and `%SerialB%`. Close all others.

Errors `invalidaccess, rangecheck, stackunderflow, typecheck`

setsoftwareiomode

Syntax *int* `setsoftwareiomode` –

Definition This operator sets the values of the interpreter and, if appropriate, the Protocol device parameters for the current communications device parameter set. The meaning of *int* is as follows:

`0` PostScript Normal
`1` Not defined
`2` Not defined
`4` Not defined
`5` LaserJet IIP Raw
`100` PostScriptBinary

Errors `invalidaccess, rangecheck, stackunderflow, typecheck`

setuserdiskpercent

Syntax *int* `setuserdiskpercent -`

Definition This operator pops *int* off the stack. It is not operative in the LaserWriter Pro printer.

Error `stackunderflow, rangecheck, typecheck`

softwareiomode

Syntax – `softwareiomode` *int*

Definition This operator returns *int,* which indicates the interpretation mode for the current communications device. See `setsoftwareiomode.`

Error `stackoverflow`

userdiskpercent

Syntax – `userdiskpercent` *int*

Definition This operator returns `0`. It is not operative in the LaserWriter Pro printer.

Error `stackoverflow`

Setting Serial Communication Parameters

The following compatibility operators set Level 2 serial communications parameters. Serial communications channel (SCC) operator encoding is described in the following section.

sccbatch

Syntax *channel* `sccbatch` *baud options*

Definition This operator returns the serial communications device parameter settings. The values are from one of the following two parameter sets:

- `%SerialB_NV%`, if *channel* equals 9,
- `%Serial_NV%`, if *channel* equals 25

baud and *options* affect the following device parameters:

- `Baud, StopBits`, and `FlowControl`
- `DataBits` and `Parity`
- `CheckParity`

See "`setsccbatch`" for further information.

The values for data bits and parity are determined by the bit positions defined in Tables 3-25 and 3-26. (See the next section, "SCC Operator Encoding.")

Baud, stop bits, and flow control are determined respectively by the corresponding settings for `Baud`, `StopBits`, and `FlowControl` device parameters.

Errors `rangecheck, stackoverflow, stackunderflow, typecheck`

sccinteractive

Syntax *channel* – `sccinteractive` *baud options*

Definition This operator pops the input argument off the stack, and pushes 0 0 onto the stack. It does not perform any operation in this application.

Errors `rangecheck, stackoverflow, stackunderflow, typecheck`

setsccbatch

Syntax *channel baud options* `setsccbatch` –

Definition This operator sets the communication device parameters for serial communications. It affects the following settings:

- `%SerialB_NV%`, if *channel* equals 9
- `%Serial_NV%`, if *channel* equals 25

baud and *options* affect the following device parameters:

- `Baud`, `StopBits`, and `FlowControl`, which are set according to the values for baud, stop bits, and flow control.
- `DataBits` and `Parity`, which are set based on the bit positions defined in Tables 3-25 and 3-26.

Error `stackoverflow`

setsccinteractive

Syntax *channel baud options* `setsccinteractive` –

Definition This operator pops the input argument off the stack, and pushes 0 0 onto the stack. It does not perform any operation in this application.

Errors `rangecheck`, `stackoverflow`, `stackunderflow`, `typecheck`

SCC Operator Encoding

The serial communications controller (SCC) operators use a 1-byte options argument that holds four SCC encoded parameters: stop bits, data bits, flow control, and parity. The argument is an integer parameter with values in the range 0–255. Table 3-24 lists the bit values for the compatibility operators options byte.

In PostScript Level 1, the data bits and parity bit interact in a nonorthogonal way, to produce a table of possible choices for data and parity. The choices include many commonly required methods of sending data. The Standard data bits setting provides compatibility with earlier versions of the PostScript Level 1 SCC operators. A standard setting could always be achieved with either a 7-bit or an 8-bit data setting.

In PostScript Level 2, there are parameters analogous to those given earlier for `%Serial%` and `%SerialB%` device parameter sets.

