Developer Note

Personal LaserWriter 320 Printer

Developer Note

12/8/93 Developer Press

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

Simultaneously published in the United States and Canada.

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

About This Note

The Personal LaserWriter 320 printer is a new member of the Apple Computer LaserWriter printer family. This developer note describes the features and capabilities of the printer, 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 and printer terminology commonly referred to in PostScript programming documentation. 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 Personal LaserWriter 320 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, explains visual cues and conventions used in the note, and lists other books to which you can refer.

What This Note Contains

This note consists of four chapters and an index.

- Chapter 1, "Introduction to the Personal LaserWriter 320 Printer," describes the hardware features of the Personal LaserWriter 320 printer, including the built-in communication port and the printer's paper-handling capabilities.
- Chapter 2, "PostScript Software," provides general information about the PostScript Level 2 programming language, the Personal LaserWriter 320 driver, the utility program, and page types.
- Chapter 3, "PostScript Level 1 Compatibility Operators," explains how to set the different software parameters using the Personal LaserWriter 320 printer's compatibility operators. They enable the Personal LaserWriter 320 printer, which uses PostScript Level 2, to maintain compatibility with printers that use PostScript Level 1.
- Chapter 4, "Communication Channels," describes the software support provided for the Personal LaserWriter 320 printer's communication channel.

Conventions and Abbreviations

This developer note uses the following typographical conventions and abbreviations.

Typographical Conventions

Computer-language text—any text that is literally the same as it appears in computer input or output—appears in `Courier` font.

Certain terms used in this note may appear in different typographical formats—for example, `BuildTime` and `buildtime`. `BuildTime` is the format used for the system parameter, and `buildtime` is the format used for the operator `buildtime`.

Note
A note like this contains information that is interesting but not essential for an understanding of the text. ◆

IMPORTANT
A note like this contains important information that you should read before proceeding. ▲

▲ **WARNING**
A note like this directs your attention to something that could cause damage or result in a loss of data. ▲

Standard Abbreviations

When unusual abbreviations appear in this developer note, the corresponding terms are also spelled out. Standard units of measure and other widely used abbreviations are not spelled out. The following abbreviations are used in this note:

AMD	Advanced Micro Devices
dpi	dots per inch
DRAM	dynamic RAM
EEPROM	electronically erasable and programmable ROM
EPROM	electronically programmable ROM
I/O	input/output
KB	kilobyte
MB	megabyte
MHz	megahertz

PDL	page-description language
ppm	pages per minute
RAM	random-access memory
ROM	read-only memory
VM	virtual memory

Other Reference Material

This developer note assumes that you are familiar with printer technology and know how to operate and program Apple LaserWriter printers. Additional information is available in the following publications:

- The owner's guide that is shipped with every Apple 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, and set up an external hard disk for fonts. It also provides basic troubleshooting information.
- *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 supplement to this manual, the *PostScript Language Reference Manual Supplement,* or simply the *Supplement*, is available from Adobe Systems, Inc.
- *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.
- *PostScript Language Program Design*, published by Addison-Wesley, is written for programmers who want to take advantage of the PostScript programming 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 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

Introduction to the Personal LaserWriter 320 Printer

The Personal LaserWriter 320 printer is a mainstream network laser printer designed for the small-business market. Replacing the Apple Personal LaserWriter NTR and LaserWriter Select 310, it supports PostScript™ Level 2 functions and produces up to four printed pages per minute. The Personal LaserWriter 320 printer has improved imaging capabilities. It supports 300-dpi resolution, with FinePrint text at 300 dpi, as well as line art anti-aliasing. FinePrint technology gives the printer the ability to print dots of different widths, producing crisper text and graphics. Anti-aliasing smooths the jagged edges of both characters and lines, producing an effective resolution much greater than 300 dpi.

The Personal LaserWriter 320 printer is intended to work with the LocalTalk interface. The LocalTalk port may be connected to a network of one or more Macintosh computers, or any other computers that can support LocalTalk. The printer is available in 110-volt and 220-volt versions.

This chapter describes

- hardware features of the printer
- LocalTalk port
- status lights
- memory capabilities
- basic operation
- page types
- paper-handling capabilities

Features of the Printer

The Personal LaserWriter 320 printer supports the entire PostScript Level 2 language as specified in the second edition of the *PostScript Language Reference Manual*. In addition, the Personal LaserWriter 320 printer has features, capabilities, and operating modes not present in other PostScript language printers. You may access these additional facilities by executing special PostScript operators that exist only in the Personal LaserWriter 320 printer's PostScript interpreter.

Table 1-1 lists functional features of the Personal LaserWriter 320 printer.

Table 1-1 Personal LaserWriter 320 printer features

Feature	Specifications
Printing speed	4 ppm
PostScript processing	equilevant to the Personal LaserWriter NTR
Imaging	User-selectable resolution and imaging features: ■ 300-dpi bilevel text and images (no FinePrint) ■ selectable 300-dpi resolution ■ selectable FinePrint text anti-aliasing ■ PhotoGrade printing (with addition of 6 MB DRAM expansion card)
Processor	RISC processor (AMD 29205, 15 MHz)
Paper handling	Standard output: 25-page face-down tray Standard inputs: manual feeder with 100-page universal cassette For more information, see sections “Page Types” and “Paper Handling,” at the end of this chapter
ROM	4 MB of onboard masked ROM
DRAM	2 MB onboard; expandable to 4 MB or 8 MB using either a 2 MB memory card or 6 MB memory card
EEPROM	1 KB of onboard electronically erasable and programmable memory
EPROM	2 MB of electronically programmable memory mounted on a plug-in secondary logic board
Interface ports	Mini-DIN 8-pin serial port for LocalTalk
Fonts	35 PostScript Type I fonts
PDL (page-description language)	Adobe PostScript Level 2, or compatible
Support for n-up printing	Allows 1, 2, or 4 pages to be printed on one sheet of paper

Communication Port for LocalTalk

The Personal LaserWriter 320 printer supports one communication port, an 8-pin mini-DIN serial port for LocalTalk. Figure 1-1 shows the connector pin designations for the 8-pin connector, and Table 1-2 lists the pin functions.

Figure 1-1 The 8-pin mini-DIN connector for LocalTalk

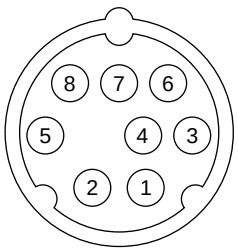

Table 1-2 Signal descriptions for LocalTalk

Pin number	Signal name	Description
1, 2, 7	NC	Not connected
3	/TXD	Transmit data (inverted)
4	GND	Signal ground
5	/RXD	Receive data (inverted)
6	TXD	Transmit data
8	RXD	Receive data

Status Lights

The Personal LaserWriter 320 printer has three colored lights on the left side of the printer. These lights indicate what the printer is doing. Table 1-3 describes the functions of the lights.

Table 1-3 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 because the printer cover is open.
	Flashing	The printer is warming up, or it is processing data for the next print job.
Paper out (amber)	On	Paper tray is empty, or it 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 (amber)	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, there may be an entry jam for a multipurpose tray job. If there is no jam, or if clearing the jam does not fix the problem, the printer requires service.

Memory Capabilities

The standard configuration of the Personal LaserWriter 320 printer comes with 2 MB of DRAM mounted on the main circuit board, or controller, which is sufficient for FinePrint printing. The printer also accommodates the addition of DRAM expansion cards that provide either 4 MB or 8 MB of printer DRAM, depending on the card installed.

