
Xerox DocuPrint 180 Laser Printing System Operations Reference

**THE DOCUMENT COMPANY
XEROX**

April 1998
721P85500

Xerox Corporation
701 S. Aviation Boulevard
El Segundo, CA 90245

©1998 by Xerox Corporation. All rights reserved.

Copyright protection claimed includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including without limitation, material generated from the software programs which are displayed on the screen, such as icons, screen displays, looks, etc.

Printed in the United States of America.

Publication number: 721P85500

Xerox® and all Xerox products mentioned in this publication are trademarks of Xerox Corporation. Products and trademarks of other companies are also acknowledged.

Changes are periodically made to this document. Changes, technical inaccuracies, and typographic errors will be corrected in subsequent editions.

This document was created on a PC using Frame software. The typeface used is Helvetica.

Related publications

The *Xerox DocuPrint 180 Laser Printing System Operations Reference* is part of the ten manual reference set for your laser printing system. The entire reference set is listed in the table below. Several other related documents are also listed for your convenience. For a complete list and description of available Xerox documentation, refer to the Xerox Documentation Catalog (Publication number 610P17417) or call the Xerox Documentation and Software Services (XDSS) at 1-800-327-9753.

Table 1. **Related Publications**

Publication	Number
<i>Xerox DocuPrint 180 Laser Printing System Operator Guide</i>	721P85490
<i>Xerox DocuPrint 180 Laser Printing System Operations Reference</i>	721P85500
<i>Xerox DocuPrint 180 Laser Printing System Message Guide</i>	721P85550
<i>Xerox DocuPrint 180 Laser Printing System PDL Reference</i>	721P85530
<i>Xerox DocuPrint 180 Laser Printing System Forms Creation Guide</i>	721P85520
<i>Xerox DocuPrint 180 Laser Printing System System Generation Guide</i>	721P85510
<i>Xerox DocuPrint 180 Laser Printing System Installation Planning Guide</i>	721P85480
<i>Xerox DocuPrint 180 Laser Printing Operator Command Summary Card</i>	721P85560
<i>Xerox DocuPrint 180 Laser Printing PC User Interface Reference</i>	721P85540
<i>Xerox DocuPrint 180 Laser Printing Product Guide</i>	721P85570
<i>Xerox Laser Printing Systems Tape Formats Manual</i>	600P86175
<i>Xerox Laser Printing Systems Standard Font Library Font User Guide</i>	600P86174
<i>Helpful Facts About Paper</i>	721P82492

Notice

This publication may contain descriptions of concepts and features not currently available for your Xerox Laser Printing System. Consult your Xerox sales representative or your operating system software program description for additional information.



Table of Contents

Related publications	iii
Notice	iii
Introduction	xxi
About the reference set	xxi
Xerox DocuPrint 180 Laser Printing System document set	xxi
Who should use this manual	xxiv
LPS system administrators, programmers, operators	xxiv
Xerox DocuPrint 180 LPS	xxv
Xerox DocuPrint 180 system controller components	xxvi
Xerox DocuPrint 180 tape and cartridge devices	xxvii
Xerox DocuPrint 180 printer components	xxviii
How to use this manual	xxix
If the LPS is new to you	xxix
Know the key terms and conventions	xxix
Use the table of contents, index, and glossary	xxix
Refer to the related publications list	xxix
Key terms and conventions	xxx
Key terms	xxx
Syntax conventions	xxx
Masking conventions	xxxi
Contents of this manual	xxxii
Statement command summaries and appendices	xxxiii
1. Setting LPS parameters	1-1
Communicating with the LPS	1-1
Installation and system generation	1-3
User interface PC UI	1-3
Powering on and booting the LPS	1-4
PC UI	1-4
Power loss	1-4
System verification messages	1-5
Printer communication link	1-6
PSC (printer subsystem controller)	1-8
Configure a finishing or feeding device	1-8
Create the FCG.LIB file	1-8
FCG (display the profile labels)	1-12

	FCG (configure a finishing or feeding device)	1-13
	FCG SHOW (check finishing or feeding device settings)	1-13
	FCG procedures	1-14
Verifying LPS status		1-15
	Verifying system version and time	1-15
	OCS (system version display)	1-15
	Time and date displays	1-15
	Verifying system status	1-16
	PROBLEM	1-16
Setting LPS parameters		1-18
	Setting system parameters	1-18
	SETTIME (set time or date)	1-18
	ALIGN (set page print alignment)	1-19
	DISPLAY (set display format for messages)	1-20
	ENERGY	1-21
Setting parameters for print jobs		1-22
	Reallocating space to the print file	1-22
	REA (reallocate print file)	1-22
	REA and disk management	1-25
	REA and online report spooling	1-25
	Setting job parameters	1-26
	FONTS	1-26
	FORMS	1-27
	GRAPHIC	1-27
	Adjusting print image darkness	1-28
	DARKNESS	1-28
	Lightening high-density images	1-29
	EDGE	1-29
	Finishing Exerciser Utility	1-30
	FEX	1-30
Setting file access and task execution privileges		1-31
	Protecting access to files	1-31
	LOGON (restrict file access)	1-31
	SECURE (restrict file access)	1-32
	Secured files, access, and manipulation	1-32
	Security breach	1-33
	Limiting task execution privileges	1-34
	RCU (restrict command usage)	1-34