The mapping between Level 1 stop bits and flow control and Level 2 `StopBits` and `FlowControl`, respectively, is straightforward and obvious. It is not possible to provide such a one-to-one correspondence between the Level 1 notion of data bits and parity and the Level 2 parameters `DataBits` and `Parity`. Tables 3-25 and 3-26 show the conversions between PostScript Level 1 data bits and parity and Level 2 `DataBits` and `Parity`.

Table 3-24 SCC compatibility operators options byte values

Bit position	Value and function
Bit 7	**Stop bits**
	`0` 1 stop bit `1` 2 stop bits
Bits 6 and 5	**Data bits**
	`0` Standard `1` 7 bits `2` 8 bits
Bits 4–2	**Flow control**
	`0` XON/XOFF `1` DTR `2` ETX/ACK
Bits 1 and 0	**Parity**
	`0` Space `1` Odd `2` Even `3` Mark

Note

In going from `DataBits` and `Parity` to data bits and parity, standard parity is never used as it is in Level 1. ◆

Table 3-25 Optionsbyte-to-device parameters

Data bits and parity	DataBits and Parity
Level 1	**Level 2**
Standard space	7 bits Space
Standard mark	8 bits None
Standard odd	7 bits Odd
Standard even	7 bits Even
7 bits space	7 bits Space
7 bits mark	7 bits Mark
7 bits odd	7 bits Odd
7 bits even	7 bits Even
8 bits space	8 bits Space
8 bits mark	8 bits Mark
8 bits odd	8 bits Odd
8 bits even	8 bits Even

Table 3-26 Device parameters-to-options conversion

DataBits and Parity	Data bits and parity
Level 2	**Level 1**
7 bits None	7 bits mark
7 bits Space	7 bits space
7 bits Mark	7 bits mark
7 bits Odd	7 bits odd
7 bits Even	7 bits even
8 bits None	8 bits mark
8 bits Space	8 bits space
8 bits Mark	8 bits mark
8 bits Odd	8 bits odd
8 bits Even	8 bits even

The conversions shown in Tables 3-25 and 3-26 provide the best possible compatibility with PostScript Level 1 behavior. However, in several cases, no correct choice is possible. For example, in Level 1 there is no support for 7 data bits with no parity, where the total number of data and parity bits is 7. The Level 2 setting of 7 bits None is imperfectly mapped to 7 bits mark. Most serial hardware does not support 8 bit Mark or 8 bit Space, and therefore, these values are never generated in mapping from Level 1 to Level 2. In fact, in Level 1, 8 bits mark and 8 bits space actually provide the equivalent of the Level 2 8 bits None functionality.

Page Size Compatibility Operators

The page size operators are in the user dictionary, `userdict`. Each operator requests a specific paper size and imaging box. See Table 3-27 on page 72. The operators use the sizes indicated in the table as a page device `PageSize` parameter. All operators set `PageSizePolicy` to 7, which guarantees that the imaging area established is correct for the size requested, regardless of which paper tray is chosen.

The only error generated is `limitcheck`, which is occurs when there is not sufficient memory for the imaging area requested.

The `note` operator modifies the current page device settings by establishing an `ImagingBBox` parameter of [25 25 *width* minus 25 *height* minus 25] if the current PageSize parameter is [*width height*].

Table 3-27 Paper size compatibility operators

Operator	PageSize	ImagingBBox
a4	[595 842]	Null
a4small	[595 842]	[25 25 570 817]
b5	[516 729]	Null
legal	[612 1008]	Null
letter	[612 792]	Null
lettersmall	[612 792]	[25 25 587 767]

NOTE Units shown (595, etc.) are points. 1 point is 1/72 inch.