Two DRAM expansion cards are available, a 2 MB card and a 6 MB card. A 2 MB card installed in the printer memory expansion slot provides a total of 4 MB of DRAM. A 6 MB card installed in the printer memory expansion slot provides a total of 8 MB of DRAM.

When configured with 8 MB of DRAM, the Personal LaserWriter 320 supports PhotoGrade printing for enhanced grayscale output.

Basic Operation

The Personal LaserWriter 320 printer operates in two modes: batch and interactive.

The printer's main function is to execute the PostScript language programs sent to it from a computer. In normal operation, the printer cycles endlessly through the following sequence of steps:

1. It sets up a clean initial execution environment (virtual memory) for the PostScript language program. This is what is known as setting up a job.
2. It executes the job by interpreting the standard input data stream, which is received on the LocalTalk port. Data stream sensing determines the start and end of PostScript print jobs.
3. When the printer encounters an end-of-job indicator (this may be a character or a packet) or when an error occurs, the printer cleans up after the job and restores the virtual memory to its initial state in preparation for the next job. Fonts downloaded outside the server loop persist in memory. These fonts may be released back to the memory pool as required. Fonts downloaded inside the server loop do not persist across jobs.

The main object of this process is to produce printed pages. However, a program may change some permanent parameters in the printer itself or may perform some computation that causes results to be sent back to the host computer rather than causing hard copy to be printed.

Batch Mode

Batch mode is the normal way of operating the Personal LaserWriter 320 printer. In this mode, the printer operates as a printing device for a computer.

A batch-mode job consists of executing a single file containing a PostScript language program. When an end-of-job character is reached, or the PostScript language terminates, the job is finished. In this mode, the only data transmitted from the Personal LaserWriter 320 printer to the host is generated by the PostScript language printer operator or by errors. The printer provides no echoing, editing, or other user amenities.

Interactive Mode

You can use the Personal LaserWriter 320 printer as a personal computer and control it directly by means of a terminal or other input device. This way of using the printer is known as interactive mode, and it allows you to experiment with the PostScript language.

In interactive mode, a job consists of a long dialogue in which you issue a PostScript language statement and the server executes the statement and prompts you for the next one. The state of the PostScript interpreter's virtual memory persists until you explicitly end the job. While you are entering a statement, the printer echoes characters and provides you with limited means for making corrections.

Page Types

The page size (the area in which printed output may appear) is constrained by

- the physical size of the paper (paper size)
- the margins required by the printing engine
- the amount of memory available for the full-page frame buffer

Table 1-4 lists the page types and sizes supported by the Personal LaserWriter 320 printer.

See the sections "Page Size Compatibility Operators" and "Paper Tray Compatibility Operators," in Chapter 3, for further information.

Table 1-4 Available page types

Name	Paper size in inches	Page size in inches	Description
`a4`	8.27 × 11.69	8.00 × 11.42	Standard page type for European A4-size paper
`a4small`	8.27 × 11.69	7.57 × 11.00	Smaller version of `a4`
`b5`	7.17 × 10.12	6.83 × 9.89	Standard page type for Japanese B5-size paper
`c5`	6.38 × 9.02	6.19 × 8.85	Standard page type for the C5-size envelope
`com10`	4.13 × 9.5	3.73 × 9.33	Standard page type for the COM10-size envelope
`dl`	4.33 × 8.66	4.16 × 8.51	Standard page type for the DL-size envelope
`executivesize`	7.25 × 10.5	6.93 × 10.26	Standard page type for executive-size paper
`legal`	8.5 × l4	8.21 × 13.71	Standard page type for legal-size paper. This legal page size applies to printers configured with either 4 MB or 8 MB of RAM.
`legal`	8.5 × l4	6.93 × 10.26	Standard page type for legal-size paper. This legal page size applies to printers configured with 2 MB of RAM.
`letter`	8.5 × 11	8.21 × 10.69	Standard page type for letter-size paper
`lettersmall`	8.5 × 11	7.89 × 10.30	Smaller version of letter size
`monarch`	3.87 × 7.5	3.73 × 7.34	Standard page type for monarch-size envelope

NOTE The margins required in all cases are 0.2 inches on each side and at the top and bottom. All images may be centered either horizontally or vertically, with the exception of `b5`, which must be centered horizontally.

Paper Handling

The Personal LaserWriter 320 printer offers a variety of paper-handling features. The printer comes with:

- an integral single-sheet feeder. You use it to manually feed single sheets of paper that are different from what is already in the paper cassette. For example, you can use the manual single-sheet feeder to insert envelopes, heavier paper, and paper of nonstandard sizes.
- a universal 100-sheet cassette that holds U.S. letter-size paper, A4-size paper, U.S. legal-size paper, and executive-size paper

CHAPTER 2

PostScript Software

This chapter describes the Personal LaserWriter 320 software. It includes

- an overview of the programming language, interpreter, driver, utility program, and page types
- a detailed description of the software parameters that enable you to set up and configure the Personal LaserWriter 320 printer, including page device parameters, product strings, interpreter parameters, and resource categories

Software Overview

This section provides an overview of the PostScript programming language, the PostScript interpreter, the printer driver, the printer utility program, and the page and envelope types supported by the printer.

Adobe PostScript Programming Language

The Personal LaserWriter 320 printer executes descriptions written in the PostScript language. The version of the PostScript language used 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 Personal LaserWriter 320 printer. You should use this note in conjunction with the *PostScript Language Reference Manual*, second edition.

PostScript Interpreter

You may access the special features of the Personal LaserWriter 320 printer by executing PostScript operators that exist only in this printer's interpreter. The PostScript Interpreter version at the time of printing is 2013.112.

The special operators are intended for use by interactive users, by programmers of host software that carries out user requests, or by users who may want to configure the Personal LaserWriter 320 printer in nonstandard ways. Normally page descriptions should not refer to the special operators, since doing so impairs portability.

Printer Driver

The Personal LaserWriter 320 printer driver and Print Manager provide a general interface to the Personal LaserWriter 320 printer. The interface should meet the needs of most Macintosh applications.

The printer driver

- provides full support for PostScript Level 2
- supports a universal paper tray and an envelope feeder

- allows you to configure the driver according to your printer configuration
- presents error messages if they are reported back by the printer: for example, printer jam status, paper out
- supports both TrueType and Type 1 fonts
- is compatible with version 7 of the Macintosh LaserWriter driver
- provides support for n-up printing: this is a new feature offered by version 8.0 of the Macintosh LaserWriter driver that allows you to print one, two, or four logical pages on a single sheet of paper

Printer Utility Program

The Personal LaserWriter 320 printer utility program, which is shipped with each printer, allows you to control and configure the printer. Using the printer utility program, you can perform the following types of functions:

- set printer parameters, such as printer name, start page mode, and so forth
- set printer density
- add or remove fonts and display or print a list of available fonts
- set page parameters and get the count of pages printed by the printer
- send PostScript files to the printer
- restart the printer
- set imaging options, such as FinePrint and PhotoGrade
- set the default paper size

Page and Envelope Types

The page and envelope types supported by the Personal LaserWriter 320 printer are listed in Table 1-4 on page 7.

The Personal LaserWriter 320 does not sense what the default paper tray is. The user must select the paper currently installed in the paper cassette.

If a job requires a particular paper size, it should invoke the following PostScript commands to select the appropriate paper size for the job:

```
<< /PageSize [x y] /InputAttributes << 0 <</PageSize [x y] >> >>
>> setpagedevice
```

Refer to Table 2-1 on page 12 for the *x* and *y* values of the paper sizes supported by the Personal LaserWriter 320. The paper size change will be in effect only for the duration of the job.