2. Managing LPS resources	2-1
Maintaining optimal disk usage	2-1
Files on system disks	2-1
Operating on system files	2-1
Necessary free space on system disks	2-3
File biasing by disk configuration	2-3
Checking usage of disk space	2-6
FCHECK (file check)	2-6
Reorganizing usage of disk space	2-9
COMPRESS	2-9
PURGE	2-11
Backing up and copying system files	2-12
Setting default tape device	2-12
Identify current TAPE & CARTRIDGE device (SUB DEV)	2-12
SUBSTITUTE DEVICE (general information)	2-12
Assign a controller device to TAPE or CARTRIDGE (SUB DEV)	2-14
Know the Xerox tape and cartridge format	2-16
Prepare the tape or cartridge	2-16
TAPE [CARTRIDGE TDn] VOLINIT	2-16
Copying files	2-17
COPY (disk to disk)	2-17
COPY (disk to labeled tape or cartridge)	2-18
COPY (disk to unlabeled tape)	2-19
COPY (labeled tape or cartridge to disk)	2-20
COPY (unlabeled tape to disk)	2-22
Manipulating files	2-23
DELETE	2-23
FID (build file list)	2-24
FILE (display file list)	2-24
FIX (modify font or logo file)	2-28
REVIEW (review files)	2-28
RNAME (rename a file or directory)	2-29
TYPE	2-29
List tape or cartridge files	2-30
TAPE [CARTRIDGE TDn] LIST	2-30
Locate tape or cartridge files	2-32
TAPE [CARTRIDGE TDn] FIND (find file)	2-32
TAPE [CARTRIDGE TDn] NEXT (move to next file)	2-32

TAPE [CARTRIDGE TDn] SKIP (skip to file)	2-33
TAPE [CARTRIDGE TDn] EOF or NOEOF or ENDFILE	2-34
Backing up system files to tape or cartridge	2-36
DSR (disk save and restore)	2-36
Starting DSR	2-36
File protection	2-38
Disk space for restore	2-38
Multivolume files	2-39
Device error handling	2-39
Aborting the save process	2-39
Incomplete restoration	2-39
Aborting the restore process	2-40
Special file handling	2-40
Copying a DSR file to the LPS	2-41
SFT (single file transfer utility)	2-41
Backing up disk files to PC UI	2-44
Know LPS disk file attributes	2-44
Convert the console log	2-44
Convert files with supported formats	2-44
Delete LPS or PC UI floppy files	2-44
Copy LPS or PC UI floppy files	2-45
List LPS or PC UI floppy files	2-45
Transfer files between the LPS and PC UI	2-45
Managing files on floppy disks	2-46
FLOPPY FORMAT	2-46
FLOPPY SAVE	2-47
FLOPPY DELETE	2-48
FLOPPY CLEAR	2-49
FLOPPY FILE	2-49
FLOPPY FCHECK	2-50
FLOPPY RESTORE	2-51
FLOPPY LIST	2-52
Copying the system software tape	2-53
SST (system software tape or cartridge)	2-53
3. Tracking system activity	3-1
Accounting data in the status file	3-1
How the system adds data to the status file	3-1
Status file creation during sysgen	3-2
Backing up status file information	3-2

Status file data in the online accounting page	3-4
Status file data	3-5
DATE entry	3-5
DEPARTMENT entry	3-5
JOB ID entry	3-5
REPORT NO. entry	3-5
FILE ID entry	3-5
INPUT PROCESSING TIME entry	3-5
OUTPUT PROCESSING TIME entry	3-5
Report completion codes	3-6
PAGES TO BIN entry	3-7
PAGES TO TRAY entry	3-7
PAPER PATH HOLES entry	3-7
GRAPHIC PAGES PRINTED entry	3-7
LINES PRINTED entry	3-8
GRAPHIC EXCEPTION CODE entry	3-8
ONLINE IDLE (SEC) entry	3-8
TAPE MOUNTS entry	3-8
BLOCKS READ entry	3-8
BLOCKS SKIPPED entry	3-8
RECORDS READ entry	3-8
DJDE RECORDS READ entry	3-8
MAXIMUM COPY COUNT entry	3-8
OVERPRINTS entry	3-8
COLLATE entry	3-9
SF/MF entry	3-9
SIMPLEX/DUPLEX entry	3-9
JDE, JDL USED entry	3-9
ACCTINFO entry	3-9
INITIAL FONT LIST entry	3-9
INITIAL FORM LIST entry	3-9
INITIAL CME LIST entry	3-10
Using accounting and system activity commands	3-10
Generating system activity reports	3-10
REPORT	3-10
Establishing accounts and copying account data to tape	3-14
ACCOUNT	3-14
Tape format of data written by ACCOUNT	3-16