Paper Tray Compatibility Operators

The paper tray operators are in the status dictionary, `statusdict`. Each operator requests a tray containing a specific paper size. The only difference between the operations is the size of paper requested. The `PageSize` parameter requested is the same as for the corresponding page size operator, and the `ImagingBBox` parameter requested is always null. These operators use the specified size as a page device `PageSize` parameter. All the operators set the `PageSizePolicy` parameter to 0, which guarantees that a `rangecheck` error is generated if a tray containing the requested paper size is not found. In addition, a `limitcheck` error can occur if there is not sufficient memory for the imaging area requested. Table 3-28 lists the paper tray compatibility operators.

Table 3-28 Paper tray compatibility operators

Operator	PageSize	ImagingBBox
a4tray	[595 842]	Null
b5tray	[516 729]	Null
legaltray	[612 1008]	Null
lettertray	[612 792]	Null

CHAPTER 4

TrueType Fonts

The LaserWriter Pro printer has built-in TrueType font-scaling software. This chapter describes the behavior of the TrueType downloadable PostScript font format as it applies to the LaserWriter Pro printer, which is described as a Type A Device. In defining requirements for the LaserWriter Pro printer, this chapter also provides information on other LaserWriter printers and PostScript devices that do not have the built-in TrueType font scaler. These printers are referred to in this chapter as Type B and Type C devices.

Note
In this context, the term "device" is used to refer to a laser printer. ◆

The TrueType font format is efficient and produces high-quality print. Designed to be as universally standardized as possible, it is nevertheless constrained by PostScript implementations in older printers. To support current users, the TrueType implementation is designed to run efficiently on the large installed base of LaserWriter printers. Future printers may be optimized to take advantage of any enhancements to TrueType, and the format is designed to allow these enhancements to be incorporated easily and dynamically.

The TrueType format places all PostScript devices in one of three classes:

- Class A devices are those with a TrueType scaler embedded in the PostScript device. The LaserWriter Pro printer is a Class A devices.
- Class B devices are those with TrueType font-scaling code downloaded to the device separately from the font itself.
- Class C devices do not have TrueType available in any form, for example, third-party PostScript compatible printers.

The downloadable TrueType font format can be used on any class device. However, performance and quality characteristics may vary widely. Because their inherent inability to use the TrueType enhancements, Class C devices show performance and quality degradations. All third-party 68000-based Adobe PostScript printers are treated as Class B devices, since they can handle downloaded assembly code.

TrueType Font Format

The TrueType PostScript format incorporates three components:

- TrueType code
- a small set of procedures, referred as the patch, which are used only during TrueType processing
- TrueType font for font definition

TrueType Code

The TrueType font-scaling code is partitioned into three pieces and sent to the device on demand for each document that uses an `'sfnt'` (scalable font) resource. The code is encrypted using the Adobe encryption mechanism and depends on the existence of the `eexec` and `cexec` operators.

Because of the way the code is encrypted, the downloadable TrueType code can be used only on Class B devices and is discarded on all other classes of devices.

You should note the following three requirements:

- If `eexec` and `cexec` are defined on a printer, their implementation must be compatible with Adobe PostScript.
- Since the TrueType code is encrypted 68000-family code, it will not run on a printer that is not based on the 68000 family of processors.
- In the case of the LaserWriter 7.0 driver, the amount of available virtual memory (VM) on the printer when the driver first encounters an `'sfnt'` resource in a document is a factor in deciding whether the TrueType code is downloaded. If the printer has less than 120,000 bytes of VM available, the TrueType code will not be downloaded.

Because it is relatively large, TrueType code is partitioned into three physical pieces in the LaserWriter 7.0 driver. When this code is downloaded, four new operators are defined in PostScript. A PostScript dictionary, `TrueDict`, is created in which those four operators (as well as some version information) are stored. One operator is used to initialize TrueType for each new `'sfnt'` resource, and the others are used within the `BuildChar` procedure in the font.

