
PostScript® Cartridge Plus Technical Reference



HP Part No. C2089-90921
Printed in U.S.A. February 1992

First Edition

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Notice

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

Hewlett-Packard Company MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

©Hewlett-Packard Company 1992. All rights reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

Printing History

This manual was created on an HP Vectra Personal Computer. The camera-ready text was printed on an HP LaserJet III Si printer and reproduced using standard offset printing techniques.

First Edition February 1992


Trademark Credits

Adobe® and *PostScript*® are registered trademarks of Adobe Systems, Incorporated in the U.S. and other countries.

NetWare® is a registered trademark of the Novell Company.

Conventions

The following conventions are used throughout this guide:

- **Bold** is used for PostScript operators or keys.
- *Italic* refers to a related document, or is used for emphasis.
- COMPUTER type indicates what you should type on the computer keyboard.
- DISPLAY type indicates printer display messages.
- Percent signs (%) used in examples designate comments about the lines of PostScript code.
-  indicates a key on the printer control panel.

Note



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

This guide uses the term *printer* to refer generically to the HP LaserJet III, the HP LaserJet IIID, or the HP LaserJet IIIP printer.

Related Documentation

The following related documents provide further information about the PostScript Cartridge Plus and the PostScript printer language:

1. *The PostScript® Language Reference Manual (Second Edition).*
2. *Adobe's PostScript Language Supplement for Version 2011 (December 20, 1991).*
3. *Adobe's PostScript Language Addendum for the PostScript (Level 2) Cartridge for the Hewlett-Packard LaserJet III Family of Printers*
4. *The PostScript Cartridge Plus User's Guide.*
5. The README file contained on the "PCL/PostScript Device Driver" disk.
6. *The PostScript Cartridge Plus Network Notes.*

Contents

1. Tips for Administrators	
Introduction	1-1
Language Switching Using the Control Panel	1-2
Writing Level 2 Programs for HP LaserJet Printers	1-3
Program Structuring Conventions	1-4
Setting PostScript Language Parameters	1-5
Viewing PostScript Language Parameters' Settings	1-7
Modifying PostScript Language Parameters	1-8
Resetting the Printer to Factory Defaults	1-10
Troubleshooting Tips	1-14
Upgrading from the Level 1 Cartridge	1-14
Viewing Errors	1-15
Setting AutoContinue	1-15
Enabling DoPrintErrors	1-17
Combining DoPrintErrors and AutoContinue	1-18
PostScript Error Messages	1-19
2. PostScript Level 2 Operators and Parameters	
Introduction	2-1
PostScript Language Parameters Illustration	2-1
Description of the Chapter Format	2-3
Operator Name	2-3
Operator Syntax	2-4
Description	2-4
Operator Example	2-4
Operator Errors	2-4

Parameter Tables	2-5
Page Device Operators and Parameters . . .	2-6
setpagedevice*	2-6
setpagedevice Examples	2-7
setpagedevice Errors	2-8
currentpagedevice	2-8
currentpagedevice Errors	2-8
Page Device Parameter Key/Value Pairs .	2-10
Multiple Copies	2-14
Resolution Enhancement (RET)	2-18
I/O Device Operators and Parameters . . .	2-19
setdevparams**	2-21
setdevparams Errors	2-21
currentdevparams	2-22
currentdevparams Errors	2-22
I/O Device Parameter Tables	2-23
Serial	2-23
Parallel	2-25
LocalTalk	2-26
OptionalIO	2-27
System Operators and Parameters	2-29
setsystemparams*	2-30
setsystemparams Examples	2-30
setsystemparams Errors	2-31
currentsystemparams	2-32
currentsystemparams Errors	2-32
System Parameters Key/Value Pairs . . .	2-33
User Operators and Parameters	2-39
setuserparams	2-40
setuserparams Examples	2-41
setuserparams Errors	2-41
currentuserparams	2-42
currentuserparams Errors	2-42

3. PostScript Level 1 and Level 2 Compatability	
Improved Support for Device-Specific Features	
in Level 2	3-6
Known Problems	3-7
Level 1	3-7
Wrong	3-7
Right	3-7
Level 2	3-8
Wrong	3-8
Right	3-8
A. Language Switching Sequence	
B. The Tagged Binary Communications Protocol	
TBCP Versus Adobe Binary Communications	
Protocol	B-3
Implementing TBCP	B-4
Encapsulated PostScript Files and User	
Interface	B-10
Index	

Tables

1-1. setpagedevice and currentpagedevice Defaults	1-11
1-2. setdevparams and currentdevparams Defaults	1-11
1-3. setsystemparams and currentsystemparams Defaults	1-12
1-4. Control Panel Defaults	1-12
1-5. PostScript Test Page Defaults	1-13
1-6. PostScript Error Messages	1-19
2-1. Page Device - Input Media Selection . . .	2-10
2-2. Page Device - Processing and Output Entries	2-13
2-3. Policy and Special Action Entries	2-15
2-3. Page Device - Policy and Special Action Entries (continued)	2-16
2-4. I/O Device - Serial Key/Value Pairs . . .	2-23
2-5. I/O Device - Parallel Key/Value Pairs . .	2-25
2-6. I/O Device - LocalTalk Key/Value Pairs .	2-26
2-7. I/O Device - OptionalIO Key/Value Pairs .	2-27
2-8. System Parameters Key/Value Pairs . . .	2-33
2-9. User Parameter Key/Value Pairs	2-43
B-1. Special Control Characters	B-5
B-2. Results of Special Control Characters . . .	B-9

Tips for Administrators

Introduction

The PostScript Cartridge Plus from Hewlett-Packard is a new PostScript cartridge for the HP LaserJet III, IIID, and IIIP printers. Its new features are:

- Language switching between PostScript and PCL
- Adobe PostScript Level 2

The PostScript Cartridge Plus allows users to switch between PCL and PostScript without turning off the printer, removing or inserting the cartridge, and turning on the printer again. To switch printer languages you can use:

- The printer control panel. (Refer to the next section for more information.)
- The HP PCL/PostScript device driver for DOS users not printing to a printer connected to a network. (Refer to the *PostScript Cartridge Plus User's Guide* for more information.)
- An HP Terminate-and-Stay-Resident (TSR) program for users printing to a printer connected to a Novell NetWare network. (Refer to the *PostScript Cartridge Plus Network Notes* for more information.)
- An HPNPRINT utility for users printing to a printer connected to a Novell NetWare network. (Refer to the *PostScript Cartridge Plus Network Notes* for more information.)
- DOS batch files. (Refer to the *PostScript Cartridge Plus User's Guide* for more information.)

Language Switching Using the Control Panel

Use the following steps to switch from PostScript to PCL, or vice versa, using the printer control panel:

- For an HP LaserJet III or IIID:
 1. Press the **ON LINE** key to take the printer offline.
 2. Hold down the **+** key, and continue to hold it while you press the **RESET** key.
 3. Release both keys when either PCL START or POSTSCRIPT START appears.
- For the HP LaserJet IIIP:
 1. Press the **ON LINE** key to take the printer offline.
 2. Hold down the **+/-** key, and continue to hold it down while you press *both* the **Alt** and **Reset** keys.
 3. Release the keys when either PCL START or PS START appears.

Writing Level 2 Programs for HP LaserJet Printers

Printer administrators may want to set the printer's default environment, and certain persistent parameters like system passwords or **DoAutocontinue**. To do this the administrator may need to write new programs or translate old Level 1 programs to Level 2 programs. This will ensure the defaults set up under the old Level 1 programs will work in Level 2. Refer to chapter 3, "PostScript Level 1 and Level 2 Compatibility," or appendix A in the *PostScript Language Reference Manual (Second Edition)* for more information.

Consider the following when writing PostScript Level 2 programs.

- Use a text type editor to create the Level 2 programs. If you use a word processor, save the program as ASCII text.
- Do not use the print option of the editor to send the program to the printer. Instead, exit the editor and use the DOS COPY command: `COPY/B filename LPT1`
- Use the % sign to make comments in the programs so you can identify them for later use.
- Be careful with the system passwords. If you forget the passwords they can be cleared using the printer cold reset. A cold reset will return them to factory default settings.
- End all PostScript programs with an End-of-File marker.
- Enable **DoPrintErrors** and/or set AutoContinue to OFF while writing PostScript programs. Refer to the section "Viewing Errors," or section 2.3 in the *PostScript Language Supplement for Version 2011* for more information on error handling routines.

Program Structuring Conventions

PostScript programs should follow the Document Structuring Conventions as described in appendix G of the *PostScript Language Reference Manual (Second Edition)*. By following those conventions (listed on pp. 621), PostScript programs are recognized as PostScript by the automatic language switching algorithms in the PostScript Cartridge Plus software.

Setting PostScript Language Parameters

All PostScript printers use a standard set of PostScript parameters to describe the appearance of a printed page. The PostScript Cartridge Plus supports this standard set of parameters. In addition, device-specific parameters are included to allow access to special printing features supported by the HP LaserJet III family of printers.

There are two general categories of Level 2 PostScript language parameters:

- Page Device parameters
- Interpreter parameters

There are three sub-categories within the Interpreter parameters category:

- System parameters
- User parameters
- Device parameters

Each category and sub-category has its own set of operators that manage access to its parameter set. The parameter sets and operators that manage them are described in chapter 2. The following categories of parameter set values are indicated in the “Type” field of the tables in chapter 2:

- *volatile* - New value is retained only for the duration of the job.
- *semi-persistent* - New value is retained across job boundaries as long as the printer remains turned on, or until a language switch occurs if set in an unencapsulated job; otherwise it is volatile.
- *persistent* - New value is retained across job boundaries, power cycles, switching languages, and control panel resets if set in an unencapsulated job; otherwise it is volatile.
- *read-only* - Value is not changeable by way of PostScript, but may be changed through the control panel.
- *constant* - Value never changes.
- *(persistent/read-only)* - Value changes by internal mechanisms only and is saved persistently.
- *write-only* - Value can be set but not read.

Write access to the parameter sets is provisionally controlled by a password mechanism. Read access is not restricted (except for the password values themselves which are write-only). Different password mechanisms are appropriate for modifying Level 2 PostScript Page Device parameters as opposed to System or Device parameters. However, the mechanisms are identical for accessing and changing either standard or device-specific features within each category or sub-category of parameters.

Viewing PostScript Language Parameters' Settings

You can view the PostScript language parameters' settings four different ways:

- Run a printer PostScript self test.

For HP LaserJet III or IIID printers:

1. Press the **ON LINE** key to take the printer offline.
2. Hold down the **TEST** key until TEST JOB QUEUED appears.
3. Press the **ON LINE** key to print the test page.

For HP LaserJet IIIP printers:

1. Press the **ON LINE** key to take the printer offline.
2. Hold down the **ALT** key and press the **TEST** key. The message TEST JOB QUEUED will appear.
3. Press the **ON LINE** key to print the test page.

The PostScript test page contains the most commonly used parameters and their values.

- Scroll through the printer control panel menu settings. The default settings, indicated by asterisks (*), will show you some of the parameter values, such as the duplex and RET settings.
- To view all of the parameters' settings, you can use a communications program and communicate with the printer interactively. (Refer to section 2.4.4 and chapter 4 of the *PostScript Language Reference Manual (Second Edition)* for more information.)
- You can write a PostScript program that causes the current parameter settings to print.

Modifying PostScript Language Parameters

Modification of System and Device parameters is controlled by a system password set up using **SystemParamsPassword**. **SystemParamsPassword** is itself a System parameter. For more information on setting up a system password, refer to the *PostScript Language Reference Manual (Second Edition)* pp. 573, 575, and 576.

The following is an example of using **SystemParamsPassword** to persistently modify System parameters which turn on the default error handler, increase the default size of the font cache, and change the default non-LocalTalk printer name:

Example 1-1

```
<</DoPrintErrors true           % turn on printing of errors
/MaxFontCache 400000           % increase default cache size
/PrinterName (HP LaserJet myname) % change to a new name
/Password (my_system_password) >> % password required if set
setsystemparams
```

Modification of Page Device parameters is controlled by alteration of initial Virtual Memory (VM) using a **StartJobPassword** with the **startjob** mechanism. For more information on **startjob**, refer to the *PostScript Language Reference Manual (Second Edition)* pp. 69-70, 743, and 576. **startjob** is similar to the Level 1 **exitserver**. For a discussion of **exitserver**, see chapter 3 and the *PostScript Language Reference Manual (Second Edition)* pp. 70-71 and 740.