IMPORTANT

If the page size currently selected does not match the paper size installed in the cassette, it is possible for the printer to print on the drum outside of the page boundary. In this situation, undesirable imaging artifacts may persist on up to 10 pages. ▲

Table 2-1 Paper size and corresponding paper size name

Paper size	Paper size name
`[612 792]`	Letter
`[612 1008]`	Legal
`[595 842]`	A4
`[516 729]`	B5
`[522 756]`	Executive

Device Setup

The PostScript language facilities set up the raster output device (printer) to fulfill the processing requirements of the page description. The `setpagedevice` operator performs the following device setup functions:

- It specifies processing requirements, such as making multiple copies.
- It selects optional printer features, such as the proper input tray, paper size, and image area.
- It establishes device-dependent rendering parameters needed to produce output.
- It specifies 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: that is, 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 so that the printer can produce the required results. This may cause further modification of the page device state.

The Personal LaserWriter 320 printer uses the Level 2 implementation, which provides device control operators defined in the special dictionary `statusdict`.

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*, second edition.

Page Device Parameters

This section describes the page device parameters present in the Personal LaserWriter 320 printer. Refer to Section 4.11.3 of the *PostScript Language Reference Manual*, second edition, for supplemental information on parameter semantics. Table 2-2 lists the page device parameters and their defaults and provides additional technical information.

Table 2-2 Page device parameters

Key	Type	Default	Description
`BeginPage`	*procedure*	`{pop}`	This parameter is executed at the beginning of each page: at the end of `setpagedevice`, at the end of `showpage` or `copypage`, and during any operation that reinstates a page device different from the existing one.
`EndPage`	*procedure*	`{exch pop 2 ne}`	This parameter is executed at the end of each page. End of page occurs at the beginning of each `showpage` or `copypage` and when the current page device is about to be replaced by a different page device.
`ExitJamRecovery`	*Boolean*	`false`	If this parameter is `true`, pages that jam in the exit path are reprinted. If it is `false` (jam recovery disabled), pages that jam are not reprinted. In this case, performance may be improved because it is possible to overlap more page processing. Value persists across power cycles.
`HWResolution`	*array*	`[300 300]`	This parameter controls the resolution of the output. It is used in conjunction with the `Policies` dictionary (described later in this table) and the amount of available memory in the printer to determine if compression will be attempted on the frame buffer and at which resolution the frame buffer will print.
`ImagingBBox`	*array* or *null*	`null`	This parameter is an optional bounding box. If not `null`, the value is an array of four numbers in the default user coordinate system stating lower-left *x*, lower-left *y*, upper-right *x*, and upper-right *y* of the page image bounding box. In some configurations, this may be set to a default to reduce the imaging area allowed to less than a full page.
`InputAttributes`	*dictionary*	Depends on configuration	This parameter contains an entry for each source of input media available for use by the printer. The values *x* and *y* depend upon which paper tray is installed. 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. There are values of matching tolerance for the `PageSize` parameter, described later in this table.

continued

Table 2-2 Page device parameters (continued)

Key	Type	Default	Description
`Install`	*procedure*		This procedure installs values in the graphics state during each invocation of `setpagedevice`, which calls this procedure after setting up the device and installing it as the current device in the graphics state but before executing the implicit `erasepage` and `initgraphics`. The default install procedure is: `{currentpagedevice dup /PreRenderingEnhanceDetails get /ActualPreRenderingEnhance get {/PreRenderingEnhanceDetails get /DefaultHalftone get /Halftone findresource} {pop /53x45 /Halftone findresource} ifelse sethalftone {} settransfer false setstrokeadjust /DefaultColorRendering /ColorRendering findresource setcolorrendering}`
`ManualFeed`	*Boolean*	`false`	This parameter determines whether the input medium (paper) is to be drawn from the manual or the automatic feeder. It is `true` for manual feeding and `false` for automatic feeding.
`ManualFeedTimeout`	*integer*	`40`	This parameter specifies the number of seconds the printer will wait for a page to be fed manually before generating a timeout error. The default is 60 seconds. If the value is set to 0, there is no timeout, and the printer waits infinitely. Value persists across power cycles.
`Margins`	*array*	`[0 0]`	This parameter is an array of two numbers that relocates the page image on the media by *x* units in the direction of the *x* coordinate, and *y* units in the direction of the *y* coordinate. The *x* and *y* values are expressed as 1/300s of an inch. Value persists across power cycles.
`MediaColor`	*string* or *null*	`null`	This parameter specifies the color of the input media.
`MediaType`	*string* or *null*	`null`	This parameter specifies the type of media: paper, transparency, and so on.
`MediaWeight`	*number* or *null*	`null`	This parameter specifies the weight of the media.
`NumCopies`	*integer* or *null*	`null`	If this parameter is not `null`, it specifies the number of copies to produce. The value applies to each individual page or to the entire document, depending on the setting of `Collate`. If `NumCopies` is `null`, `showpage` and `copypage` should consult the value of `#copies` in the current dictionary stack each time they are executed.
`OutputFaceUp`	*Boolean*	`false`	This parameter determines whether the printed pages are output face up or face down in the output tray. If the value is `false`, the pages are output face up. If it is `true`, they are output face down. Value persists across power cycles.

continued

Table 2-2 Page device parameters (continued)

Key	Type	Default	Description
`OutputPage`	*Boolean*	`true`	If this parameter is `true`, pages are printed normally and output into the output tray. If it is `false`, no pages are actually printed. However, all other processing is done as if the pages were to be printed, including rasterizing to a frame buffer. In this case, the time required to process a page includes everything except the time spent waiting for the marking engine. In addition, rasterization occurs synchronously with the execution of `showpage` instead of being overlapped with the execution of subsequent pages. This function is used to measure the complete cost of executing a page.
`PageSize`	*array*	Depends on configuration	This parameter defines the overall page size that was assumed during generation of the page description. `PageSize` is an array of two numbers `[width height]`, which specify the overall size of the page, including borders. Matching tolerance is five default user space units in either dimension. Landscape mode ([792 612]) is also valid.

`Policies`	*dictionary*		This dictionary contains feature-policy pairs that specify what `setpagedevice` should do when a feature request cannot be satisfied. The default procedure is
			`<</PolicyNotFound 1`
			`/PageSize \\0\\\`
			`/PolicyReport {pop}>>`

`PostRendering Enhance`	*Boolean*	`true`	If this parameter is `true`, product-specific image enhancements are enabled. These enhancements are made after the page is rasterized in memory. Value persists across power cycles.
`PostRendering EnhanceDetails`	*dictionary*	Hardware dependent	This dictionary describes product-specific details related to the post-rendering image enhancement. Refer to "Details Dictionary," later in this chapter, for further information.
`PreRenderingEnhance`	*Boolean*	`true`	If the value for this parameter is `true`, product-specific image enhancements are enabled. These enhancements are made before the image is rasterized into memory. `PreRenderingEnhance` in the page device dictionary is treated as a hint rather than an assertion. If there is not sufficient memory to create an enhanced frame buffer of the requested size, it is treated as an unsatisfied request to be handled by the `Policies` dictionary. Value persists across power cycles.
`PreRendering EnhanceDetails`	*dictionary*	Hardware dependent	This dictionary describes product-specific details related to prerendering image enhancement. Refer to "Details Dictionary," later in this chapter, for further information.

NOTE All the terms in the first column are one word. They may be split onto multiple lines because of column width restrictions.

Table 2-3 lists the different paper sizes.