The TrueType code renders characters in either bitmap or PostScript path form. The path form is invoked only when a character path is required during rendering via `charpath` or an outline (`PaintType 2`) style. If the bitmap size for a character exceeds 10,000 bytes (which is roughly the memory needed for a 100-point character at 300 dpi), the scaler is asked to band the bitmap, and the character is printed in bands. Future drivers or other applications may download different operators. If so, these operators will have different names if their semantics differ from those defined by the LaserWriter 7.0 driver. The entries in the font dictionary for a Class B printer (defined in "TrueType Font Dictionary Entries" later in this chapter) remain the same.

Patch

The patch, redefines the PostScript `charpath` operator. It signals whether characters are to be rendered using the PostScript path or via a bitmap. The patched `charpath` simply sets a global flag to signal that `charpath` is in effect. This flag is then examined when characters are being rendered. If the flag is `true`, the characters are always constructed using a PostScript path rather than a bitmap.

Note
As with the TrueType code, the redefinition of PostScript operators is ignored on Class A and Class C devices. ◆

TrueType Font Definition

The third component is the actual TrueType font. The font has the minimum but essential parts of a normal PostScript font: a font dictionary containing a font type, font matrix, font bounding box, and an encoding vector. In addition, the font should contain a font name, paint type, stroke width (for outline styles), the TrueType `'sfnt'` (font data) as it exists in the `'sfnt'` resource on the host, and for Class B devices only, the TrueType state information and a `BuildChar` procedure. The major bulk of the font is the `'sfnt'` font data.

The `FontType` entries for the three different classes of devices are as follows:

- For Class A devices, such as the LaserWriter Pro printer, the TrueType font is `42`, and the `BuildChar` procedure is therefore implicit. For example, based on the `FontType` entry, the font-rendering machinery will know where to find, and how to execute, the font data. The Type 42 font format is described in "TrueType Font Dictionary Entries," later in this chapter.
- For Class B devices, the `FontType` entry is `3`, indicating that it is a user-defined font as understood by PostScript.
- For Class C devices, the `FontType` entry is `1`.

The TrueType font has a `UniqueID` entry, a 24-bit number derived from the checksum in the `'sfnt'` header. The presence of `UniqueID` in the font makes the PostScript font cache operate more efficiently and avoids rerendering characters across jobs. Two `UniqueID` entries are given to the font, one for the hinted font (Class A or Class B) and the other for the unhinted font (Class C). The `UniqueID` for a Class C font is further restricted to be in the range between `4000000` and `4999999`. This range is reserved as an open range by the Type 1 specification. Only one ID is used on any particular printer. Providing two `UniqueID` entries helps to avoid a situation where even though the printer is capable of rendering hinted characters, it receives unhinted characters that were stored in the font cache by a previous job. This could happen if the previous job did not have enough memory to download TrueType code.

The `'sfnt'` array can contain any number of data strings, no single one of which can be larger than 65,536 bytes. The `'sfnt'`array is divided into the required number of pieces at arbitrary table or glyph boundaries within the `'sfnt'`. To guarantee word alignment of the data, there is always one extra byte at the end of each string in the `'sfnt'` array. The strings are internally linked or combined at run time to simulate a continuous string of data. There is no loss in performance speed because of this restriction in string size. The `'sfnt'` data exists in two forms: the actual `'sfnt'` data straight out of the `'sfnt'` resource and the unfolded glyph data (as Type 1 `CharStrings`) for Class C devices. The printer ignores the data it does not need so that the font size as it is stored on the printer is not increased.

The entries in a TrueType font dictionary for a Class A or Class B printer are listed and described in "TrueType Font Dictionary Entries," later in this chapter. The font dictionary for a Class C printer follows the Adobe Systems, Incorporated, Type 1 font format specification.

Device Operation

There are three classes of printers, known as Class A devices, Class B devices, and Class C devices. The subject of this developer note, LaserWriter Pro printer, is a Class A device. This section describes how TrueType fonts are handled on Class A printers. It also describes how TrueType fonts are handled on Class B, and Class C devices, to provide definition by comparison for the LaserWriter Pro printer.