The following example persistently modifies the priority array to select tray 1 first for imaging and turns off tray switching for paper-out conditions:

Example 1-2

```
true (my_job_password) startjob not           % exit encapsulated job state
<< /Input Attributes << /Priority [1 2 0] >> % modify priority array
/TraySwitch false >>                         % turn off TraySwitch
setpagedevice
false (my_job_password) startjob not         % reenter encapsulated state
```

The following example shows how to set up a password for **startjob** when a **SystemParamsPassword** is active:

Example 1-3

```
<< /Password (my_system_password)           % password required if set
/StartJobPassword (my_job_password)         % set up job password
>> setsystemparams
```

Note



Password values are strings. Integer arguments to either password mechanism converts to a string object.

Modification of User parameters does not require a password since modification does not extend beyond PostScript job boundaries.



Resetting the Printer to Factory Defaults

You can return all PostScript control panel settings and persistent parameters (*except for the PostScript page count*) to factory defaults by performing a “cold” reset. A cold reset does not change the PCL settings while the PostScript Cartridge Plus is installed. If you want to return the PCL settings to factory defaults, perform a cold reset without the PostScript Cartridge Plus installed.

Note



You reset the printer’s configuration by performing a cold reset. For example, the I/O setting is set to PARALLEL after a cold reset. If you are using the serial or AppleTalk port, you need to reconfigure the I/O setting before printing.

To perform a cold reset:

1. Turn off the printer.
2. Turn on the printer while holding down the **ON LINE** key.
3. Keep holding down the **ON LINE** key until **⊘⊘ COLD RESET** appears.

The parameters and keys in the following five tables are reset to the factory defaults when you perform a cold reset.

Table 1-1.
setpagedevice and currentpagedevice
Defaults

Parameter	Type	Default Value
Priority	array [x y z]	[0 1 2]
NumCopies	integer or null	null
ManualFeed	boolean	false
Duplex	boolean	false
Tumble	boolean	false
Margins	array [x y]	[0 0]
ExitJamRecovery	boolean	false
ManualFeedTimeOut	integer	60
TraySwitch	boolean	true

Table 1-2.
setdevparams and currentdevparams Defaults

Parameter	Type	Default Value
LocalTalkType	string	LaserWriter

Table 1-3.
setsystemparams and currentsystemparams
Defaults

Parameter	Type	Default Value
SystemParamsPassword	string	empty string ()
StartJobPassword	string	empty string ()
MaxFontCache	integer	350000
SerialNumber	integer	none
DoPrintErrors	boolean	false
PrinterName	string	HP LaserJet

Note



PageCount is stored in non-volatile memory, but is not reset with a cold reset.

Table 1-4. Control Panel Defaults

Key	Default Value
COPIES	1
DUPLEX	OFF
TRAY	ANY
MANUAL FEED	OFF
AUTO CONT	ON
I/O	PARALLEL
START PAGE	ON
LANGUAGE	ENGLISH
RET	MEDIUM
SW MODE	PS

Table 1-5. PostScript Test Page Defaults

Key	Default Value
Password	empty string () shown as 0
Jam Recovery	false
Auto Continue	true
Print Errors	false
RET	2
Start Page	true
Left Margin	0
Top Margin	0
Page Stack Order	true
Maximum Size of Font Cache	350000
Printer Name	HP LaserJet
Active Communications Channel	(%Parallel%)
Job Timeout	0
Manual Feed Timeout	60
Wait Timeout	40
Duplex Mode	false
Tumble Mode	false
Paper Tray	0
Tray Switch	true

Troubleshooting Tips



When the PostScript Cartridge Plus is installed, and the printer is in PostScript mode, the control panel settings are stored in the printer in a block of non-volatile RAM (NVRAM). This block of PostScript NVRAM is separate from the block of NVRAM in which the PCL settings are stored. You must set the `LANG=` menu item in *both* PostScript and PCL modes if language switching is to occur. For example, setting a menu item like `DUPLEX=ON` in PostScript mode has no effect on the PCL duplex setting. The separate blocks of NVRAM are the reason why the `SW MODE=` menu item only appears if the printer is in PostScript mode.

Consider the information in the following sections when you encounter problems.

Upgrading from the Level 1 Cartridge

If you upgrade a printer from the Level 1 PostScript Cartridge (33439P or 33439Q) to the PostScript Cartridge Plus, all PostScript control panel settings and persistent parameters (including the PostScript page count) return to factory defaults. However, this does not reset the PCL configuration settings.

This also happens if you switch from the PostScript Cartridge Plus to the Level 1 cartridge. Because the two cartridges share the same block of printer NVRAM, but use it differently, the block of NVRAM must be reinitialized for the other cartridge to use it.

Viewing Errors

One class of errors that can be generated by the PostScript Cartridge Plus are PostScript errors (PS ERROR xx). When the PostScript Cartridge Plus detects a PostScript error, two things control how it reports the error. One is the AUTO CONT control panel setting (which corresponds to the Level 2 system parameter **DoAutoContinue** and the Level 1 **doautocontinue** operator). The other is the system parameter **DoPrintErrors** (which corresponds to the Level 1 **doprinterrors** operator).

Setting AutoContinue

AUTO CONT controls whether or not a PS ERROR message appears on the control panel when a PostScript error occurs. AUTO CONT= can be set to ON* or OFF* on the control panel, or by sending the appropriate PostScript code from the host to set the system parameter **DoAutoContinue** to "true" or "false." The factory default for AUTO CONT is ON* and when you are not in troubleshooting mode keep this set to ON*, especially for shared printers.

Hewlett-Packard recommends that you set AutoContinue to OFF if you suspect an error.

Use the following steps to turn AutoContinue to OFF on the HP LaserJet III or IIID printer control panel:

1. Press the **ON LINE** key to take the printer offline.
2. Hold down the **MENU** key for 3 to 5 seconds until AUTO CONT= appears.
3. Press the **+** key repeatedly until AUTO CONT=OFF appears.
4. Press the **ENTER** key to save the selection. An asterisk (*) will appear.
5. Press the **ON LINE** key to return the printer online.

Use the following steps to turn AutoContinue to OFF on the HP LaserJet IIIP printer control panels:

1. Press the **ON LINE** key to take the printer offline.
2. Press the **MENU** key repeatedly until DEVICE CONFIG appears.
3. Press the **ENTER** key.
4. Press the **MENU** key until AUTOCONT appears.
5. Press the **+** key until OFF appears in the second line of the display.
6. Press the **ENTER** key to save the selection. An asterisk (*) will appear.
7. Press the **ON LINE** key to return the printer online.

Enabling DoPrintErrors

You can also view errors by sending the following text file to turn on print errors. The system parameter **DoPrintErrors** controls whether or not the printer prints a page describing the PostScript error. The contents of the page are:

ERROR - This is the PostScript name for the error (for example, **undefined**). Information about the error can be found in the section “PostScript Error Messages” and the *PostScript Language Reference Manual (Second Edition)*.

OFFENDING COMMAND - This is the PostScript object that was being executed at the time of the error (for example **moveto**).

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

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

A PostScript job that sets DoAutoContinue to “false” and DoPrintErrors to “true” looks like this:

Example 1-4

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

Combining DoPrintErrors and AutoContinue

The following are the possible combinations of using **DoPrintErrors** and **AutoContinue**, and the results when an error occurs in each combination.

- **AUTO CONT=ON** and **DoPrintErrors** set to “false,” the printer:
 - does not display the PS ERROR xx on the control panel.
 - does not print a page describing the error.
 - flushes the job until a Control-D is encountered.
- **AUTO CONT=OFF** and **DoPrintErrors** is set to “false,” the printer:
 - displays the PS ERROR on the control panel.
 - waits for the **CONTINUE** key to be pressed on the control panel.
 - does not print a page describing the error.
 - flushes the job until a Control-D is encountered.
- **AUTO CONT=ON** and **DoPrintErrors** is set to “true,” the printer:
 - does not display the PS ERROR on the control panel.
 - prints a page describing the error.
 - flushes the job until a Control-D is encountered.
- **AUTO CONT=OFF** and **DoPrintErrors** is set to “true,” the printer:
 - displays the PS ERROR on the control panel.
 - waits for the **CONTINUE** key to be pressed on the control panel.
 - prints a page describing the error.
 - flushes the job until a Control-D is encountered.

PostScript Error Messages



POSTSCRIPT

Various errors can occur during the execution of a PostScript program. Each error has a name and is associated with a PS ERROR *number* that appears on the printer, and an error handler. For each error, the error number is given, followed by the error name and description. Most PostScript errors do not appear on the printer control panel if AutoContinue (AUTOCCONT) is set to ON.

Errors 25-29 are unique to the Level 2 implementation. All others existed in Level 1, although QuitExecuted and fatal error have been expanded into 2 separate errors. (See the *PostScript Language Reference Manual (Second Edition)* (pp. 99-101) for more information.)

Table 1-6. PostScript Error Messages

PS ERROR Number	Name	Description
00	dictfull	Dictionary full. A def , put or store operator attempted to define a new entry in a dictionary that is already full (for example, whose length and maxlength are already equal).
01	dictstackoverflow	Dictionary stack overflow. The dictionary stack is too large. Too many begins (without corresponding ends) have pushed too many dictionaries on the dictionary stack.
02	dictstackunderflow	Dictionary stack underflow. An attempt has been made to remove the bottommost instance of userdict from the dictionary stack. dictstackunderflow occurs if an end is executed for which there was no corresponding begin .
03	execstackoverflow	Execution stack overflow. The execution stack has grown too large; procedure invocation is nested deeper than the PostScript interpreter permits.

Table 1-6. PostScript Error Messages (continued)

PS ERROR Number	Name	Description
04	invalidaccess	Invalid access of array, dictionary, file, or string object. An attempt has been made to reference an array, dictionary, file, or string object in a way that violates its access attribute (for example, store into a read-only array). This error also occurs if pathforall is executed when the current path includes the result of a charpath.
05	invalidexit	No loop to exit. An exit has been executed for which there is no dynamically enclosing looping context (for , loop , repeat , or pathforall), or it has attempted to leave the context of a run or stopped operator.
06	invalidfileaccess	Invalid file access. The access string specification to the file operator is unacceptable.
07	invalidfont	Invalid font name or dictionary. The operand to makefont or setfont is not a well-formed font dictionary. (The invalidfont error may also be executed by other font operators when a font dictionary is malformed in some way.)
08	invalidrestore	An improper restore has been attempted. One or more of the operand, dictionary, or execution stacks contains composite objects whose values were created more recently than the save whose context is being restored. Since restore would destroy those values (but the stacks are unaffected by restore) the outcome would be undefined and cannot be allowed.
09	ioerror	An exception (other than end-of-file) has occurred during execution of one of the file operators. Attempting to write to an input file or to a file that has been closed also causes an ioerror error.
10	limitcheck	A PostScript implementation limit has been exceeded.

Table 1-6. PostScript Error Messages (continued)

PS ERROR Number	Name	Description
11	nocurrentpoint	The current path is empty, and thus there is no current point, but an operator requiring a current point has been executed (for example, lineto , curveto , currentpoint , or show). The most common cause of this error is neglecting to perform an initial moveto .
12	rangecheck	Operand out of bounds. A numeric operand's value is outside the range expected by an operator (for example, an array or string index is out of bounds, or a negative number appears where a non-negative number is required).
13	stackoverflow	Operand stack overflow. The operand stack has grown too large. Too many objects have been pushed on the stack and not popped off.
14	stackunderflow	Operand stack underflow. An attempt has been made to remove an object from the operand stack when it is empty. This usually occurs because an operator did not have all of its required operands on the stack.
15	syntaxerror	Syntax error in program text. The scanner has encountered program text that does not conform the PostScript syntax rules. This occurs either during interpretation of an executable file or string object or during explicit invocation of the token operator.
16	timeout	A time limit has been exceeded. The PostScript interpreter timed out waiting for media to be manually fed, waiting for data to be received from the host, or due to a job executing too long.