Table 2-3 Paper sizes

Name	Paper size
A4	[595 842]
B5	[516 729]
C5 envelope	[459 649]
COM10 envelope	[297 684]
DL envelope	[312 624]
Executive	[522 756]
Legal	[612 1008]
Letter	[612 792]
Monarch envelope	[279 540]

NOTE Page size is indicated by an array of two numbers ([595 842], and so on) that indicate width and height. The units are equivalent to 1/72 of an inch.

The Personal LaserWriter 320 has only one input paper tray. Table 2-4 lists the paper tray slot number and corresponding input source.

Table 2-4 Paper tray slot number and input source

Slot number	Input source
0	Cassette (100 sheets)

Details Dictionary

Certain page device features have many variables that decide how the features function. These variables are different on different products. The feature is enabled or disabled by a primary page device entry. However, the exact way in which the feature functions is decided by secondary entries in a `Details` dictionary page device entry. This means an application that is not knowledgeable about the details of the feature can enable and disable the feature, while more sophisticated utilities configure the details separately. Section 2.1 of the *PostScript Language Reference Manual Supplement* provides more information on this subject.

Product Strings

The Personal LaserWriter 320 printer's strings contain characters that provide information about the printer and the printer software. Table 2-5 lists values assigned to the Personal LaserWriter 320 product strings.

Table 2-5 Product string values

String name	Type	Value	Definition
`languagelevel`	*integer*	2	Level of the PostScript language
`product`	*string*	Personal LaserWriter 320	Printer name
`revision`	*integer*	1	Current revision level of the printer
`serialnumber`	*integer*	Depends on printer	Serial number of the printer
`version`	*string*	2013.112	Version of the PostScript language

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 Personal LaserWriter 320 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. There are two types of device parameters for the Personal LaserWriter 320, including communications and engine.

Each parameter 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 can find information on parameter semantics in the *PostScript Language Reference Manual*, second edition.

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. Using `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 2-6 lists the user parameters present in the Personal LaserWriter 320 printer.

Table 2-6 User parameters in the Personal LaserWriter 320 printer

Key	Type	Default	Description
`AccurateScreens`	*Boolean*	`false`	An optional parameter. If the value is `true`, the parameter invokes a special halftone algorithm that is extremely precise but requires a lot of computation.
`JobName`	*string*	`()`	Establishes *string* as the name of the current job. It should contain no more than 32 characters.
`JobTimeout`	*integer*	`0`	Sets the number of seconds a job is allowed to be executed before it is aborted and a `timeout` error is generated. It may be any number larger than 0. If you set this parameter to 0, timeout is disabled.
`MaxDictStack`	*integer*	`530`	Determines the maximum number of elements in the dictionary stack. It may be set to 0 or any number larger than 0.
`MaxExecStack`	*integer*	`10015`	Determines the maximum number of elements in the execution stack. It may be set to 0 or any number larger than 0.
`MaxFontItem`	*integer*	`12500`	Determines the maximum number of bytes occupied by the pixel array of a single character in the font cache. It may be set to 0 or any number larger than 0.
`MaxFormItem`	*integer*	`100000`	Determines the number of bytes occupied by a single cached character. It may be set to 0 or any number larger than 0.
`MaxLocalVM`	*integer*	`2147483647`	Determines the maximum number of bytes occupied by values in local virtual memory. It may be set to 0 or any number larger than 0.

continued

Table 2-6 User parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Description
`MaxOpStack`	*integer*	`100000`	Determines the maximum number of elements in the operand stack. It may be set to 0 or any number larger than 0.
`MaxPatternItem`	*integer*	`20000`	Determines the maximum number of bytes occupied by a single cached pattern. It may be set to 0 or any number larger than 0.
`MaxScreenItem`	*integer*	`12000`	Determines the maximum number of bytes occupied by a single halftone screen. It may be set to 0 or any number larger than 0. The initial value is 3000 bytes per MB of installed RAM, up to a maximum of 12,000 bytes.
`MaxUPathItem`	*integer*	`5000`	Determines the maximum number of bytes occupied by a single cached user path. It may be set to 0 or any number larger than 0.
`MinFontCompress`	*integer*	`1250`	Sets the threshold at which a cached character is stored in compressed form instead of as a full pixel array. It may be set to 0 or any number larger than 0.
`VMReclaim`	*integer*	`0`	Enables or disables local garbage collection: `0` enables automatic collection `-1` disables it for local VM `-2` disables it for both local and global VM
`VMThreshold`	*integer*	`40000`	This is the frequency of garbage collection. It is triggered whenever the number of bytes indicated by the setting has been allocated. It may be set to 0 or any number larger than 0.
`WaitTimeout`	*integer*	`40`	Indicates the current wait timeout, which is the number of seconds the interpreter waits to receive additional characters from the host before it aborts the current job by executing a `timeout` error. It may be set to 0 or any number larger than 0.

System Parameters

System parameters alter the overall configuration of the printer. You can set system parameters using the `setsystemparams` operator and read them using the `currentsystemparams` operator. You must use a password to change system parameters. System parameters are not subject to `save` and `restore`. Their values persist across jobs and may persist across power cycles. Table 2-7 lists the system parameters present in the Personal LaserWriter 320 printer.

Note

For further information about parameters listed in Table 2-7, refer to the *PostScript Language Reference Manual Supplement*, Section 3.4 and Section 3.9. ◆

Table 2-7 System parameters in the Personal LaserWriter 320 printer

Key	Type	Default	Details
`BuildTime`	*integer*	Actual date the interpreter was built	This read-only parameter is a time stamp that identifies the date the PostScript interpreter was built.
`ByteOrder`	*Boolean*	`false`	Determines the order of multiple-byte numbers in binary-encoded tokens: `false` indicates high-order byte first, `true` indicates low-order byte first.
`CurDisplayList`	*integer*	`0`	This read-only parameter identifies amount of RAM currently occupied by the display list.
`CurFontCache`	*integer*	`0`	This read-only parameter identifies amount of RAM currently occupied by the font cache.
`CurFormCache`	*integer*	`0`	This read-only parameter identifies amount of RAM currently occupied by the form cache.
`CurInputDevice`	*string*	`(%LocalTalk%)`	This read-only parameter indicates the name of the communications device that corresponds to the current input file for the PostScript language program currently being executed.
`CurOutlineCache`	*integer*	`0`	This read-only parameter identifies the amount of RAM currently occupied by the outline cache.
`CurOutputDevice`	*string*	`(%LocalTalk%)`	This read-only parameter indicates the name of the communications device that corresponds to the current output file for the PostScript language program currently being executed.
`CurPatternCache`	*integer*	`0`	This read-only parameter identifies the amount of RAM currently occupied by the pattern cache and indicates the name of the communications device that corresponds to the current input file for the PostScript language program currently being executed.

continued

Table 2-7 System parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Details
`CurScreenStorage`	*integer*	`0`	This read-only parameter identifies the amount of RAM currently occupied by screen storage and indicates the name of the communications device that corresponds to the current input file for the PostScript language program currently being executed.
`CurSourceList`	*integer*	`0`	This read-only parameter indicates the number of bytes currently occupied by source lists. It may be set to 0 or any number larger than 0.
`CurUPathCache`	*integer*	`0`	This read-only parameter indicates the number of bytes currently occupied by the U path cache. It may be set to 0 or any number larger than 0.
`DoStartPage`	*Boolean*	`true`	Indicates whether or not the start page should print during system initialization. The start page prints if the value is `true`. Value is persistent across power cycles.
`FactoryDefaults`	*Boolean*	`false`	This parameter is generally `false`. However, if you set it to `true` and immediately turn off the printer, all nonvolatile parameters will revert to the factory default values the next time the printer is turned on.
`FatalErrorAddress`	*integer*	`0`	This integer is the hardware address of the last call to the fatal error handler. A nonzero value for this parameter indicates that a fatal system error has occurred earlier.
`FontResourceDir`	*string*	`(fonts/)`	Controls the location of external fonts, which are resources in PostScript Level 2.
`GenericResourceDir`	*string*	`(Resource/)`	Controls the location of external resources for the `Generic` category and all other categories based upon it.