Class A Devices

On Class A devices, such as the LaserWriter Pro printer, the downloadable TrueType code is extraneous data and is discarded when the printer determines it is not needed. Any low-level patches are similarly discarded. A system-level operator or flag on Class A devices is invoked to determine whether the TrueType code and patches are needed. The entries and behavior of Type 42 fonts are intended to be very similar to the LaserWriter built-in PostScript fonts (Type 1). Like Type 1 fonts, Type 42 fonts have an implicit `BuildChar` procedure, as opposed to the explicit `BuildChar` entry for Type 3 fonts.

When a character bitmap is needed from a Type 42 font, the character cache is checked first. If the bitmap is not cached, the character code is used as an index into the font's encoding array, returning a character name. This name is used as an index into the `CharStrings` dictionary, which is a required entry in the font dictionary. The value corresponding to the character name is an integer, representing the glyph index in the `'sfnt'` resource. The `'sfnt'` resource has a table for mapping character codes to glyph indexes, but PostScript allows an extra level of indirection in this mapping to reencode fonts. The glyph index and the `'sfnt'` data itself from the scalable fonts entry are used to rasterize the character.

Adobe built-in font formats (Type 1) have a capability called charstring procedures, that allows user-defined characters to be added to the `CharStrings` dictionary. If the value of the `CharStrings` entry corresponding to a character name is an executable array (procedure), the following steps take place:

1. The `systemdict` dictionary and the font dictionary are pushed onto the dictionary stack.
2. The character code is pushed onto the operand stack.
3. The procedure is executed.
4. The `systemdict` dictionary and the font dictionary are popped from the dictionary stack.

The semantics of the procedure are almost identical to those of the Type 3 font format `BuildChar` procedure, except that in the Type 3 procedure, nothing is pushed onto the dictionary stack and the font dictionary is passed onto the operand stack. The contents of the procedure must follow the same rules as the Type 3 font format `BuildChar`

procedure with respect to `setcharwidth`, `setcachedevice`, and so on. This behavior has existed in all PostScript font formats, but it has only been documented as part of Level 2. This behavior is part of Type 42 `BuildChar`.

Class B Devices

Class B devices provide the primary motivation and design center for the TrueType font format in its current configuration. The small low-level patches are downloaded to Class B devices to assist the TrueType code in its operation and to provide the necessary hooks into the PostScript code. The definitions provided here are downloaded in the `userdict` at the beginning of every job.

As stated earlier, TrueType code is downloaded on demand. If the `eexec` and `cexec` operators are provided, their implementation must be compatible with Adobe PostScript, or TrueType will not print on that implementation. Depending on the level of compatibility, a PostScript error may be raised, or the printer may crash.

Since the content of the font on a Class A device may differ from that on a Class B or Class C device, executing a PostScript `forall` operation within the context of a TrueType font dictionary produces different results on different machines. This should not be of significant concern, since the main contents and required definitions are the same.

Class C Devices

One crucial assumption made by the TrueType code is that all Class C devices support the Adobe Type 1 font format. TrueType cannot be printed on a PostScript-compatible printer that cannot interpret the Type 1 font format. An alternative solution (not documented in this note) would allow TrueType characters to be printed on any PostScript-compatible device by downloading a Type 3 (user-defined) font with a `BuildChar` procedure. This would convert TrueType data into cubic Bezier* curves to be filled by PostScript. This is not an efficient solution. With Adobe Systems, Incorporated making the Type 1 font format public, more PostScript-compatible printers will support the Type 1 format and therefore print TrueType.

Downloading TrueType Fonts to Disk

TrueType fonts may be downloaded to printers equipped with hard disk drives that store fonts. The entire font may be stored and used just like any other PostScript font, or, depending on the intelligence of the font-downloading utility, the font can be stripped of unnecessary items that will not be used on a particular class of printer. To facilitate the

* A Bezier curve is a freehand curve, named after the mathematician who first described it. It is similar to a hand-drawn curve, and you may use it when you need a continuous curve. A filled curve is one in which the enclosed area of the curve is shaded.

operation of intelligent font downloaders, there are several conventions that must be used for the textual definition of the font.