Table 1-6. PostScript Error Messages (continued)

PS ERROR Number	Name	Description
17	typecheck	The wrong type of operand. The type of an operand might be different from what the operator expects. The typecheck error is often the result of faulty stack manipulation, such as operands supplied in the wrong order, or procedures leaving results on the stack when they are not supposed to.
18	undefined	Name not known. A name used as a dictionary key in some context cannot be found. This occurs if a name is looked up explicitly in a specified dictionary (get) or in the current dictionary stack (load) and is not found. It also occurs if an executable name is encountered by the interpreter and is not found in any dictionary on the dictionary stack. The undefined error is commonly caused by misspelling the operator name or by not having the dictionary containing the operator definition on the dictionary stack.
19	undefinedfilename	File not found. A file identified by a name string operand of file or run cannot be found or cannot be opened. The undefinedfilename error also occurs if the special file '%statementedit' or '%lineedit' is opened when the standard input file has reached end-of-file.
20	undefinedresult	Over/underflow or meaningless result. A numeric computation produces a meaningless result or one that cannot be represented as a PostScript number. Possible causes include numeric overflow or underflow, division by zero, or inverse transformation of a non-invertible matrix.
21	unmatchedmark	Expected mark not on stack. A mark object is sought on the operand stack by the] , cleartomark , or counttomark operator, but none is present.

Table 1-6. PostScript Error Messages (continued)

PS ERROR Number	Name	Description
22	unregistered	Internal error. An operator object has been executed for which the interpreter has no built-in action. This represents an internal malfunction in the PostScript interpreter and should never occur.
23	VMerror	An error has occurred in the virtual memory (VM) machinery. The most common problem is exhausted virtual memory. This error causes a control panel reset.
24	quitexecuted	The systemdict quit operator executed. The PostScript interpreter re-initialized virtual memory. All non-persistent information known by the PostScript interpreter was lost. This error causes a control panel reset.
25	fatal error	A firmware error internal to PostScript was detected. This error will cause a control panel reset.
26	configurationerror	A setpagedevice request cannot be satisfied.
27	interrupt	External interrupt request (for example, Control-C).
28	undefinedresource	Resource instance not found.

Any errors not described in table 1-6 are not implemented in the PostScript Cartridge Plus product.

See the *PostScript Language Reference Manual (Second Edition)* pp.99-101 for a discussion of the **handleerror** name.

PostScript Level 2 Operators and Parameters

Introduction



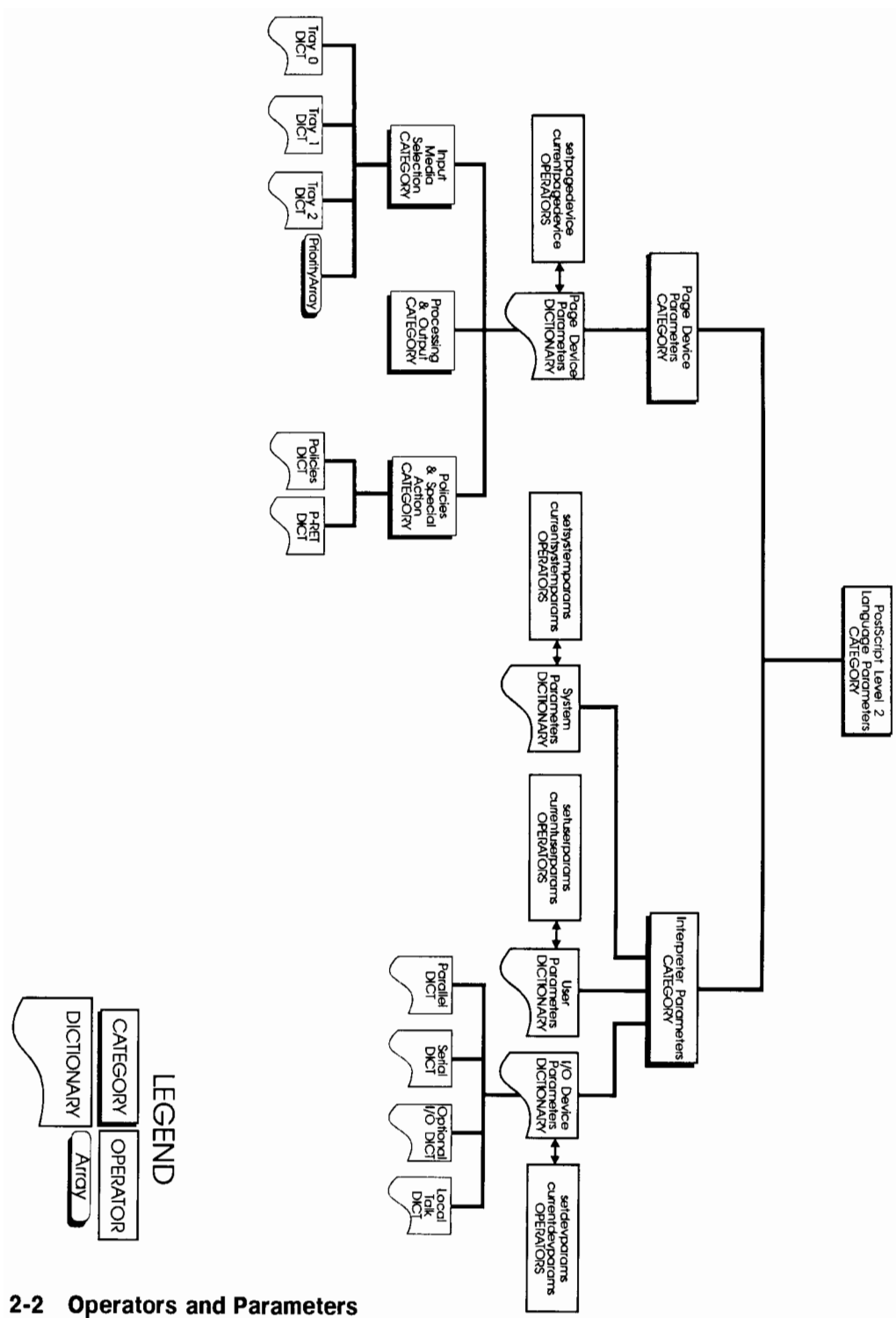
POSTSCRIPT®

The following section, “Description of the Chapter Format,” describes how each operand and its explanation is formatted in the chapter. The sections, “Page Device Operators and Parameters,” “I/O Device Operators and Parameters,” “System Operators and Parameters,” and “User Operators and Parameters,” list the language parameter operators defined for HP LaserJet printers. These operators are grouped into sections related to their function. Operators used to change persistent parameters are marked with an asterisk (*).

PostScript Language Parameters Illustration

The illustration on the next page describes how the PostScript Language parameters, their subcategories, and their subdictionaries are ordered in this chapter.





2-2 Operators and Parameters

Description of the Chapter Format

This chapter documents Hewlett-Packard's implementation of PostScript Level 2 operators. The operators are grouped by functionality. Each operator is documented in the following manner:

Operator Name	Operator Syntax
	Description
	Operator Example
	Operator Errors
	Parameter Table

Operator Name

If a single asterisk (*) appears following the operator name, then the operator modifies a persistent parameter and changes non-volatile memory (NVRAM). To successfully execute the operator, the PostScript interpreter must be executing an unencapsulated job (unless otherwise noted).

If a double asterisk (**) appears following the operator name, then the operator applies in a semi-persistent manner. System default values are used as defaults when the printer is turned off, then back on. There are few semi-persistent operators in this implementation.

Operators with no asterisks make changes that remain in effect ONLY through the current print job, but only if the operator is not read-only.

Operator Syntax

The first line describes the usage of the operator. The general format for operator usage is:

*operand*₁..*operand*_{*n*} **operator** *result*₁..*result*_{*m*}

where *operand*₁ through *operand*_{*n*} are the operands required by the operator, with *operand*_{*n*} being the topmost element on the operand stack, **operator** is the name of the operator, and *result*₁ through *result*_{*m*} are the results left on the operand stack after the successful execution of the operator.

The notation '-' in the operand position indicates that the operator does not consume any objects off of the operand stack. The notation '-' in the result position indicates the operator does not leave any results on the operand stack.

Description

The third part of an operator sub-section is the operator description. A paragraph describes function and limitations of the operator.

Operator Example

The fourth part of an operator sub-section provides you an example of using the operator.

Operator Errors

This section lists the PostScript errors that can occur when misusing the operator.

Parameter Tables

The tables at the end of each section list the parameters implemented in this product. A brief description is included, but for additional information refer to the *PostScript Language Supplement for Version 2011* and the *PostScript Language Reference Manual (Second Edition)* (sections 3.3.9 and 4.1.1). Any parameters not included here, but described in those documents are not supported in this product.

Refer to the *PostScript Language Reference Manual (Second Edition)* chapter 8 for more information on operand syntax.

Page Device Operators and Parameters

The Page Device parameters are one category of PostScript language parameters for Level 2. The Page Device parameter sets are modified by the **setpagedevice** operator and returned by the **currentpagedevice** operator. The Page Device parameters relate to printing capabilities in general. For example, these parameters determine how input media should be selected, what choices for image sizes are available, how pages should be processed/output, and how exceptions/errors should be handled. First, the operators are described, then the parameter sets are described in tables. These tables have been organized into three broad categories (Input Media, Processing and Output, and Policy and Special Action) but there is only *one* dictionary containing all of the Page Device parameters.

setpagedevice*

dict setpagedevice -

dict consists of the key/value pair parameters described in tables 2-1, 2-2, and 2-3.

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

Calls to **setpagedevice** are cumulative: information established in a previous call to **setpagedevice** will persist unless overridden explicitly in a subsequent call. Therefore, there are no required keys in any particular call. This behavior not only applies to the top-level dictionary, but also to the sub-dictionary **Policies**. An exception is **InputAttributes**. It is necessary to specify all of the entries for a given tray to change any one entry. See the **setpagedevice** example 2-2 on changing the **InputAttributes** dictionary.

setpagedevice Examples

A page description might request an executive image **PageSize** which would initiate a search for a tray matching the requested **PageSize**. If no tray matched the request, a policy is invoked according to the **PageSize** value in the **Policies** dictionary:

setpagedevice Example 2-1

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

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

setpagedevice Example 2-2

```
true 0 startjob not           % exit encapsulated job state
<</InputAttributes           % access InputAttributes
<<0<</Pagesize [612 792]/MediaColor (blue) % access Tray 0 and modify Color,
                               % Weight, Value of media
/MediaWeight (20)/MediaType (Letterhead)>> % completes statement to access
                                           % InputAttributes
/Priority [1 2 0]             % also modify the Priority array to
                               % make Tray 1 default
>>>>setpagedevice
false 0 startjob not         % return to encapsulated job state;
                               % changes take affect
```

setpagedevice Errors

- **invalidaccess**
- **limitcheck**
- **rangecheck**
- **stackunderflow**
- **typecheck**
- **configurationerror**

currentpagedevice – **currentpagedevice** *dict*

dict consists of the key/value pair parameters described in tables 2-1, 2-2, and 2-3.

currentpagedevice returns a read-only dictionary containing the key/value pairs of the pagedevice dictionary described in tables 2-1, 2-2, and 2-3. This read-only dictionary describes the page-oriented output device in the current graphics state. **currentpagedevice** creates a new dictionary if necessary. If the device in the current graphics state is not a page device, **currentpagedevice** returns an empty dictionary. Refer to the *PostScript Language Reference Manual (Second Edition)* pp. 229 and 390 for more information.

currentpagedevice Example 2-3

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

currentpagedevice Errors

- **stackoverflow**
- **VMerror**

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

- **InputAttributes**
- **Policies**
- **PostRenderingEnhanceDetails**

Note that the **InputAttributes** dictionary has the 0, 1, and 2 subdictionaries in addition to its entries. These subdictionaries are always present. If they are not applicable to a particular printer, they have a null value.