continued

Table 2-7 System parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Details
`GenericResourcePathSep`	*string*	`(/)`	Used in conjunction with `GenericResourceDir` to control the location of external resources for the `Generic` category and all other categories based upon it. With `GenericResourceDir` as `(Resource/)` and `GenericResourcePathSep` as `(/)`, the `AdobeLogo` resource of the `Pattern` category would be in `Resource/Pattern/AdobeLogo`.
`JobTimeout`	*integer*	`0`	Indicates the value in seconds to which the user parameter `JobTimeout` will be initialized at the beginning of each job. It may be set to 0 or any number larger than 0.
`LicenseID`	*string*	`(LN-000-001)`	Contains the Adobe™-assigned license identification. The value is unique to each printer. Any string of non-null characters is legal.
`MaxDisplayList`	*integer*	Function of RAM size	Indicates the maximum number of bytes occupied by display lists, excluding those held in caches. Initial value is 4 percent of installed RAM. This number is recomputed when the RAM configuration changes. It may be set to 0 or any number larger than 0.
`MaxFontCache`	*integer*	Function of RAM size	Indicates the maximum number of bytes occupied by the font cache. Initial value is based on the amount of RAM installed. It is 167,772 bytes for 4 MB of RAM. Otherwise, it is 10 percent of installed RAM. This number is recomputed when the RAM configuration changes.
`MaxFormCache`	*integer*	`100000`	Indicates the maximum number of bytes occupied by the form cache. It may be set to 0 or any number larger than 0.

continued

Table 2-7 System parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Details
`MaxImageBuffer`	*integer*	`65536`	Indicates the maximum number of bytes that can be used for a single image buffer. The image buffer holds an internal data representation for sampled image source data. The interpreter may round the value down .If the value requested is out of range,
`MaxOutlineCache`	*integer*	`65536`	Indicates the maximum number of bytes occupied by cached character outlines (`CharStrings`) for fonts whose definitions are kept on disk instead of in VM. It may be set to 0 or any number larger than 0.
`MaxPatternCache`	*integer*	`100000`	Indicates the maximum number of bytes occupied by the pattern cache. It may be set to 0 or any number larger than 0.
`MaxRasterMemory`	*integer*	`0`	Indicates the largest amount of memory, in bytes, 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. The implementation ignores values that are too small and guarantees that an `A4small`, `lettersmall`, or `B5` size frame buffer can be allocated. It may be set to 0 or any number larger than 0.
`MaxScreenStorage`	*integer*	Function of RAM size	Indicates the maximum number of bytes occupied by all active half-tone screens. 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. It may be set to 0 or any number larger than 0.
`MaxSourceList`	*integer*	`24576`	Indicates the maximum number of bytes that can be used by source lists. It may be set to 0 or any number larger than 0.
`MaxUPathCache`	*integer*	`300000`	Indicates the maximum number of bytes occupied by the user path. It may be set to 0 or any number larger than 0.

continued

Table 2-7 System parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Details
`PageCount`	*integer*	`0`	This read-only parameter indicates how many pages have been successfully printed since manufacture.
`PrinterName`	*string*	`(Personal LaserWriter 320)`	Establishes *string* as the current name of the printer. You may set this parameter to any string of 32 or fewer characters. The colon (`:`) and the at symbol (`@`) are not allowed.
`RamSize`	*integer*	Function of RAM size	This read-only parameter indicates in bytes the amount of installed RAM available to the printer. It may be set to 0 or any number larger than 0.
`RealFormat`	*string*	`(IEEE)`	Provides native representation of real numbers in binary-encoded tokens.
`Revision`	*integer*	`1`	This read-only parameter designates the current revision level of the ROM in which the interpreter is running.
`StartJobPassword`	*string*	`0`	This write-only password authorizes the use of the `startjob` operator. Any string of 32 or fewer characters may be used.
`StartupMode`	*integer*	`0`	Controls whether the system start file or some other startup procedure should be executed during system initialization. If the value is 0, there are no special startup procedures. Other values may be used that are product specific, and they result in product-dependent startup procedures.
`SystemParams Password`	*string*	`0`	This write-only password authorizes the use of the `setsystemparams` and `setdevparams` operators. Any string of 32 or fewer characters may be used. Value is persistent across power cycles.
`ValidNV`	*Boolean*	`true`	Indicates whether nonvolatile memory is currently used to store persistent parameters. This is a read-only parameter.

continued

Table 2-7 System parameters in the Personal LaserWriter 320 printer (continued)

Key	Type	Default	Details
`WaitTimeout`	*integer*	`40`	Indicates the value in seconds to which the user parameter `WaitTimeout` will be initialized at the beginning of each job. It may be set to 0 or any number larger than 0. A value of 0 indicates an infinite wait period.

NOTE All the terms in the first column are one word. They may be split onto multiple lines because of column width restrictions.

Device Parameters

Each PostScript interpreter supports a collection of input/output storage devices, such as communication channels, disks, and cartridges. You may set device parameters using the `setdevparams` operator, and you may read them using the `currentdevparams` operator. Like system parameters, device parameters require a password, are global to the PostScript environment, and have similar persistence characteristics, and some of them can be stored in nonvolatile storage.

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 communications device (`%Serial%`, `%LocalTalk%`, and so on). Some device parameters correspond to a software entity such as a language emulator.

Note

Even if two printers are using the same I/O storage device, the parameters in the set may be different, because the hardware support for that device is different. ◆

Communication Device Parameters

The Personal LaserWriter 320 printer has an 8-pin serial connector configured to use only the LocalTalk protocol. It supports the `%LocalTalk%` channel.

Section 3.5.2 in the *PostScript Language Reference Manual Supplement* also provides further information on the LocalTalk parameter set.

Table 2-8 lists the factory defaults settings for %LocalTalk%, %LocalTalk%_NV, and %LocalTalk_Pending%.

Table 2-8 Parameters for %LocalTalk%, %LocalTalk_NV%, and %LocalTalk_Pending%

Key	Type	Default	Description
Enabled	*Boolean*	true	Indicates whether data arriving at the printer should be scheduled for execution. If it is true, data is executed. If it is false, data is not executed.
HasNames	*Boolean*	false	This read-only constant indicates whether the printer supports named files. If the printer is not mounted, or if Type is /Communications, this value is false.
Interpreter	*name*	/PostScript	Indicates the type of executable job represented by the arriving data.
LocalTalkType	*string*	(LaserWriter)	Represents the Type piece of the LocalTalk entity name. It is set to the name of the printer type. In the case of the Personal LaserWriter 320 printer, the type is LaserWriter.
NodeID	*integer*	0	This read-only constant represents the local network address of the printer. Legal addresses are values between 128 and 254. A value of 0 indicates that the address has not yet been set.
On	*Boolean*	true	Indicates whether or not the printer driver for the communications device is turned on and able to receive and send data. If this value is false, data sent to the printer is lost.
Type	*name*	/Communications	This read-only constant indicates the general category of device represented by the parameter set.

NOTE All values, with the exception of Type, persist across cycles and restarts.