The first line in the PostScript font file is

`%!PS-TrueTypeFont-`*sfntFormat-fontRevision-commentFormat*

where *sfntFormat* is the version number of the `'sfnt'` format (from the `'sfnt'` header), *fontRevision* is the font manufacturer's revision of the font (also from the `'sfnt'` header), and *commentFormat* refers to this version of the commenting convention. An intelligent downloading utility can use this line to identify TrueType fonts on a printer's hard disk.

If this line appears as the first line of a font program, the following conventions must be strictly followed or an error may occur.

- The token `/sfnts` must be followed by the token `[` and either `<` or `(`, depending on the encoding of the binary `'sfnt'` data. There may be whitespace and/or control characters (`<CR>, <LF>, <TAB>`) between these tokens.
- All of the strings defined in the `'sfnt'` array of a font program must use the same encoding (ASCII or ASCIIHex). Different font programs may use different encodings.
- The characters representing the `'sfnt'` data must follow a sequence of *N* characters of data followed by *M* characters of whitespace, repeating until the string's data is exhausted. The last sequence of character data may be less than *N* characters long. The last tokens in each string should be *M* characters of whitespace, followed by the character(s) for 1 pad byte of data, followed immediately by the string terminator (either > or), depending on the data encoding. There may be whitespace and/or control characters between string definitions. The values of *M* and *N* must be constants for a font program. Different font programs may use different values of *M* and *N*. The value for *N* must be between 0 and 2048, inclusive.

Note

The 1024 bytes of binary `'sfnt'` data require 2048 characters to represent them in the ASCII Hex encoding. ◆

- The last string definition in the `'sfnt'` array must be followed by the token] and the token `def`. There may be whitespace or control characters, or a combination of whitespace and control characters, between these tokens.
- The PostScript code for different classes of printers must be bracketed with `begin` and `end` comments. The `end` comments are already used by the `checkload` and `fcheckload` procedures when discarding sections of PostScript code that are not appropriate for a given class of printer. The `begin` comments are for the font downloader, which does not have a PostScript interpreter to do the discarding automatically. Because of a limitation in the readline operator in early versions of the PostScript interpreter, the `end` comments must be bracketed by only linefeed (ASCII 10) characters.

Table 4-1 lists the comments that delineate sections of code and indicates the classes of printers for which they are required. <SP> indicates the space character. You must enter a space at the beginning of each `end` comment line.

Table 4-1 Section code comments

Comments	Description
`%beginsfnt` `<SP>%endsfnt`	These comments bracket the creation of the common entries in the font dictionary for Class A and Class B devices. It may be discarded on Class C devices.
`%beginsfntBC` `<SP>%endsfntBC`	These comments bracket the definition of entries specific to Class B (`TrueState` and `BuildChar`) in the TrueType font dictionary. They may be discarded on Class A and Class C devices.
`%beginsfntdef` `<SP>%endsfntdef`	These comments bracket the call to `definefont`, which registers the font dictionary for Class A and Class B devices. They may be discarded on Class C devices.
`%beginType1` `<SP>%endType1`	These comments bracket the definition of the Type 1 font dictionary for Class C devices. They may be discarded on Class A and Class B devices.

TrueType Font Dictionary Entries

In the Macintosh system software, TrueType fonts are represented as a resource, called `'sfnt'` for scalable font. In PostScript interpreters, fonts are represented as dictionaries with certain special key-value pairs. One of these entries, `FontType`, identifies the font format and tells the PostScript font mechanism how to interpret the remaining entries. The `FontType` entry for TrueType fonts on Class A devices is `42`. This section describes the remaining entries and their semantics for Type 42 font dictionaries.

The following tables represent possible entries in a TrueType font dictionary for Class A or Class B devices. Class C devices use the Type 1 font format, as documented by the *PostScript Language Reference Manual*. Certain entries are required either only for Class A devices or only for Class B. Other entries have different values, depending on the type of device. Still others are optional and are not used by the font-rendering code itself.