The subdictionaries 0, 1, and 2 are defined for the different HP LaserJet printers in the following way:

HP LaserJet III printer

- 0 = Upper Cassette
- 1 = null
- 2 = null

HP LaserJet IIID printer

- 0 = Upper Cassette
- 1 = Lower Cassette
- 2 = Envelope Feeder (if installed)

HP LaserJet IIIP printer

- 0 = Upper/Multipurpose Tray
- 1 = Lower Cassette (if installed)
- 2 = null

**Page Device
Parameter Key/Value
Pairs**

The Page Device dictionary in this product has the following entries:

Table 2-1. Page Device - Input Media Selection

Key	Type	Default	Semantics
PageSize	array [x y] <i>(persistent/ semi-persistent)</i>	[612 792] initially (letter size)	currentpagedevice entry specifying size of page requested. Default PageSize of [x=width y=height] installed at start of job, based on the first element in the Priority array.
MediaColor	string or null <i>(semi-persistent)</i>	null	currentpagedevice entry specifying media color requested.
MediaWeight	number or null <i>(semi-persistent)</i>	null	currentpagedevice entry specifying media weight requested.
MediaType	string or null <i>(semi-persistent)</i>	null	currentpagedevice entry specifying media type requested.
ManualFeed	boolean <i>(persistent)</i>	false	If true, media is drawn from the manual feed position.

InputAttributes dictionary - This table describes the parameters for the Tray 0 subdictionary.

Key	Type	Default	Semantics
0	dictionary (<i>read-only</i>)		Upper Tray or Multipurpose Tray
PageSize	array [x y] (<i>read-only</i>)	[612 792] initially	Specifies size of media in this tray. Determined by tray sensor hardware.
MediaColor	string or null (<i>semi-persistent</i>)	null	Specifies color of media in this tray.
MediaWeight	number or null (<i>semi-persistent</i>)	null	Specifies weight of media in this tray.
MediaType	string or null (<i>semi-persistent</i>)	null	Arbitrary string specifying the type of the media in this tray.

InputAttributes dictionary - This table describes the parameters for the Tray 1 subdictionary.

1	dictionary (<i>read-only</i>)		Lower Tray For HP LaserJet IIID and IIP only.
PageSize	array[x y] (<i>read-only</i>)	[612 792] initially	Specifies size of media in this tray. Determined by tray sensor hardware.
MediaColor	string or null (<i>semi-persistent</i>)	null	Specifies color of the media in this tray.
MediaWeight	number or null (<i>semi-persistent</i>)	null	Specifies weight of the media in this tray.
MediaType	string or null (<i>semi-persistent</i>)	null	Arbitrary string specifying the type of the media in this tray.

InputAttributes dictionary - This table describes the parameters for the Tray 2 subdictionary.

Key	Type	Default	Semantics
2	dictionary (<i>read-only</i>)		Envelope Feeder (HP LaserJet IID printer only.)
PageSize	array [x y] (<i>read-only</i>)	[297 684] initially	Specifies size of media in this tray. Com10 envelope size is the initial default.
MediaColor	string or null (<i>semi-persistent</i>)	null	Specifies color of the media in this tray.
MediaWeight	number or null (<i>semi-persistent</i>)	null	Specifies weight of the media in this tray.
MediaType	string or null (<i>semi-persistent</i>)	null	Arbitrary string specifying the type of the media in this tray.

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

Priority	array [x y z] (<i>persistent/ semi-persistent</i>)	[0 1 2]	Prioritizes the order in which the paper or envelope trays are accessed. Priority can be set persistently for [0 1 2], [1 2 0], or [2 0 1]. Other values are accepted for semi-persistent operation.
----------	---	---------	--

InputAttributes PageSizes are set up automatically at the beginning of each job according to the installed trays. Because HP LaserJet printers can sense the upper and lower cassette sizes, **InputAttributes** PageSizes should not be modified by any software. On some printers, one or more of the **InputAttributes** dictionaries (0, 1, or 2) may have a null value, indicating that no tray is installed.

2-12 Operators and Parameters

Table 2-2. Page Device - Processing and Output Entries

Key	Type	Default	Semantics
ImagingBBox	array or null [lx ly ux uy] (<i>semi-persistent</i>)	null	Optional page bounding box which is described as a four element array of integers.
NumCopies ¹	integer or null (<i>persistent</i>)	null	Default number of copies.
Duplex	boolean (<i>persistent</i>)	false	On HP LaserJet IIID, this specifies whether consecutive pages are to be printed on opposite sides of the same sheet of paper.
Tumble	boolean (<i>persistent</i>)	false	false = The second side of a duplexed page has the same orientation as the first. true = The second side of a duplexed page is upside down in comparison to the first.
OutputFaceUp	boolean (<i>read-only</i>)	false	On an HP LaserJet IIID, this value is determined by the position of the Output Selector knob on the back of the printer.
HWResolution	array [x y] (<i>constant</i>)	[300 300]	This is always [300 300] since the supported HP LaserJet printers print 300 dots per inch (DPI).
Margins	array [x y] (<i>persistent</i>)	[0 0]	Relocates page image on the media by [x y] device units.
OutputPage	boolean (<i>semi-persistent</i>)	true	Indicates whether a page should be output. If it is set to "false" the program is processed, but no pages print.

¹ Refer to the following section, "Multiple Copies," for more information on NumCopies.

Note



There is no **OutputAttributes** dictionary or **OutputType** since the printers supported do not have multiple destination trays.

Multiple Copies

Three separate mechanisms are involved with determining the number of copies to be output. Each is referred to in the following manner:

NumCopies = Level 2 Page Device key

#copies = Level 1 entry now in Systemdict

COPIES = control panel setting

If the **NumCopies** entry in the page device is not null, it specifies the number of copies to produce of each page of the document. A null value indicates that **showpage** and **coppypage** should consult the value of **#copies** in the current dictionary stack each time they are executed.

Additionally, the number of copies can also be specified from the control panel. At the beginning of each job, if the control panel value of **COPIES** has changed and the current value of **NumCopies** is not null, the control panel value is copied into **NumCopies**. If the control panel value of **COPIES** has changed and the value of **NumCopies** is null, the control panel value is copied into **#copies**.

If the **NumCopies** entry is set persistently in an unencapsulated job, the value is copied back to the control panel value of **COPIES** at the end of the job. Setting **#copies** in an unencapsulated job causes its value to be retained for subsequent jobs (semi-persistent operation), but does not affect either the **NumCopies** entry or the control panel setting of **COPIES**. In an initialized printer, the default control panel value of **COPIES** is 1, the value of **NumCopies** is null, and the value of **#copies** is 1.

Table 2-3. Policy and Special Action Entries

Key	Type	Default	Semantics
Policies	dictionary (read-only)		Contains feature-policy pairs that specify what setpagedevice should do when a feature request cannot be satisfied.
PolicyNotFound	integer (<i>semi-persistent</i>)	1	0 = Generate a configurationerror . 1 = Ignore the feature request. 2 = Interact with a human (Not implemented in this product).
PageSize	integer (<i>semi-persistent</i>)	6	0 = Generate a configurationerror . 1 = Ignore the requested PageSize . 2 = Interact with a human (Not implemented in this product). 3 = Select the nearest available medium and adjust the page to fit. 4 = Select the next larger medium and adjust the page to fit. 5 = Select the nearest available medium but do not adjust the page to fit. 6 = Select the next larger medium but do not adjust the page to fit. 7 = Select the next larger medium (or nearest if no larger medium is available), but do not adjust the page to fit.
PolicyReport	procedure (<i>semi-persistent</i>)	{pop}	Called upon successful completion of setpagedevice if it is needed to consult policies in order to handle one or more unsatisfied feature requests.
<i>any feature name</i> (See the note at the end of table 2-3.)	integer (<i>semi-persistent</i>)	1	OPTIONAL. 0 = Generate a configurationerror . 1 = Ignore the feature request. 2 = Interact with a human (Not implemented in this product). <i>See the note at the end of this table.</i>

Table 2-3. Page Device - Policy and Special Action Entries (continued)

Key	Type	Default	Semantics
Install ¹	procedure (<i>semi-persistent</i>)	See footnote 1	Executed to install values in the graphics state during each invocation of setpagedevice .
BeginPage	procedure (<i>semi-persistent</i>)	{pop}	The BeginPage procedure is executed at the beginning of each page. This can be any PostScript procedure.
EndPage	procedure (<i>semi-persistent</i>)	{ exch pop 2 ne }	The EndPage procedure is executed at the end of each page. This can be any PostScript procedure.
ExitJamRecovery	boolean (<i>persistent</i>)	false	Indicates whether jam recovery on the Exit path is active. A dramatic performance degradation may result when printing single page jobs if this feature is set to <i>true</i> . Entry jam recovery is always active.
ManualFeed Timeout	integer (<i>persistent</i>)	60	Indicates the number of seconds the printer waits for a job to be manually fed before cancelling the job. Only active if the ManualFeed boolean is true.
TraySwitch	boolean (<i>persistent</i>)	true	Specifies whether automatic tray switching is active when paper out conditions occur.
PostRendering Enhance	boolean (<i>volatile</i>)	true	Indicates whether product supports any type of Resolution Enhancement feature. This key is implementation independent.

¹ Install procedure is as follows: {/defaultHalftone/Halftone findresource sethalftone {} settransfer false setstrokeadjust/DefaultColorRendering /ColorRendering findresource setcolorrendering} bind

2-16 Operators and Parameters

PostRenderingEnhanceDetails dictionary

- This table describes the parameters for the **PostRenderingEnhanceDetails** dictionary. These following parameters contain details about Hewlett-Packard's specific Resolution Enhancement features. (Refer to the next section, "Resolution Enhancement (RET)," for more information.)

Key	Type	Default	Semantics
Type	integer (constant)	8	HP unique ID value
REValue	integer (volatile)	2	Specifies the current value of RET: 0 = Off, 1 = Light, 2 = Medium, 3 = Dark.

Note



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

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

Example 2-4

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

When the Page Device parameters are set during an unencapsulated Postscript job, the following values become permanently stored in NVRAM: **Duplex**, **Tumble**, **Margins**, **NumCopies**, **ManualFeed**, **ManualFeedTimeout**, and **DoJamRecovery**. Note that **REValue** is not included here.

At the start of each job, the Page Device is initialized to the values indicated by the hardware state and/or to default values.

Resolution Enhancement (RET)

A general mechanism for supporting Resolution Enhancement features is provided in Level 2 Postscript using the **PostRenderingEnhance** entry in the **pagedevice**. When this Boolean entry is set to true, the **PostRenderingEnhanceDetails** entry then specifies the type of feature provided.

The **PostRenderingEnhanceDetails** dictionary contains a **Type** entry, which is a read-only id assigned specifically for this product or group of products, and (**REValue**), a product-specific entry specifying the current value of RET. Legal values for RET are: 0, 1, 2, or 3.

At the beginning of a job, if the desired value of RET is nonzero, the **PostRenderingEnhance** key is set to true and the **PostRenderingEnhanceDetails** dictionary is set to <</Type 8 /REValue x>>, where x is the RET value. If the RET value is zero, the **PostRenderingEnhance** key is set to false and the **PostRenderingEnhanceDetails** dictionary is set to <</Type 8 /REValue 0>>.

I/O Device Operators and Parameters

I/O Device parameters are Level 2 parameters that control the operation and behavior of the PostScript Interpreter. These are referred to in the *PostScript Language Reference Manual (Second Edition)* as device parameters, but are entirely concerned with I/O devices and I/O channel parameters.

The I/O Device Parameter dictionaries containing the parameter sets for each device are described in this section. They follow a discussion of the operators that manipulate these dictionaries. Typically, I/O Device parameters are similar to system parameters. They require a password, are global to the PostScript environment, and have similar persistence characteristics.

As with system parameters, some of these parameters are stored in NVRAM. One way in which I/O parameters differ from system or user parameters is that they are interdependent; the setting of a given value for a parameter may restrict the ability to set or change another parameter. For example, in this implementation only one communications channel can be on at any given time; thus, only one device set can have On set to true.

In this implementation, all of these parameters are read-only except for **LocalTalkType** in the LocalTalk device, which can be written by way of the PostScript interpreter. Any attempt to write read-only values is ignored. The value of LocalTalkType is semi-persistent if set in an unencapsulated job. It cannot be set persistently and does not require a password. See the **appletalktype** Level 1 compatibility operator for more information.

The choice of I/O channel (Serial, Parallel, LocalTalk, or OptionalIO) can be made only from the control panel. The following Serial parameters can be set from the control panel: Baud, RobustXON, DTRHigh, and SerialMode (RS422/232 can be set on HP LaserJet III and IIID printers). All other I/O device parameters are determined by hardware or firmware design and cannot be changed.