Engine Device Parameters

The %Engine% device contains parameters that control the print engine itself. The Personal LaserWriter 320 printer's %Engine% device contains the parameters listed in Table 2-9.

Table 2-9 %Engine% communication parameters

Key	Type	Default	Details
Darkness	*real*	0.484375	Controls the amount of toner applied to the paper. A value of 0.0 signifies the minimum darkness, and a value of 1.0 signifies the maximum darkness. Values outside this range are not legal. The Personal LaserWriter 320 printer supports 16 levels of darkness, so this parameter is divided into 16 steps. Multiply the integer portion of Darkness by 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. This value persists across power cycles and restarts.
Type	*name*	/Parameters	This read-only constant always returns a value of /Parameters.

Resource Categories

In PostScript language 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*, second edition, discusses named resources in detail.

There are several groups of resources:

- New resources in the regular resource categories can be added. These include such items as font and pattern resources (see Table 2-10).
- Categories of implicit resources represent built-in capabilities of the Personal LaserWriter 320 interpreter. For example, the FormType category indicates that the interpreter understands Type 1 only (see Table 2-11).
- Some resources are used to define new categories (see Table 2-12).

Most of the instances listed in the following tables are described in the *PostScript Language Reference Manual*, second edition, or the *PostScript Language Reference Manual Supplement*.

Table 2-10 lists the new resources in regular resource categories.

Table 2-10 Regular resource categories

Category name	Instances
Font	AvantGarde-Book AvantGarde-BookOblique AvantGarde-Demi AvantGarde-DemiOblique Bookman-Demi Bookman-DemiItalic Bookman-Light Bookman-LightItalic Courier Courier-Bold Courier-BoldOblique Courier-Oblique 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-Medium Italic ZapfDingbats
Encoding	ISOLatin1Encoding StandardEncoding
Form	No instances defined
Pattern	No instances defined

continued

Table 2-10 Regular resource categories (continued)

Category name	Instances
ProcSet	DiagnosticProcs SamplePages Test The Personal LaserWriter 320 printer has two predefined ProcSet instances, one that is a dictionary that contains named test procedures (the SendCommand procedure) called DiagnosticProcs, and another called SamplePages, which is a dictionary containing named start page procedures (StartPage procedure).
Halftone	DefaultHalftone 150 × 0 106 × 45 83 × 56 71 × 45 53 × 45 80 × 45

Table 2-11 lists categories of implicit resources with the built-in capabilities of the Personal LaserWriter 320 interpreter.

Table 2-11 Resources with implicit instances

Category name	Instances
ColorRenderingType	1
ColorRendering	DefaultColorRendering
ColorSpaceFamily	CIEBasedA CIEBasedABC DeviceCMYK DeviceGray DeviceRGB Indexed Pattern Separation
ColorSpace	No instances defined

continued

Table 2-11 Resources with implicit instances (continued)

Category name	Instances
Filter	ASCII85Decode ASCIIHexDecode ASCII85Encode ASCIIHexEncode CCITTFaxDecode CCITTFaxEncode DCTDecode DCTEncode LZWDecode LZWEncode NullEncode RunLengthDecode RunLengthEncode SubFileDecode
FMaptype	2, 3, 4, 5, 6, 7, 8
FormType	1
FontType	0, 1, 3, 4, 5, 6, 42 The integers 0, 1, 3, 4, 5, and 6 are the instances supported for the Personal LaserWriter 320 printer. Type 42, a TrueType font with the PostScript wrapper, is also supported.
HalftoneType	1, 2, 3, 4, 5, 6
ImageType	1
IODevice	%Engine% %LocalTalk% %LocalTalk_NV% %LocalTalk_Pending% %rom% %Serial% %Serial_NV% %Serial_Pending%
PatternType	1

Table 2-12 defines resources used to define new categories.

Table 2-12 Resources to define new categories

Category	Instances
Category	Category ColorRendering ColorRenderingType ColorSpace ColorSpaceFamily Emulator Encoding Filter FMapType Font FontType Form FormType Generic Halftone HalftoneType HWOptions ImageType IODevice OutputDevice Pattern PatternType ProcSet
Generic	No instances defined

CHAPTER 3

PostScript Level 1 Compatibility Operators

The PostScript 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 and for that purpose has undergone a number of significant extensions. Appendix D, "Compatibility Strategies," in the *PostScript Language Reference Manual*, second edition, presents guidelines for taking advantage of language extensions while maintaining compatibility with PostScript interpreters.

The Personal LaserWriter 320 printer is a Level 2 printer. This chapter describes the compatibility operators that make the Personal LaserWriter 320 printer compatible with existing PostScript Level 1 language driver software. It also explains how to set system, page device, user, device, and communication parameters.

Overview of Compatibility Operators

The compatibility operators present in the Personal LaserWriter 320 printer appear in two dictionaries: `statusdict` and `userdict`. These operators set

- system parameters
- page device parameters
- user parameters
- device parameters
- communication parameters

This chapter describes the page size and paper tray compatibility operators. It also shows you how to set the parameters just listed.

▲ **WARNING**
The operators described in this chapter are included only to support compatibility. You should not use them in PostScript Level 2 programs. ▲

Table 3-1 provides a complete list of compatibility operators arranged by dictionary group.

Table 3-1 Compatibility operators

statusdict

a4tray	monarchtray
appletalktype	pagecount
b5tray	pagestackorder
buildtime	papersize
byteorder	printername
c5tray	product
checkpassword	ramsize
com10tray	realformat
defaulttimeouts	revision
dltray	setdefaulttimeouts
dostartpage	setdostartpage
executivetray	setjobtimeout
jobname	setmargins
jobtimeout	setpagestackorder
legaltray	setprintername
lettertray	setsoftwareiomode
manualfeed	softwareiomode
manualfeedtimeout	waittimeout
margins	

userdict

#copies	legalsmall
=string	letter
a4	lettersmall
a4small	monarch
b5	note
c5	quit
com10	serverdict
dl	smooth4
executivepage	stretch
legal	

Page Size Compatibility Operators

The page size operators are in the user dictionary `userdict`. Each operator requests a specific paper size and imaging boundary box, as shown in Table 3-2. 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 occurs when there is not sufficient memory for the imaging area requested.

Table 3-2 Page size compatibility operators

Operator	Page size	Imaging boundary box
`a4`	[595 842]	[10 7 586 829]
`a4small`	[595 842]	[25 25 570 817]
`b5`	[516 729]	[12 7 503 718]
`c5`	[459 649]	[8 2 453 640]
`com10`	[297 684]	[6 5 290 676]
`dl`	[312 624]	[8 6 307 618]
`executivepage`	[522 756]	[9 6 508 745]
`legal`	[612 1008]	[9 9 600 996]
`legalsmall`	[612 1008]	[58 54 550 954]
`letter`	[612 792]	[9 14 600 783]
`lettersmall`	[612 792]	[24 24 593 766]
`monarch`	[279 540]	[6 5 275 534]
`note`	[width height]	[24 24 *width*–25 *height*–25]

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

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*].

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 operators is the size of paper requested. The `PageSize` and `ImagingBBox` parameters requested are the same as those for the corresponding page size operator. 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.

The paper tray compatibility operators and associated page sizes and imaging boundary box parameters are shown in Table 3-3.