A valid Type 42 font dictionary must have certain key-value pairs. Table 4-2 lists the entries common to all PostScript fonts. Table 4-3 lists Type 1 specific entries. Some of these are supported in exactly the same way as in Type 1, and others are ignored by Type 42 fonts. Table 4-4 lists the entries specific to Type 42 font dictionaries. Table 4-5 lists the entries in the optional FontInfo dictionary and indicates where the corresponding information is found in the `'sfnt'` format.

Table 4-2 Type 42 key-value pairs common to all PostScript font dictionaries

Key	Type	Semantics
`Encoding`	*array*	Required. An array of 256 names that maps character codes (integers) to character names. Note that Apple TrueType fonts have an encoding vector different from the StandardEncoding used by Type 1 fonts. Conventional value: derived from information in the `'sfnt'` post table.
`FontBBox`	*array*	Required. An array of four numbers in the character coordinate system giving lower-left x, lower-left y, upper-right x, and upper-right y of the font bounding box. To ensure compatibility with certain versions of the LaserWriter driver, this array should have the executable attribute.
`FontInfo`	*dictionary*	Optional. This entry is for information only. `FontInfo` is not used by the PostScript interpreter. See Table 4-5 for the entries that can be included in this dictionary.
`FontMatrix`	*array*	Required. Transformation matrix for transforming the character coordinate system into the user coordinate system. TrueType fonts maintain this value internally (for example, Apple TrueType fonts use a 2048-unit coordinate system), so the PostScript coordinate system transformation is the identity matrix. Value for Type 42 fonts: `[100100]` Note that certain PostScript programs (for example, program 16 in the *PostScript Language Tutorial and Cookbook*) incorrectly assume that all PostScript fonts have a 1000-unit coordinate system. These programs may exhibit incorrect behavior when used with Type 42 fonts.
`FontName`	*name*	Optional. This entry is for information only. `FontName` is not used by the PostScript interpreter. Conventional value: PostScript name from the `'sfnt'` name table
`FontType`	*integer*	Required. Indicates where the information for the character descriptions is found and how it is represented. Value for TrueType fonts: `42`
`LanguageLevel`	*integer*	Optional. This integer indicates the minimum language level required for correct behavior of the font. This entry is for information only. `LanguageLevel` is not used by the PostScript interpreter. Default value: `1`
`UniqueID`	*integer*	Optional. An integer in the range `0` to `16777215 (224 – 1)` that uniquely identifies this font for the purposes of caching character bitmaps and metrics. Conventional value: the lower 24 bits of the `'sfnt'` checksum.

continued

Table 4-2 Type 42 key-value pairs common to all PostScript font dictionaries (continued)

Key	Type	Semantics
`WMode`	*integer*	Optional. Indicates which of two sets of metrics is used when characters are shown from this font. If this entry (or the WMode entry of the root font from which this font is a descendant) has the value 1, then this font must have a CDevProc entry (see Table 4-3). See Section 5.9 of the *PostScript Language Reference Manual*, for information about composite fonts. Default value: 0
`XUID`	*array*	Optional. An array of integers that uniquely identifies this font or any variant of it for the purposes of caching character bitmaps and metrics.