Previously in the Level 1 implementation, it was possible to change some of the Serial parameters by way of the Level 1 operator **setscbatch**. Note that this is no longer possible. This is one area of incompatibility with the Level 1 cartridge. It has been determined that few PostScript applications, drivers, or utilities make use of this operator, so the impact of this incompatibility should be minimal. See the **setscbatch** Level 1 compatibility operator for more information.

setdevparams** *string dict setdevparams*

dict consists of the key/value pair parameters described in tables 2-4, 2-5, 2-6, and 2-7.

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

In general, *string* refers to an input/output or storage device (*%Device%*) and *dict* contains one or more key/value pairs constituting the parameter set for the device.

setdevparams Example 2-5

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

setdevparams Errors

- **invalidaccess**
- **rangecheck**
- **stackunderflow**
- **typecheck**
- **undefined**
- **configurationerror**

currentdevparams *string* **currentdevparams** *dict*

dict consists of the key/value pair parameters described in tables 2-4, 2-5, 2-6, and 2-7.

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

currentdevparams Example 2-6

(<i>%Serial%</i>) currentdevparams	% puts a copy of the dictionary containing the % Serial Parameter Sets on the stack
{ } forall pstack	% pops off of the stack all of the keys preceded by % their values and prints them out

currentdevparams Errors

- **stackoverflow**
- **VMerror**

I/O Device Parameter Tables

The four sets of key/value pairs, Serial, Parallel, LocalTalk, and OptionalIO, reside in the dictionary returned by **currentdevparams**. Writeable values are changed by a call to **setdevparams**.

Serial

There is one set of Serial parameters, all read-only, called (%Serial%), containing the key/value pairs in table 2-4:

Table 2-4. I/O Device - Serial Key/Value Pairs

Key	Type	Default	Semantics
On	boolean (read-only)	false	Contains a true or false depending on the Active I/O Channel set at the printer control panel.
Enabled	boolean (read-only)	false	The same as On. On some implementations, this key is used to indicate the presence of a pure data channel for use by an active PostScript job.
SerialMode	string (read-only)	RS232	RS232 or RS422 depending on the printer control panel or jumper setting. Some non-HP implementations have a separate parameter set for RS422.
Baud	integer (read-only)	9600	One of the following: 19200, 9600, 4800, 2400, 1200, 600, 300. Settable at printer control panel.
RobustXon	boolean (read-only)	true	true = on and false = off. Settable at the printer control panel.
DTRHigh	boolean (read-only)	true	true = high and false = low. This has meaning only when SerialMode = RS232. Settable at the printer control panel.

Table 2-4. I/O Device - Serial Key/Value Pairs (continued)

Key	Type	Default	Semantics
CheckParity	boolean (<i>read-only</i>)	false	Always is false, which indicates parity is not being checked. Can validate Parity field.
StopBits	boolean (<i>read-only</i>)	1	Always = 1.
DataBits	integer (<i>read-only</i>)	8	Always = 8.
FlowControl	string (<i>read-only</i>)	h/w and s/w	One of the following: 1) h/w 2) s/w 3) h/w and s/w. Determined by active SerialMode .
Parity	string (<i>read-only</i>)	none	Always = none since CheckParity is always false.
HasNames	boolean (<i>read-only</i>)	false	Always false. In other implementations, this is used to indicate whether this is a named device (such as a disk).
Type	name (<i>read-only</i>)	Communications	Always = Communications.

Parallel

Parallel is the default I/O. There is one set of Parallel parameters, all read-only, called (%Parallel%), containing the key-value pairs in table 2-5:

Table 2-5. I/O Device - Parallel Key/Value Pairs

Key	Type	Default	Semantics
On	boolean (read-only)	true	Contains a true or false depending on the Active I/O Channel set at the printer control panel.
Enabled	boolean (read-only)	true	The same as On. For some implementations, this key is used to indicate the presence of a pure data channel for an active PostScript job.
HasNames	boolean (read-only)	false	Always false. In other implementations, this is used to indicate whether this is a named device (such as a disk) .
Type	name (read-only)	Communications	Always = Communication.

LocalTalk

There is one set of LocalTalk parameters, called (%LocalTalk%), containing the key-value pairs in table 2-6:

Table 2-6. I/O Device - LocalTalk Key/Value Pairs

Key	Type	Default	Semantics
On	boolean (<i>read-only</i>)	false	Contains a true or false depending on the Active I/O Channel set at the printer control panel.
Enabled	boolean (<i>read-only</i>)	false	The same as On. For some implementations, this key is used to indicate the presence of a pure data channel for an active PostScript job.
LocalTalkType**	string (<i>read/writeable</i>)	LaserWriter	Maximum size of 31 characters plus null termination. This is the <i>only</i> write parameter.
NodeID	integer (<i>read-only</i>)	0	On some non-HP products may be used by AppleTalk code to cache last successfully obtained ID; should not be used by any driver or application, and will not reflect the actual NodeID value for HP LaserJet printers.
HasNames	boolean (<i>read-only</i>)	false	Always = false. In other implementations, this is used to indicate whether this is a named device (such as a disk).
Type	name (<i>read-only</i>)	Communications	Always = Communications.

OptionalIO

There is one set of Optional I/O parameters, all read-only, called (%OptionalIO%) containing the key-value pairs in table 2-7:

Table 2-7. I/O Device - OptionalIO Key/Value Pairs

Key	Type	Default	Semantics
On	boolean (<i>read-only</i>)	false	Contains a true or false depending on the Active I/O Channel set at the printer control panel.
Enabled	boolean (<i>read-only</i>)	false	Exactly the same as On. For some implementations, this key is used to indicate the presence of a pure data channel for an active PostScript job.
HasNames	boolean (<i>read-only</i>)	false	Always = false. In other implementations this is used to indicate whether this is a named device (such as a disk).
Type	name (<i>read-only</i>)	Communications	Always = Communications

Note



A standard optional I/O (XIO) board causes OptionalIO key/value pairs to be used. XIO-AppleTalk (and IIIP/Triple I/O-AppleTalk) will use the AppleTalk parameter set as does any XIO board that appears identical to the XIO-AppleTalk board.

The following key/value pairs have been omitted from all four parameter sets:

/Interpreter = PostScript. This is included in some implementations for **setsoftwareiomode** and **emulate** which were not supported on our Level 1 cartridge product.

/Protocol = Adobe Standard Protocol (ASP), Binary Communications Protocol (BCP) or Clear Binary. This key/value pair is not needed with Tagged Binary Communications Protocol (TBCP).

System Operators and Parameters

System parameters are Level 2 parameters that control the operation and behavior of the PostScript Interpreter. The **setsystemparams** and **currentsystemparams** operators manipulate the system parameters described in this section. Alterations to these parameters have a permanent, system-wide effect that may persist through restarts of the PostScript interpreter, as well as power down and restart of the printer. The persistent parameters **DoStartPage** and **DoAutoContinue** are stored in NVRAM.

System parameters can be altered only by a program that presents a valid **SystemParamsPassword** string. The **SystemParamsPassword** is itself a System parameter (write only). A user application should never attempt to alter system parameters. A system manager should alter these parameters. Those parameters that can also be changed by way of the printer control panel are so indicated.

setsystemparams*

dict **setsystemparams** -

dict consists of the key/value pair parameters described in table 2-8.

setsystemparams attempts to set one or more system parameters whose keys and new values are contained in the *dict* operand. The dictionary is a container for key-value pairs; **setsystemparams** reads the information from the dictionary, but does not retain the dictionary itself. System parameters whose keys are not referenced are left unchanged.

The dictionary must contain an entry named **Password** whose value is the system parameter password (a string or integer). If the password is incorrect, **setsystemparams** executes an **invalidaccess** error and does not alter any parameters. The **Password** entry is not required if the **SystemParamsPassword** is the default empty string (). See the *PostScript Language Reference Manual (Second Edition)* pp. 516, 571, 574 for more information. Refer to table 2-8 for operator parameters.

setsystemparams Examples

Set up an active password for the System parameters when the current password is the empty string ():

setsystemparams Example 2-7

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

A system manager might wish to persistently turn off the Startup Page, turn on Autocontinue, turn off the printing of error messages and set the host I/O wait timeout to 600 seconds. The following arguments accomplish this:

setsystemparams Example 2-8

```
<</DoStartPage false /DoAutoContinue true      % any number of parameters can
/DoPrintErrors false /WaitTimeout 600          % be changed with no affect on
                                                % other System parameters
/Password (system_password)>> setsystemparams  % note password string is now
                                                % required
```

Set up a **StartJob** password to protect the settings in the Page Device parameter set:

setsystemparams Example 2-9

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

setsystemparams Errors

- **invalidaccess**
- **stackunderflow**
- **typecheck**

currentsystemparams - **currentsystemparams** *dict*

dict consists of the key/value pair parameters described in table 2-8.

currentsystemparams returns a dictionary containing the keys and current values of all readable system parameters that are defined in the implementation. The returned dictionary is merely a container for key-value pairs. Each execution of **currentsystemparams** allocates and returns a new dictionary. See the *PostScript Language Reference Manual (Second Edition)* pp. 392, 571, 575 for more information. Refer to table 2-8 for operator parameters.

currentsystemparams Example 2-10

```
currentsystemparams /PrinterName get    % puts a copy of the Printer Name
                                         % key/value pair on the stack
==                                       % pop off of the stack the value and
                                         % displays it
```

The Interpreter returns the value of PrinterName (HP LaserJet myname).

currentsystemparams Errors

- **stackoverflow**
- **VMemory**

System Parameters Key/Value Pairs

The following System Parameters from appendix C of the *PostScript Language Reference Manual (Second Edition)* are implemented in this product. A brief description is included. For additional information, refer to the *PostScript Language Reference Manual (Second Edition)* pp. 571-579, the *PostScript Language Supplement for Version 2011* (section 3.2), and the *PostScript Language Addendum for the PostScript (Level 2) Cartridge for the Hewlett-Packard LaserJet III Family of Printers* (section 2.2):

Table 2-8. System Parameters Key/Value Pairs

Key	Type	Default	Semantics
SystemParams-Password	string (<i>persistent/ write only</i>)	empty string ()	Controls the ability of the setsystemparams operator to change the values of system parameters.
StartJobPassword	string (<i>persistent/ write only</i>)	empty string ()	Controls the ability of the startjob operator to alter initial VM.
BuildTime	integer (<i>read-only</i>)	688872737 (for final build of code)	Timestamp identifying specific build of the PostScript interpreter (Adobe 2010.118f).
ByteOrder	boolean (<i>read-only</i>)	false	Native (preferred) order of multiple-byte numbers in binary encoded tokens; false indicates high-order first; true indicates low-order first.
RealFormat	string (<i>read-only</i>)	IEEE	Native (preferred) representation for real numbers in binary encoded tokens.

Table 2-8. System Parameters Key/Value Pairs (continued)

Key	Type	Default	Semantics
MaxFontCache	integer (<i>persistent</i>)	349952	Maximum bytes occupied by the font cache. This value is dependent on the amount of optional memory installed in the printer.
CurFontCache	integer (<i>read-only</i>)	106388	Bytes currently occupied by the font cache.
MaxOutlineCache	integer (<i>semi-persistent</i>)	65536	Maximum bytes occupied by cached character outlines (CharStrings) for fonts whose definitions are kept on disk instead of VM.
CurOutlineCache	integer (<i>read-only</i>)	0	Bytes currently occupied by CharStrings .
MaxUPathCache	integer (<i>semi-persistent</i>)	300000	Maximum bytes occupied by the user path cache.
CurUPathCache	integer (<i>read-only</i>)	0	Bytes currently occupied by the user path cache.
MaxFormCache	integer (<i>semi-persistent</i>)	100000	Maximum bytes occupied by the form cache.
CurFormCache	integer (<i>read-only</i>)	0	Bytes currently occupied by the form cache.
MaxPatternCache	integer (<i>semi-persistent</i>)	100000	Maximum bytes occupied by the pattern cache.
CurPatternCache	integer (<i>read-only</i>)	0	Bytes currently occupied by the pattern cache.
MaxScreenStorage	integer (<i>semi-persistent</i>)	120000	Maximum bytes occupied by all active halftone screens, including ones created by setscreen and saved by gsave.