Table 3-3 Paper tray compatibilty operators

Operator	Page size	Imaging boundary box
`a4tray`	[595 842]	[10 7 586 829]
`b5tray`	[516 729]	[12 7 503 718]
`c5tray`	[459 649]	[8 2 453 640]
`com10tray`	[297 684]	[25 25 593 767]
`dltray`	[312 624]	[8 6 307 618]
`executivetray`	[522 756]	[9 6 508 745]
`legaltray`	[612 1008]	[9 9 600 996]
`lettertray`	[612 792]	[9 14 600 783]
`monarchtray`	[279 540]	[6 5 290 676]

Setting System Parameters

System parameters have a system-wide impact, and they may be changed only by a program that presents a valid password. Alterations made to system parameters may persist through restarts of the PostScript interpreter. This section describes the compatibility operators that set Level 2 system parameters.

buildtime

Syntax – **`buildtime`** *int*

Definition This operator is a time stamp that identifies a specific time when a build of the PostScript interpreter took place. It returns an integer with the same value as the system parameter `BuildTime`.

Default value: `false`

Error(s) `stackoverflow`

byteorder

Syntax - **`byteorder`** *bool*

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

Default value: `false`

Error(s) `stackoverflow`

checkpassword

Syntax *int* **checkpassword** *bool*

or

string **checkpassword** *bool*

Definition This operator checks whether *string* or *int* (*int* is converted to a string) is the valid password for either `SystemParamsPassword` or `StartJobPassword`. If the password is valid, it returns `true`. Otherwise, after delaying for one second, it returns `false`.

Default value: `0`

Error(s) `stackoverflow`, `stackunderflow`, `typecheck`

defaulttimeouts

Syntax – **defaulttimeouts** *job manualfeed wait*

Definition This operator returns the following values:

- default job
- manual feed
- wait timeout values

Standard value: `0 60 40`

Error(s) `stackoverflow`

dostartpage

Syntax – **dostartpage** *bool*

Definition This operator returns the Boolean value set during the most recent execution of `DoStartPage`.

Standard value: `true`

Error(s) `stackoverflow`

printername

Syntax *string* **printername** *substring*

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

Standard value: `(Personal LaserWriter 320)`

Error(s) `invalidaccess`, `rangecheck`, `stackunderflow`, `typecheck`

product

Syntax – **`product`** *string*

Definition This operator is a *string* object that is the name of the laser printer product. If a program needs to know what type of printer it is running on, it should check this string.

Standard value: `(Personal LaserWriter 320)`

Error(s) `stackoverflow`

ramsize

Syntax – **`ramsize`** *int*

Definition This operator returns the number of bytes of RAM in the printer. It does this by returning an integer with the same value as the system parameter `RamSize`.

Error(s) `stackoverflow`

realformat

Syntax – **`realformat`** *string*

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

Error(s) `stackoverflow`

revision

Syntax – **`revision`** *int*

Definition This operator is an integer that designates the current revision level of the machine-dependent portion of the PostScript interpreter. It does this by returning an integer with the same value as the system parameter `Revision`.

Error(s) `stackoverflow`

setdefaulttimeouts

Syntax *job manualfeed wait* **`setdefaultimeouts`** –

Definition This operator establishes the default values for the three timeouts. It returns the following:

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

Error(s) `invalidaccess`, `rangecheck`, `stackunderflow`, `typecheck`

setdostartpage

Syntax *bool* **`setdostartpage`** –

Definition Since the Personal LaserWriter 320 printer has no start page, executing this operator with a value of `true` has no effect and is ignored. The operator is present to maintain compatibility with the Personal LaserWriter NT.

Error(s) `invalidaccess`, `stackunderflow`, `typecheck`

setprintername

Syntax *string* **`setprintername`** –

Definition This operator establishes the string to be the printer's name by setting the system parameter `PrinterName` to the value of *string*.

The string should be no longer than 32 characters. It should consist entirely of printing characters and should not contain the following three characters: colon (`:`), comma (`,`) or at sign (`@`).

Error(s) `invalidaccess`, `limitcheck`, `stackunderflow`, `typecheck`

setsoftwareiomode

Syntax *int* **`setsoftwareiomode`** –

Definition This operator sets the values of the interpreter and, if appropriate, protocol device parameters for the current communication device parameter set. The following integer values are used with the Personal LaserWriter 320 printer:

int	Interpreter value	Protocol value
`0`	PostScript	Normal
`100`	PostScript	Binary

Binary protocol is standard in the Personal LaserWriter 320 printer, and it should be used by any driver that sends binary data to the printer. A driver that does not use the binary protocol and does not filter the back channel data when `setsoftwareiomode` is `100` will see receive control characters as quoted characters. For example, Control-D will be seen as Control-A / ASCII-D.

The `softwareiomode` operator does not need to be set outside the server loop to be compatible with other printer implementations of `setsoftwareiomode`. However, changes to it do not take effect until the job that makes the changes is completed.

Standard value: `100`

Error(s) `invalidaccess`, `rangecheck`, `stackunderflow`, `typecheck`

softwareiomode

Syntax `-` **`softwareiomode`** *int*

Definition This operator returns an integer value, which indicates the interpretation code for the current communications device. See `setsoftwareiomode`.

Error(s) `stackoverflow`

Setting Page Device Parameters

Page device parameters control page formatting, for example, margins and paper size. They also control the output processing of pages, determining whether pages are output face up or face down, which paper tray is selected, and so forth. This section describes compatibility operators that set Level 2 page device parameters in the Personal LaserWriter 320 printer.

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.

Standard value: `0 0`

Error(s) `stackoverflow`

pagecount

Syntax – **`pagecount`** *int*

Definition This operator returns the value of the system parameter `PageCount`. That is, it returns the number of pages that have been printed by the Personal LaserWriter 320 printer.

Error(s) `stackoverflow`

pagestackorder

Syntax – **`pagestackorder`** *bool*

Definition This operator returns the last value set by `setpagestackorder`. It should be `true` if the pages are to be stacked face down in the output tray, and `false` if the pages are to be stacked face up.

Error(s) `stackoverflow`

papersize

Syntax – **`papersize`** *name bool*

Definition This operator returns the name of the compatibility 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.

Note that if there is more than one tray installed with the same paper size, and the operator returned by `papersize` is executed again later, the operator will not necessarily select the same tray it selected the previous time.

Error(s) `stackoverflow`

setmargins

Syntax *top left* **`setmargins`** –

Definition This operator sets the two margin adjustment parameters set by `setmargins`.

Error(s) `invalidaccess`, `rangecheck`, `stackunderflow`, `typecheck`

setpagestackorder

Syntax *bool* **setpagestackorder** –

Definition This operator sets the value returned by `pagestackorder`. A value of `true` indicates that the output is going to the face-down tray. A value of `false` indicates that the output is directed to the face-up tray. Since the Personal LaserWriter 320 printer has only a face-down tray, `true` is the only correct value for this parameter.

Error(s) `invalidaccess`, `rangecheck`, `stackunderflow`, `typecheck`

Setting User Parameters

User parameters enable you to control certain printer functions, such as defining job names and selecting the length of time the printer will wait before aborting a print job. Using a PostScript language program, you can change user parameters within reasonable limits, without special authorization. This section describes the compatibility operators that set Level 2 user parameters in the Personal LaserWriter 320 printer.

jobname

Syntax – **jobname** *string*

Definition This operator is a string with the same value as the user parameter `JobName`. It specifies the name of the current job. If a PostScript language program defines `jobname`, status responses generated during the remainder of the job in progress will include a job field that reports the text of this string. The string should not contain the character semicolon (`;`) or end bracket (`]`), since they disrupt the syntax of the status messages.