Table 4-3 Entries for Type 1 specific font dictionaries

Key	Type	Semantics
`CharStrings`	*dictionary*	Required. Associates character names (keys) with glyph IDs (integers). These IDs access data in the `'sfnt'` format. Every Type 42 font must have a `notdef` entry (usually with glyph ID 0). The value can also be an executable PostScript procedure. See Section 5.6.3 of the *PostScript Language Reference Manual*.
`CDevProc`	*procedure*	Optional. A procedure that derives global changes algorithmically from a font's metrics. If this font (or the root font for which this font is a descendant) has a `WMode` of 1, this entry is required. See Section 5.6.2 of the *PostScript Language Reference Manual*, for the semantics of this procedure.
`Metrics`	*dictionary*	Ignored. Adding a `Metrics` entry will have no effect on a Type 42 font.
`Metrics2`	*dictionary*	Ignored. Adding a `Metrics2` entry will have no effect on a Type 42 font.
`PaintType`	*integer*	Required. A code indicating how the characters of the font are to be painted: `0` The character outlines are filled. `2` The outlines (designed to be filled) are stroked. TrueType fonts are ordinarily created with a `PaintType` of 0. A program desiring to convert it to a stroked outline font can copy the font dictionary, change the `PaintType` from 0 to 2, add a `StrokeWidth` entry, and define a new font using this dictionary. Note that if `PaintType 0` is chosen, the TrueType scan converter is used to render the character. If `PaintType2` is chosen, the grid-fitted TrueType outline is converted to PostScript path segments and the PostScript scan converter strokes the path.
`Private`	*dictionary*	Ignored. Type 42 fonts do not require a `Private` dictionary.

continued

Table 4-3 Entries for Type 1 specific font dictionaries (continued)

Key	Type	Semantics
`StrokeWidth`	*number*	Optional. Indicates the stroke width (in units of the character coordinate system) for stroked outline fonts (`PaintType` 2). This field is not initially present in filled font descriptions. It must be added when creating a stroked font from an existing font. Note that certain PostScript programs (for example, program 16 in the *PostScript Language Tutorial and Cookbook*) incorrectly assume that all PostScript fonts have a 1000-unit coordinate system. These programs may exhibit incorrect behavior when used with Type 42 fonts.

Table 4-4 Font dictionary entries specific to Type 42 fonts

Key	Type	Semantics
`sfnts`	*array*	Required. An array of PostScript string objects that contains the font description in the `'sfnt'` format. Because PostScript strings can be no more than 65,535 bytes long, `'sfnt'` descriptions that are longer than 65,535 bytes must be broken into separate strings. The `'sfnt'` data should be divided at both a longword and a table boundary. If a single table exceeds 64K bytes, it should be divided at the nearest longword and glyph boundary. For compatibility with certain versions of the LaserWriter driver, each string in the `'sfnt'` array must contain a single pad byte at the end.

Table 4-5 Optional entries for `FontInfo` dictionary

Key	Type	Semantics
`FamilyName`	*string*	Name for a group of fonts that are stylistic variants of a single design. All fonts that are members of such a group should have exactly the same `FamilyName`. Conventional value: font family name from the `'sfnt'` name table.
`FullName`	*string*	Unique name for an individual font. Conventional value: full font name from the `'sfnt'` name table.
`isFixedPitch`	*boolean*	If `true`, indicates that the font is a fixed-pitch (monospaced) font. Conventional value: `isFixedPitch` Boolean value from the `'sfnt'` post table.
`ItalicAngle`	*number*	Angle in degrees counterclockwise from the vertical of the dominant vertical strokes in the font. Conventional value: italic angle from the `'sfnt'` post table.

continued

Table 4-5 Optional entries for `FontInfo` dictionary (continued)

Key	Type	Semantics
`Notice`	*string*	Trademark or copyright notice, if applicable. Conventional value: copyright notice from the `'sfnt'` name table.
`UnderlinePosition`	*number*	Recommended distance from the baseline for positioning underlining strokes. This number is the y coordinate (in character space) of the center of the stroke. Conventional value: underline position from the `'sfnt'` post table.
`UnderlineThickness`	*number*	Recommended stroke width for underlining, in units of the character coordinate system. Conventional value: underline thickness from the `'sfnt'` post table.
`version`	*string*	Version number of the font program. Conventional value: Version string from the `'sfnt'` name table (not the version entry in the `'sfnt'` post table).
`Weight`	*string*	Name for the weight, or boldness, attribute of a font. Conventional value: font subfamily name from the `'sfnt'` name table.

TrueType Fonts