2-34 Operators and Parameters

Table 2-8. System Parameters Key/Value Pairs (continued)

Key	Type	Default	Semantics
CurScreenStorage	integer (<i>read-only</i>)	3978	Bytes currently occupied by all active halftone screens.
MaxDisplayList	integer (<i>semi-persistent</i>)	209715	Maximum bytes occupied by display lists, excluding those held in caches.
CurDisplayList	integer (<i>read-only</i>)	0	Bytes currently occupied by display lists.
CurInputDevice	string (<i>read-only</i>)	(%Parallel%)	Reflects the input device for the currently active communication mode. On this implementation, only one communication channel is active at any given time.
CurOutputDevice	string (<i>read-only</i>)	null ()	Reflects the output device for the currently active communication mode. This value is null for unidirectional devices such as (%Parallel%).
GenericResourceDir	string (<i>semi-persistent</i>)	(%null%)	Specifies the location in the file system for resources files.
SerialNumber	integer (<i>persistent/ read-only</i>)	none	Random number generated by the interpreter at powerup and needed for composite font support. This is only generated once at initial startup and anytime a cold reset occurs.
Revision	integer (<i>constant</i>)	2	Indicates the current revision level of the machine dependent portion of the PostScript interpreter.
MaxSourceList	integer (<i>semi-persistent</i>)	24576	Indicates the maximum number of bytes that can be utilized for source lists.

Table 2-8. System Parameters Key/Value Pairs (continued)

Key	Type	Default	Semantics
CurSourceList	integer <i>(read-only)</i>	0	Indicates the number of bytes currently occupied by source lists.
StartupMode	integer <i>(constant)</i>	0	Indicates whether the system start file (Sys/Start) or another start-up procedure should be executed when the printer is powered on. There is no alternate procedure on this product.
JobTimeout	integer <i>(persistent)</i>	0	Indicates the number of seconds a job is allowed to execute before it is aborted and a timeout error is generated. This default value may be overridden within a job by a call to setuserparams which also contains a key by this name. Range: 0 or $15 \leq x \leq 4095$. A value of 0 implies an infinite timeout. This is a product-specific parameter.
WaitTimeout	integer <i>(persistent)</i>	40	Indicates the number of seconds the printer waits to receive additional characters from the host before it aborts the current job and executes a timeout error. This default value may be overridden within a job by a call to setuserparams which also contains a key by this name. Range: 0 or $15 \leq x \leq 4095$. A value of 0 implies an infinite timeout. This is a product specific parameter.

Table 2-8. System Parameters Key/Value Pairs (continued)

Key	Type	Default	Semantics
DoStartPage	boolean (<i>persistent</i>)	true	Specifies whether a startup page is to be printed. This value can be set persistently from the printer control panel with STARTUP PAGE. This is a product-specific parameter.
DoAutoContinue	boolean (<i>semi-persistent</i>)	true	Specifies whether to continue processing when a PostScript error occurs. This value can be set persistently from the Printer control panel with AUTOCONT. If true, some printer errors are displayed for 10 seconds, and then the printer continues processing. PostScript Errors are not displayed. If false, System and PostScript Errors are displayed until the user presses the continue key. This is a product-specific parameter.
DoPrintErrors	boolean (<i>persistent</i>)	false	Specifies whether to print error pages using a built-in error handler when a PostScript error occurs. See <i>PostScript Language Reference Manual (Second Edition)</i> p.99. This is a product-specific parameter.
RamSize	integer (<i>read-only</i>)	memory configuration dependent	5242880 with maximum memory configuration. This is the maximum amount of memory that is available to the PostScript Interpreter. This is a product-specific parameter.



Table 2-8. System Parameters Key/Value Pairs (continued)

Key	Type	Default	Semantics
PageCount	integer (persistent/ read-only)	0 initially	This indicates the number of pages that have been printed while PostScript is active. PageCount is incremented (by 10) every 10 pages. PageCount is not affected by a cold reset which returns values in NVRAM to factory default values. This is a product-specific parameter.
PrinterName	string (persistent/semi-persistent)	HP LaserJet	This is the default (non-LocalTalk) name of the printer. If a LocalTalk interface is installed, there are potentially two different default printrnames, depending on which communication mode is active. (The LocalTalk name is stored in the LocalTalk board's NVRAM where all of the other Parameters are stored.) Any 31 character string will be accepted for either name for semi-persistent operation. The new (non-LocalTalk) printrname is saved persistently only if the first 11 characters are <i>HP LaserJet</i> . Note that the string must be entered exactly as shown. An additional 6 characters may be appended to this string to satisfy the criteria for persistent storage. This is a product specific-parameter.

User Operators and Parameters

User parameters are PostScript Level 2 that control the operation and behavior of the PostScript Interpreter. Like System parameters, these are concerned with printer behavior such as printer resource management, timeouts, and memory management (including garbage collection). However, unlike System parameters, User parameters only allow the programmer to modify their execution environment within the bounds of their own job, and do not affect the global default configuration of the printer. Since User parameters do not have any permanent affect on the behavior of the printer, they may be altered at will by the programmer, and do not require any special authorization such as a password.

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

The **setuserparams** and **currentuserparams** operators manipulate the User parameter sets described in table 2-9. The operators are described first and then the tables are presented.

setuserparams *dict* setuserparams -

dict consists of the key/value pair parameters described in table 2-9.

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

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

Some user parameters, such as **JobTimeOut** and **WaitTimeOut**, have default values that are system parameters with the same names. These defaults can be set by **setsystemparams**.

setuserparams Examples

A user might wish to change the value of **JobTimeOut** to limit the processing time for his PostScript job, and change the value of **WaitTimeOut** to limit the amount of time the interpreter should wait for host I/O data to be sent. These values would take affect for the duration of the user's job only. The following arguments accomplish this:

setuserparams Example 2-11

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

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

setuserparams Example 2-12

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

setuserparams Errors

- **invalidaccess**
- **stackunderflow**
- **typecheck**

currentuserparams - **currentuserparams** *dict*

dict consists of the key/value pair parameters described in table 2-9.

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

In this example **currentuserparams** retrieves the names in the User parameters dictionary:

currentuserparams Example 2-13

<code>currentuserparams</code>	<code>% put a copy of the User parameters dictionary on the stack</code>
<code>{pop=} forall</code>	<code>% pop off all of the keys (or names) in the dictionary</code>

The Interpreter returns all key values present (**MaxExecStack**, **VMThreshold**, ... **MaxDictStack**).

currentuserparams Errors

- **stackoverflow**
- **VMerror**

The User Parameters found in table 2-9 are implemented in this product. A brief description is included in table 2-9. For additional information, refer to the *PostScript Language Reference Manual (Second Edition)* pp 571-579, the *PostScript Language Supplement for Version 2011* (section 3.1), and the *PostScript Language Addendum for the PostScript (Level 2) Cartridge for the Hewlett-Packard LaserJet III Family of Printers* (section 2.1).

Table 2-9. User Parameter Key/Value Pairs

Key	Type	Default	Semantics
MaxFontItem	integer (volatile)	12500	Maximum number of bytes occupied by the pixel array of a single character in the font cache.
MinFontCompress	integer (volatile)	1250	Threshold at which a cached character is stored in compressed form instead of as a full pixel array.
MaxUPathItem	integer (volatile)	5000	Maximum bytes occupied by a single cached user path.
MaxFormItem	integer (volatile)	100000	Maximum bytes occupied by a single cached form.
MaxPatternItem	integer (volatile)	20000	Maximum bytes occupied by a single cached pattern.
MaxScreenItem	integer (volatile)	12000	Maximum bytes occupied by a single halftone screen.
MaxOpStack	integer (volatile)	100000	Maximum elements in operand stack.
MaxDictStack	integer (volatile)	530	Maximum elements in dictionary stack.
MaxExecStack	integer (volatile)	10015	Maximum elements in execution stack.

Table 2-9. User Parameter Key/Value Pairs (continued)

Key	Type	Default	Semantics
MaxLocalVM	integer (<i>read-only</i>)	2147483647	Maximum bytes occupied by values in local VM.
VMReclaim	integer (<i>volatile</i>)	0	0 = automatic garbage collection is enabled. -1 = automatic garbage collection is disabled for local VM. -2 = automatic garbage collection is disabled for both local and global VM.
VMThreshold	integer (<i>volatile</i>)	40000	Frequency of automatic garbage collection, which is triggered whenever this many bytes have been allocated since the previous collection.
JobTimeOut	integer (<i>volatile</i>)	0	Current value in seconds of JobTimeOut. This value is the only User parameter that is <i>NOT</i> subject to save and restore . This parameter is initialized to the value of the JobTimeOut System parameter at the beginning of each job. This is a product-specific parameter.
WaitTimeOut	integer (<i>volatile</i>)	0	Current value in seconds of WaitTimeOut. This parameter is initialized to the value of the WaitTimeOut System parameter at the beginning of each job. This is a product-specific parameter.
JobName	string (<i>volatile</i>)	empty string ()	Specifies the name of the current job for status responses. This is a product-specific parameter.

Note



WaitTimeOut value is always 0 when the printer is in PostScript Interactive mode, regardless of the system default value.

PostScript Level 1 and Level 2 Compatability

In general, PostScript Level 2 is compatible with PostScript Level 1. However, there are some differences. The differences between the current PostScript Cartridge and the new PostScript Cartridge Plus, both from Hewlett-Packard are:

- **Determining whether a device is monochrome or color.**

Only Level 1 devices with the color extensions have color operators like **setmykcolor** and **setcolorscreen**. All Level 2 devices have color operators, even monochrome devices. There are some applications which determine whether or not a device supports color by checking for the existence of a color operator (for example **setmykcolor known**). When such an application prints to a monochrome Level 2 device, halftone images tend to look blotchy or dark. The application sets up halftone screens with the **setcolorscreen** operator using four colors, and the printer ignores all but the black screen.

The correct way to determine whether a device is monochrome or color is:

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

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

Another way to determine whether a device is monochrome or color is to use the **DefaultColorModel* and **ColorDevice* keys in the device's PostScript Printer Description (PPD) file.

■ **Duplexing**

For duplex documents, calling any device setup Level 1 operator (or Level 2's **setpagedevice** operator) between pages will cause the previous page to be simplexed if it was an odd numbered page. Device setup Level 1 operators include all Level 1 operators equivalent to the **setpagedevice** parameters documented in Adobe's *PostScript Language Addendum for the PostScript (Level 2) Cartridge for the Hewlett-Packard LaserJet III Family of Printers*.

■ **All statusdict compatibility operators that set defaults**

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

■ **All page type operators (letter, legal, executivepage, a4, b5, com10envelope, monarcenvelope, dlenvelope, c5envelope)**

In Level 2, all page-type operators have a side effect, they initiate a search for a tray with the correct size paper to match the requested image size. This can result in paper being pulled from a different tray than the default or requested tray. If the desired tray size is not found, the policy is to find the next larger available medium, but not adjust the image size. For example, it is now more difficult to create a letter-size image on legal-sized paper. Also, all page-type operators reside in **userdict** rather than **statusdict**.

- **setdefaultduplexmode, setdefaultpapertray, setdefaulttimeouts, setdefaulttrayswitch, setdefaulttumble, and setdostartpage**

In Level 1, these operators did not take effect until the end of the job when executed in an unencapsulated job. In Level 2, they take effect immediately.

- **setdefaulttimeouts and setjobtimeout**

Larger persistent values can be set using these operators in Level 2 than in Level 1. **Manualfeedtimeout** has increased from an 8-bit value to a 16-bit value; **jobtimeout** and **waittimeout** have increased from an 8-bit value to a 12-bit value.

- **setdoidlefonts, doidlefonts, and idlefonts**

Level 2 has its own mechanism for font caching, so these operators do not affect anything. **Setdoidlefonts** pops the argument off the stack, **doidlefonts** always returns “false,” and **idlefonts** places a mark on the stack.

- **setduplexmode, setpapertray, settrayswitch, and settumble**

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

- **setscbatch/scbatch**

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



- **cexec**
ccexec is an unsupported operator in Level 1 and Level 2. In Level 2, executing it results in an *undefined* error (PS ERROR 18).
- **checkpassword**
In Level 1, **checkpassword** returns *true* if the password argument matches the Level 1 system password. In Level 2, **checkpassword** returns *true* if the password argument matches either of the Level 2 passwords.
- **gsave/grestore**
In Level 1, language operators like tray selection and duplexing set by **setpapertray** and **setduplexmode** are not included in the saved and restored graphics state. In Level 2, all of these Level 1 operators, and the **Page Device** dictionary are included in the graphics state. Even if **Page Device** dictionary is set by Level 1 operators, it is still included in the graphics state. Refer to Adobe's *PostScript Language Addendum for the PostScript (Level 2) Cartridge for the Hewlett-Packard LaserJet III Family of Printers* for a complete list of **Page Device** parameters.
- **manualfeedtimeout**
This is the only Level 1 operator in **statusdict** that is not initialized to a default value. It should never be read without first being set. The timeout algorithm for Level 2 does not use its value.
- **papertype**
In Level 1, this is an undocumented operator. In Level 2, it is no longer present and executing it results in an *undefined* error (PS ERROR 18).