Standard value: `( )`

Error(s) `stackoverflow`

jobtimeout

Syntax – **jobtimeout** *int*

Definition This operator returns the number of seconds remaining before the job timeout will occur. It does this by returning the value of the user parameter `JobTimeout`. If the returned value is `0`, the job will never time out.

Standard value: `0`

Error(s) `stackoverflow`

setjobtimeout

Syntax *int* **`setjobtimeout`** –

Definition This operator sets the timeout for the current job to the value *int*, a non-negative integer specifying a time interval in seconds. If the current job continues for *int* seconds without either completing or executing `setjobtimeout` again, the PostScript interpreter executes a `timeout` error. The value `0` disables the job timeout altogether.

At the beginning of a job, the server initially sets the job timeout to the default job timeout returned by `defaulttimeouts`. However, in interactive mode, the initial job timeout is always `0`.

Error(s) `rangecheck`, `stackunderflow`, `typecheck`

waittimeout

Syntax – **`waittimeout`** *int*

Definition This operator is the wait timeout currently in effect; that is, it is the number of seconds the Personal LaserWriter 320 printer will wait to receive additional characters from the host before it aborts the current job by executing a `timeout`. At the beginning of a job, the server initializes `waittimeout` to the default wait time returned by `defaulttimeout`. However, a PostScript language program may change it to any non-negative integer value. In interactive mode, the wait timeout is always `0`.

Standard value: `40`

Error(s) `stackoverflow`

Setting Device Parameters

Each PostScript interpreter supports a collection of input/output devices, such as disks, cartridges, and printers. Device parameters perform functions similar to the functions performed by system parameters. However, they are device dependent, which means they impact only the printer for which they are set. This section describes the compatibility operator that sets Level 2 device parameters.

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`, then 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.

Standard value: `false`

Error(s) `stackoverflow`

Setting Communication Parameters

Communication parameters control the functions of the different communication channels, such as AppleTalk and the serial channel. The following compatibility operator sets Level 2 communications parameters in the Personal LaserWriter 320 printer.

appletalktype

Syntax – **`appletalktype`** *string*

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

Error(s) `stackoverflow`

CHAPTER 4

Communication Channels

This chapter deals with the software support for the LocalTalk communication channel. It describes

- LocalTalk communication
- communication protocols
- communication dynamics between the host computer and the printer
- the queries and messages that enable the host computer or the user to know what the printer is doing

You will find information about the physical characteristics of the communication channel connectors in the section "Communication Port for LocalTalk" in Chapter 1. Further information is available in Chapter 2, in the section "Communication Device Parameters," and in Chapter 3, in the section "Setting Communication Parameters."

LocalTalk

The Personal LaserWriter 320 printer can communicate with the host computer or other peripheral devices using the AppleTalk network system. The printer implements the AppleTalk standard protocol using the LocalTalk physical link. The transceiver for transmitting and receiving information over LocalTalk is built into every Macintosh host computer as well as into the Personal LaserWriter 320 printer, making it easy to set up the printer-host interface.

Communication Protocols

Since the Personal LaserWriter 320 is strictly designed to provide a low-cost, high-resolution printing solution for the Macintosh computer, it does not support the simple or binary serial communication protocols.

Communication Dynamics

Data transmitted by the Personal LaserWriter 320 printer, whether it is generated by executing the PostScript language program or by some other spontaneous event such as an error, is logically asynchronous with respect to the data received. This means that the host computer must be prepared to consume data received from the printer while waiting to send more data to the printer. If the host computer is not set up to do this, the printer and the host may each wait for the other to consume data, and a deadlock will occur.

Typically, characters written to the standard output file by PostScript operators such as `print` are not sent immediately. They are buffered until a flush is executed. A flush occurs automatically

- at the end of a job
- in interactive mode, whenever the user is prompted to make an entry

IMPORTANT

If a PostScript language program writes data that is needed immediately by the host, for example, a reply to an environmental query, it is important to flush after writing the data. Otherwise, a deadlock may occur. ▲

Status Queries and Spontaneous Messages

The Personal LaserWriter 320 printer provides a status query facility that enables the host or user to determine what the printer is doing. The printer responds to a status query asynchronously with respect to normal job execution. That is, it sends a response immediately, regardless of what has gone on before or how much input data has been buffered. This facility primarily enables spoolers (printer control programs) to track the activities of the Personal LaserWriter 320 printers under their control.

If the printer receives a Control-T character from the active input channel, it replies with a one-line status message over the active port's output channel. The message is bracketed by the text sequences `%%[` and `]%%`, to enable the host software to extract the message from the ordinary data generated by the job being executed.

The status message has standardized syntax that is intended to be machine readable. It consists of one or more key value pairs, separated by semicolons—for example:

```
%%[job: Jane's report; status: busy; source: LocalTalk ]%%
```

The possible keys, values, and meanings are as follows:

`job`	The name of the job is stored as `jobname` entry in `statusdict` (see Chapter 3). This field is omitted if the current job has not defined `jobname`.
`status`	Indicates what the printer is currently doing: × `idle` indicates no job is in progress ■ `busy` means the printer is executing the user's PostScript language program ■ `waiting` means that the I/O is waiting in the middle of a job ■ `printing` indicates that the printer is printing and that paper is in motion ■ `PrinterError: reason` means that there is a printer error such ■ as a paper jam or printer out of paper `initializing` indicates the printer is starting up
`source`	`serial`, `parallel`, or `LocalTalk` indicates the source of the job that the server is currently executing (this field is omitted if the server is idle)

All messages generated spontaneously by the server (as opposed to those messages produced when the PostScript language program executes `print`) conform to the same syntax as status messages. They are sent as ordinary data through the communication channel in sequence with any other characters written to the standard output file. Consequently, they are always bracketed with `%%[` and `]%%`, for either serial or parallel channels. The following messages are generated spontaneously by the server:

```
%%[Error:error; OffendingCommand: operator ]%%
```

An error has been detected by the PostScript interpreter, and the standard error handler (`handleerror`) has been invoked.

`error` is the name of the error operator originally invoked. `operator` is the operator or other PostScript object being executed at the time of the error.

Refer to the *PostScript Language Reference Manual*, second edition, for further information on error handling.

```
%%[PrinterError:reason]%%
```

A problem has been reported by the printer mechanism. The type of problem is indicated by `reason`: no paper, no paper tray, paper jam, cover open, and so forth.

A printer error can occur only during execution of `showpage` or `copypage`, that is, when the printer is actually trying to print a page. After generating this message, the server usually waits for the condition to be corrected and then continues printing automatically.

```
%%[Flushing: rest of job (to end-of-file) will be ignored ]%%
```

Because of a previous error or abort condition, for example, `stop` or Control-C interrupt, the remainder of the current job is being discarded. The server reads and discards characters from the standard input file until it receives an end-of-file indication.

```
%%[exit server: permanent state may be changed ]%%
```

The PostScript language program has successfully exited from the server's normal `save/restore` context and may now make permanent changes to the system parameters or to the virtual memory.

Index

F

G

H

I

J, K

L

M

N

O

P, Q

R

S

T

U

V

W, X, Y, Z

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh computers and FrameMaker software. Proof pages were created on an Apple LaserWriter IINTX printer. Final pages were printed on the Apple LaserWriter Pro 630. Line art was created using Adobe™ Illustrator. PostScript™, the page-description language for the LaserWriter, was developed by Adobe Systems Incorporated.

Text type is Palatino® and display type is Helvetica®. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Apple Courier.

WRITER
Steve Schwander

DEVELOPMENTAL EDITOR
Beverly Zegarski

Special thanks to Lisa Choy and Timothy Mac

Acknowledgments to Adobe Systems Incorporated