- **product**

product is now found in **systemdict** as well as **statusdict**.

- **setpassword**

This operator is not supported in Level 2. It could not be mapped into the dual password architecture of Level 2.

Executing it results in an *undefined* error (PS ERROR 18).

Improved Support for Device-Specific Features in Level 2

Level 1 has device control operators defined in a special dictionary called **statusdict**. The contents of **statusdict** are product dependent, although an attempt was made to maintain consistency.

Level 2 has a new Page Device setup operator called **setpagedevice**. This operator provides a standard framework for specifying the requirements of a page description, and for controlling standard and optional features of a device. It uses a standard syntax to request features supported by all devices (such as selecting a page size) and features supported only by some devices (such as duplex printing). **setpagedevice** provides a standard mechanism for determining action when a page description makes requests the device cannot fill. **setpagedevice** makes print jobs with device-specific feature support more portable.

For an explanation of how **setpagedevice** works, see the *PostScript Language Reference Manual (Second Edition)*, especially Section 4.11. For a list of **setpagedevice** parameters (like *PageSize* and *Duplex*) supported for the PostScript Cartridge Plus, see chapter 2.

Known Problems

Hewlett-Packard wants developers and programmers to be aware of the following known problem in the PostScript Cartridge Plus:

1. The **resourceforall** operator generates an **undefined** error upon completion of its execution.
2. If duplexing is disabled for only part of a job, and if between pages another device setup operator is used before the duplexing operator, the printer may hang when the job is sent to the printer twice in a row.

The following are examples of what to do and what not to do between pages, in Level 1 and Level 2:

Level 1

Wrong

```
statusdict begin
1 setpapertray
true setduplexmode
end
```

Right

```
statusdict begin
true setduplexmode
1 setpapertray
end
```

Level 2

Wrong

```
<</Priority [1 0 2]>> setpagedevice  
<</Duplex true>> setpagedevice
```

Right

There are three possible correct ways to do it in Level 2:

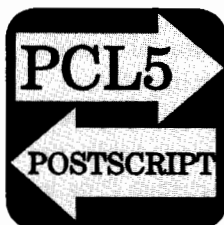
1.

```
<</Duplex true>> setpagedevice  
<</Priority [1 0 2]>> setpagedevice
```
2.

```
<</Priority [1 0 2] /Duplex true>> setpagedevice
```
3.

```
<</Duplex true /Priority [1 0 2]>> setpagedevice
```

Language Switching Sequence



Chapter 1 described the different ways users could perform printer language switching. The host-based methods (the PCL/PostScript device driver, the TSRs, and the HPNPRINT utility) involve sending a language switching sequence which tells the printer to switch to PCL, or another one which tells the printer to switch to PostScript.

The sequences appears as follows:

Switch to PCL

```
<ESC>%-12345X<9K nulls><ESC>%-12345X@PJL ENTER LANGUAGE=PCL<LF><100nulls>
```

Switch to PostScript

```
<ESC>%-12345X<9K nulls><ESC>%-123245X@PJL ENTER LANGUAGE=POSTSCRIPT<LF><100 nulls>
```

Where <ESC> represents a byte with the hex value 1B, <9K nulls> represents 9216 bytes with the hex value 00, and <LF> represents a byte with the hex value 0A.

You can find the language switching sequences on the “PCL/PostScript Device Driver and Installation” disk in the files TOPS.ASC and TOPCL.ASC.

The language switching sequences for the PostScript Cartridge Plus are not the same as those for the HP LaserJet IIISi printer. This means that the PostScript Cartridge Plus does not support PJL, and that the HP LaserJet IIISi printer’s language switching sequences will not work with the PostScript Cartridge Plus.

However, they are similar enough that the PostScript Cartridge Plus sequences work with the HP LaserJet IIISi printer. Implementing PjL in the PostScript Cartridge Plus would have been ideal, but it was not feasible because the cartridge is for existing printers that were not designed for language switching.

There is another difference between the HP LaserJet IIISi printer and the PostScript Cartridge Plus. For the HP LaserJet IIISi printer, both PostScript and PCL jobs are terminated with <ESC>%-12345X to take the printer out of PostScript or PCL mode and put it into PjL mode. In addition, the IIISi has a control panel setting (SYS=) which establishes the default language for jobs that are received without a language switching sequence while the printer is in PjL mode. The PostScript Cartridge Plus has no PjL mode. PCL jobs should never be terminated with <ESC>%-12345X. Doing so usually results in an error (PjL ERROR 2).

PostScript jobs are not terminated with <ESC>%-12345X, unless the Tagged Binary Communications Protocol (TBCP) is used. With TBCP, however, the <ESC>%-12345X serves only to end TBCP. It does not take the printer out of PostScript mode and into PjL mode. (TBCP is covered in detail in appendix B.)

Since there is no default language mechanism on the PostScript Cartridge Plus, jobs received without a language switching sequence always print in the language of the previous job. The printer's language at power-up is controlled by a control panel setting (SW MODE=), which can be set to PostScript or PCL while in PostScript mode.

Note



The SW MODE= menu item for the PostScript Cartridge Plus is not the same as the SYS= menu item for the HP LaserJet IIISi printer.

A-2 Language Switching Sequence

The Tagged Binary Communications Protocol

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

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

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

Note



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

TBCP Versus Adobe Binary Communications Protocol

TBCP was created by Adobe and HP as an alternative to Adobe Binary Communications Protocol (BCP). The two protocols are similar but not compatible. The PostScript Cartridge Plus supports TBCP, and not BCP, and errors result if BCP is sent to it. Likewise, errors result if TBCP is sent to a printer that supports only BCP.

A comparison between TBCP and BCP highlights the differences in implementation and is helpful for understanding the implementation of TBCP. Both BCP and TBCP reside in the protocol layer receiving the data, which is then passed on to the PostScript interpreter. In BCP, the binary data protocol is enabled by the interpreter. In TBCP, the binary data protocol is enabled in the I/O protocol layer, before the interpreter. In BCP, input data received immediately after the PostScript interpreter commands to enable or disable BCP may be parsed incorrectly at the I/O protocol layer because of the latency that exists between the I/O protocol layer and PostScript interpreter layer.

Therefore, mixed ASP and BCP jobs are not always handled correctly. Since TBCP controls the binary data protocol through the I/O protocol layer, it is always in sync with incoming data. Thus PostScript jobs which mix the ASP and TBCP protocol are always handled correctly. ASP jobs sent with BCP mode that rely on CR/LF folding (the conversion of a CR/LF into a LF), may not work correctly.

Implementing TBCP

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

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

where Control-A represents a byte with the hex value 01 and <ESC> represents a byte with the hex value 1B. The Control-A M enables TBCP. The <ESC>%-12345X disables TBCP and puts the printer back into ASP mode.

If you normally put a Control-D at the beginning of your jobs, it should come immediately after the Control-A M. At the end of a job, you can put the Control-D immediately before the <ESC>%-12345X. This is optional. If it is not present, the <ESC>%-12345X performs the exact same function as a Control-D in addition to disabling TBCP.

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

The following is a list of special control characters:

Table B-1. Special Control Characters

ASCII Keyboard	ASCII Name	Hex Value	Type	Control Function
Control-A	SOH	01	synchronous	Quote data character.
Control-C	ETX	03	asynchronous	Abort job and flush to end of file.
Control-D	EOT	04	synchronous	End-of-file marker.
Control-E	ENQ	05		Reserved for future use.
Control-Q	DC1	11	asynchronous	XON in XON/XOFF flow control.
Control-S	DC3	13	asynchronous	XOFF in XON/XOFF flow control.
Control-T	DC4	14	asynchronous	Job status request.
Control-[ESC	1B	synchronous	Start of end-protocol sequence.
Control-\	FS	1C		Reserved for future use.

To differentiate data bytes from special control characters, any data byte that is the same as one of the above special control characters must be quoted. A data byte is quoted by replacing it with a two-byte sequence. The first byte is a Control-A (hex 01) and the second byte is the original byte XORed with the hex value 40. For example, to send the hex value 14 as data, send the two byte hex sequence 01 54 instead.

This method of quoting guarantees that whenever any of the nine special control characters are received by the printer, the control function is intended, regardless of whether the preceding character is a Control-A. The generation and processing of asynchronous special control characters, therefore, may occur at a lower level than the generation and consumption of the data stream. In particular, on a host machine, the Control-A quoting convention may be implemented by an application driver, while XON/XOFF processing is performed independently by a lower level communications layer.

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

The following is a sample implementation of quoting PostScript data, in a C program:

```
#define UCHAR unsigned char

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

/* QuoteByte quotes a character, if necessary, using
   the Adobe Tagged Binary Communications Protocol,
   and then writes the resultant byte(s) to the printer
   by calling outbyte()
*/
```

```

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

```

After a Control-A is received, the next character received must be an "M," the result of XORing one of the special control characters with the hex value 40, or one of the asynchronous special control characters. Receipt of any other character is considered an error in the input. Between the Control-A and the XORed character, any number of asynchronous special control characters may appear. However, receipt of a synchronous special control character (Control-D or Control-A) between a Control-A and the XORed character is considered an error.

Note



The primary difference between synchronous and asynchronous special control characters is that synchronous special control characters are generated as part of the PostScript data stream and belong in particular locations within the PostScript data stream. Asynchronous special control characters are typically generated by a lower level communications layer, and appear in random locations within the PostScript data stream.

If one of the special control characters arrives unquoted, and it does not specify a control function for the channel, the character is discarded. For example, if XON or XOFF is received over a parallel channel, it is discarded.

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

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

Table B-2. Results of Special Control Characters

Byte Received by the I/O Protocol Layer	Byte Sent to the PostScript Interpreter	PostScript Interpreter Action
Control-C		PostScript interrupt, abort job
Control-T		Status query
Control-Q		Flow control - XON
Control-S		Flow control - XOFF
Control-D	End-of-file mark	
Control-E		
Control-\		
Control-[%-12345X	End-of-file mark	TBCP disabled
Control-A A	<Control-A>	
Control-A C	<Control-C>	
Control-A D	<Control-D>	
Control-A E	<Control-E>	
Control-A M		TBCP enabled
Control-A Q	<Control-Q>	
Control-A S	<Control-S>	
Control-A T	<Control-T>	
Control-A [<Control-[>	
Control-A \	<Control-\>	
Control-A <any other char>		Error
<any other char>	<char received>	

Encapsulated PostScript Files and User Interface

TBCP is a device dependent feature, and should never be used in Encapsulated PostScript (EPS) files. Hewlett-Packard and Adobe recommend allowing the user to enable or disable TBCP at the user interface level, because using TBCP gains performance but loses portability.

When presenting the TBCP option to the user, do not use the term TBCP. Allow the user to select binary communications for printing directly to the printer, or Hexidecimal ASCII when printing to a file which may be sent to another printer. The selection may be made in an area titled "Performance Options."

Index

A

- administrators' tips, 1-1
- Adobe Standard Protocol, B-1
- any feature name*, 2-14
- ASP, B-1
- AutoContinue, 1-15
 - procedure, 1-16
- AutoContinue and DoPrintErrors combinations, 1-18

B

- Baud, 2-23
- BeginPage, 2-16
- BuildTime, 2-33
- ByteOrder, 2-33

C

- categories of parameter set values, 1-6
- cexec, 3-4
- changing System parameters, 2-29
- changing Serial parameters, 2-20
- chapter 2 format, 2-3
- CheckParity, 2-24
- checkpassword, 3-4
- cold reset, 1-10
- compatibility between Level 1 and Level 2, 3-1
- control panel settings' defaults, 1-12
- #copies, 2-14
- COPIES, 2-14
- CurDisplayList, 2-35

- CurFontCache, 2-34
- CurInputDevice, 2-35
- CurOutputDevice, 2-35
- currentdevparams, 2-22
- currentdevparams errors, 2-22
- currentdevparams example, 2-22
- currentdevparms
 - factory defaults, 1-11
- currentpagedevice, 2-8
 - factory defaults, 1-11
- currentpagedevice errors, 2-8
- currentpagedevice example, 2-8
- currentpagedevice subdictionaries, 2-9
- currentsystemparams
 - factory defaults, 1-11
- currentsystemparams definition, 2-32
- currentsystemparams errors, 2-32
- currentsystemparams example, 2-32
- currentsystemparams operator, 2-32
- currentuserparams definition, 2-42
- currentuserparams errors, 2-42
- currentuserparams example, 2-42
- currentuserparams operator, 2-42
- CurSourceList, 2-36
- CutFormCache, 2-34
- CutOutlineCache, 2-34
- CutPatternCache, 2-34
- CutScreenStorage, 2-34
- CutUPathCache, 2-34

D

- DataBits, 2-24
- default I/O, 2-25
- definition
 - currentdevparams, 2-22
 - currentpagedevice, 2-8
 - setdevparams, 2-21
 - setpagedevice, 2-6
- description
 - operator errors, 2-4
 - operator example, 2-4
 - operator name, 2-3, 2-4
 - parameter tables, 2-5
- determining a color device, 3-1
- determining a monochrome device, 3-1
- device-specific features, 3-6
- DoAutoContinue, 2-37
- doidlefonts, 3-3
- DoPrintErrors, 2-37
 - enabling, 1-17
- DoPrintErrors and AutoContinue combinations, 1-18
- DOS COPY command, 1-3
- DoStartPage, 2-37
- DTRHigh, 2-23
- Duplex, 2-13
- duplexing
 - known problems, 3-7
- duplexing in Level 2, 3-2

E

- Enabled (LocalTalk parameter), 2-26
- Enabled (OptionalIO parameter), 2-27
- Enabled (Parallel parameter), 2-25
- Enabled (Serial parameter), 2-23
- enabling DoPrintErrors, 1-17
- ending Level 2 programs, 1-3
- EndPage, 2-16
- error handling procedures, 1-3

Index-2

errors

- currentdevparams, 2-22
- currentpagedevice, 2-8
- currentsystemparams, 2-32
- currentuserparams, 2-42
- setdevparams, 2-21
- setpagedevice, 2-8
- setsystemparams, 2-31
- setuserparams, 2-41

examples

- currentdevparams, 2-22
- currentpagedevice, 2-8
- currentsystemparams, 2-32
- currentuserparams, 2-42
- setdevparams, 2-21
- setpagedevice, 2-7
- setsystemparams, 2-30
- setuserparams, 2-41

ExitJamRecovery, 2-16

F

- factory default settings, 1-11
- FlowControl, 2-24

G

- GenericResourceDir, 2-35
- gsave/grestore, 3-4

H

- HasNames (LocalTalk parameter), 2-26
- HasNames (OptionalIO parameter), 2-27
- HasNames (Parallel parameter), 2-25
- HasNames (Serial parameter), 2-24
- HP LaserJet IIISi
 - job termination, A-2
 - language switching sequence, A-1
- HWResolution, 2-13

I

- idlefonts, 3-3
- ImagingBBox, 2-13
- implementing TBCP, B-4
- including comments (%), 1-3
- InputAttributes dictionary, 2-10
 - Priority array, 2-12
 - Tray 0 subdictionary, 2-11
 - Tray 1 subdictionary, 2-11
 - Tray 2 subdictionary, 2-12
- InputAttributes subdictionaries, 2-9
- Input Media Selection, 2-10
- Install, 2-16
- Interpreter sub-categories, 1-5
- I/O channel, 2-20
- I/O Device operators and parameters, 2-19
- I/O Device parameter tables, 2-23

J

- JobName, 2-44
- JobTimeout (System parameter), 2-36
- JobTimeOut (User parameter), 2-44

K

- key/value pairs
 - Page Device, 2-10
- known problems, 3-7
 - duplexing, 3-7
 - resourceforall, 3-7

L

- language switching
 - DOS batch files, 1-1
 - HPNPRINT, 1-1
 - HP PCL/PostScript device driver, 1-1
 - printer control panel, 1-1, 1-2
 - Terminate-and-Stay Resident program, 1-1
 - language switching sequence, A-1
 - Level 2
 - device-specific features, 3-6
 - Level 2 Operators, 2-1
 - Level 2 Operators and Parameters, 2-1
 - LocalTalk parameters, 2-26
 - LocalTalkType, 2-26

M

- ManualFeed, 2-10
- manualfeedtimeout, 3-4
- ManualFeedTimeout, 2-17
- Margins, 2-13
- MaxDictStack, 2-43
- MaxDisplayList, 2-35
- MaxExecStack, 2-43
- MaxFontCache, 2-34
- MaxFontItem, 2-43
- MaxFormCache, 2-34
- MaxFormItem, 2-43
- MaxLocalVM, 2-44
- MaxOpStack, 2-43
- MaxOutlineCache, 2-34
- MaxPatternCache, 2-34
- MaxPatternItem, 2-43
- MaxScreenItem, 2-43
- MaxScreenStorage, 2-34
- MaxSourceList, 2-35
- MaxUPathCache, 2-34
- MaxUPathItem, 2-43
- MediaColor, 2-10
- MediaType, 2-10
- MediaWeight, 2-10
- MinFontCompress, 2-43
- mixed ASP and BCP jobs, B-3
- mixed ASP and TBCP jobs, B-3
- modifying PostScript language parameters, 1-8
- multiple copies, 2-14

N

NodeID, 2-26
NumCopies, 2-13, 2-14

O

omitted key/value pairs, 2-28
On (LocalTalk parameter), 2-26
On (OptionalIO parameter), 2-27
On (Parallel parameter), 2-25
On (Serial parameter), 2-23
operator description, 2-4
operator errors
 description, 2-4
operator example
 description, 2-4
operator name
 description, 2-3
operators
 currentdevparams, 2-22
 currentpagedevice, 2-8
 currentsystemparams, 2-32
 currentuserparams, 2-42
 setdevparams, 2-21
 setpagedevice, 2-6
 setsystemparams, 2-30
 setuserparams, 2-40
Operators and Parameters
 Introduction, 2-1
operator syntax
 description, 2-4
OptionalIO parameters, 2-27
OutputFaceUp, 2-13
OutputPage, 2-13

P

PageCount, 2-38
Page Device
 Input Media Selection, 2-10
 Policy and Special Action Entries,
 2-14

Processing and Output Entries,
 2-13

Page Device key/value pairs, 2-10
Page Device operators, 2-6
Page Device parameters, 2-6
PageSize, 2-10, 2-14
page type operators, 3-2
papertype, 3-4
Parallel parameters, 2-25
parameter settings
 modifying, 1-8
 viewing, 1-7
parameter tables
 description, 2-5
Parity, 2-24
performing a cold reset, 1-10
persistent definition, 1-6
persistent/read-only definition, 1-6
Policies, 2-14
 setting, 2-18
Policy and Special Action Entries,
 2-14
PolicyNotFound, 2-14
PolicyReport, 2-14
PostRenderingEnhance, 2-16
PostRenderingEnhanceDetails
 Type, 2-17
PostScript error messages, 1-19-23
PostScript Language Parameters
 Illustration, 2-2
PostScript Level 2 general categories,
 1-5
PostScript self test, 1-7
PostScript test page defaults, 1-13
PrinterName, 2-38
printing a PostScript self test, 1-7
Priority array, 2-12
problems that are known, 3-7
Processing and Output Entries, 2-13
product, 3-5
program structuring conventions, 1-4

PS ERROR 00, 1-19
PS ERROR 01, 1-19
PS ERROR 02, 1-19
PS ERROR 03, 1-19
PS ERROR 04, 1-20
PS ERROR 05, 1-20
PS ERROR 06, 1-20
PS ERROR 07, 1-20
PS ERROR 08, 1-20
PS ERROR 09, 1-20
PS ERROR 10, 1-20
PS ERROR 11, 1-21
PS ERROR 12, 1-21
PS ERROR 13, 1-21
PS ERROR 14, 1-21
PS ERROR 15, 1-21
PS ERROR 16, 1-21
PS ERROR 17, 1-22
PS ERROR 18, 1-22
PS ERROR 19, 1-22
PS ERROR 20, 1-22
PS ERROR 21, 1-22
PS ERROR 22, 1-23
PS ERROR 23, 1-23
PS ERROR 24, 1-23
PS ERROR 25, 1-23
PS ERROR 26, 1-23
PS ERROR 27, 1-23
PS ERROR 28, 1-23

R

RamSize, 2-37
read-only definition, 1-6
RealFormat, 2-33
reporting PostScript errors, 1-15
resetting to factory defaults, 1-10
Resolution Enhancement, 2-18
resourceforall
 known problems, 3-7
results of special control characters,
 B-8

RET, 2-18
REValue, 2-17
Revision, 2-35
RobustXon, 2-23

S

saving Level 2 programs, 1-3
selecting TBCP to print, B-10
semi-persistent definition, 1-6
SerialMode, 2-23
SerialNumber, 2-35
Serial parameters, 2-23
setdefaultduplexmode, 3-3
setdefaultpapertray, 3-3
setdefaulttimeouts, 3-3
setdefaulttrayswitch, 3-3
setdefaulttumble, 3-3
setdevparams, 2-21
 factory defaults, 1-11
setdevparams errors, 2-21
setdevparams example, 2-21
setdoidlefonts, 3-3
setdostartpage, 3-3
setduplexmode, 3-3
setjobtimeout, 3-3
setpagedevice, 2-6
 factory defaults, 1-11
setpagedevice errors, 2-8
setpagedevice examples, 2-7
setpapertray, 3-3
setpassword, 3-5
setsccbatch, 2-20
setsccbatch/sccbatch, 3-3
setsystemparams
 factory defaults, 1-11
setsystemparams definition, 2-30
setsystemparams errors, 2-31
setsystemparams examples, 2-30
setsystemparams operator, 2-30
setting AutoContinue, 1-15
setting DoPrintErrors, 1-17

- setting PostScript language
 - parameters, 1-5
- settrayswitch, 3-3
- setttumble, 3-3
- setuserparams definition, 2-40
- setuserparams errors, 2-41
- setuserparams examples, 2-41
- setuserparams operator, 2-40
- StartJobPassword, 2-33
- StartupMode, 2-36
- statusdict compatibility operators,
 - 3-2
- StopBits, 2-24
- storing I/O Device parameters, 2-19
- storing Page Device parameters, 2-18
- subdictionaries
 - currentpagedevice, 2-9
 - InputAttributes, 2-9
- switching between Level 1 and Level
 - 2 cartridges, 1-14
- System operators and parameters,
 - 2-29
- System parameter tables, 2-33
- SystemParamsPassword, 2-33
- system passwords, 1-3

T

- Tagged Binary Communications
 - Protocol
 - definition, B-1
 - job termination, A-2
 - special control characters, B-5
- TBCP
 - definition, B-1
 - Encapsulated PostScript files, B-10
 - job termination, A-2
 - selecting to print, B-10
 - special control characters, B-5

- TBCP versus BCP, B-3
- tips for administrators, 1-1
- Tray 0 definition, 2-9
- Tray 0 subdictionary, 2-11
- Tray 1 definition, 2-9
- Tray 1 subdictionary, 2-11
- Tray 2 definition, 2-9
- Tray 2 subdictionary, 2-12
- TraySwitch, 2-17
- troubleshooting tips, 1-14
- Tumble, 2-13
- Type (LocalTalk parameter), 2-26
- Type (OptionalIO parameter), 2-27
- Type (Parallel parameter), 2-25
- Type (Serial parameter), 2-24

U

- upgrading from a Level 1 cartridge,
 - 1-14
- User operators and parameters, 2-39
- User parameter tables, 2-43

V

- value types, 1-6
- viewing errors, 1-15
- viewing parameter settings, 1-7
- VMReclaim, 2-44
- VMThreshold, 2-44
- volatile definition, 1-6

W

- WaitTimeout (System parameter),
 - 2-36
- WaitTimeOut (User parameter), 2-44
- ways to language switch, 1-1
- ways to view parameter settings, 1-7
- write-only definition, 1-6
- writing Level 2 programs, 1-3