Modifying the status file	3-18
SFS (status file services)	3-18
SFS CLEAR	3-19
SFS CREATE	3-20
SFS DISPLAY	3-20
SFS file-name	3-23
SFS HARDCOPY	3-24
SFS HOST	3-26
SFS TAPE	3-27
SFS TDn	3-28
SFS ?	3-28
SFS !	3-28
Creating the SFS user form file	3-29
Tape format of data written by SFS	3-31
Structure of an SFS report entry	3-34
Forward and backward pointers	3-35
Dumping the status file to tape	3-35
Audit Log	3-36
Waste Management using the Audit Log Utility	3-36
Audit Log Utility	3-36
Audit Log Window	3-36
Audit Log Print Options Window	3-37
Audit Log Display Options Window	3-45
Audit Log Display Window	3-45
Audit Log Save Window	3-45
Audit Log Search Window	3-46
Reconciliation Error window	3-46
Sequence Error window	3-46
Stale Date window	3-47
Audit Log Constraints/Exceptions	3-47
Audit Log Administration Utility	3-48
Audit Log Administration window	3-48
Audit Log Default Options window	3-48
Audit Log Disk Usage window	3-48
Disk Approaching Full	3-48

4. Editing source files	4-1
What the editor is	4-1
How the editor works	4-1
Source files	4-1
Viewing source files	4-2
Storing source files	4-2
Files you can edit	4-2
Editor commands	4-4
Three types of editor commands	4-4
Composite editor commands	4-4
Editor syntax conventions	4-5
Using the <TAB> key	4-6
Stack printing	4-6
Running an editing session	4-7
EDIT	4-7
CONVERT	4-8
NOCONVERT	4-8
GET	4-8
KEYS	4-9
MERGE	4-9
SAVE	4-10
CLEAR	4-10
END	4-11
Modifying records	4-12
DISPLAY (display lines)	4-12
DUPLICATE (copy lines)	4-13
FIND (find lines with match)	4-13
INSERT (insert lines)	4-14
MODIFY (modify lines)	4-14
MOVE (move lines)	4-15
PRINT (print lines)	4-15
REMOVE (delete lines)	4-16
RENUMBER (renumber lines)	4-16
REPLACE (substitute lines)	4-17
SORT (sort lines)	4-17
STEP (display next line)	4-18
Modifying parts of records	4-18
C (change string)	4-18
D (delete string)	4-19
P (insert prior)	4-20

	F (insert following)	4-20
	O (overlay string)	4-20
	S (substitute string)	4-21
	Examples of editor sessions	4-22
	Create and save a .JSL file	4-22
	Modify and resave the .JSL file	4-23
	List other .JSL files and insert one into the .JSL file	4-24
5. Using command files		5-1
Command files		5-1
System, editor, and font editor command files		5-1
Using system command files		5-2
Operator entries during execution		5-2
Error processing		5-2
.PAUSE processing		5-2
.WAIT processing		5-3
Using editor command files		5-4
Points to note		5-4
Using font editor command files		5-4
Verifying command file type		5-5
.BCP		5-5
Commands in command files		5-6
Displaying messages		5-6
Documenting the command file		5-6
Summary table of command file commands		5-7
Using the basic command file commands		5-8
.CHAIN		5-8
.DELAY		5-8
.EXIT		5-8
.HOME		5-9
.PAUSE		5-9
.WAIT		5-9
Using the logical processing command file commands		5-10
Using symbol setting commands		5-10
Using string substitution commands		5-11
Using sequence control commands		5-12
Using debugging commands		5-12
Sample command files		5-13
Editor command file to delete files		5-13
Sample verification of command file type		5-14

6. Editing font files	6-1
About the font editor	6-1
Reference documentation for fonts	6-1
Font editor	6-1
How to use the font editor	6-2
Summary table of font editor commands	6-3
Running a font editing session	6-4
Messages during the session	6-4
Starting a font editor session	6-4
Reordering a font file	6-5
REORDER and FEDIT	6-5
Specifying the primary source font file	6-6
FEDIT	6-6
INPUT	6-6
Selecting characters using the correct mode	6-8
Specifying new font file characters	6-8
INCLUDE	6-8
SUBSTITUTE	6-9
RECODE	6-10
Using a secondary source font file	6-11
OPEN	6-11
CLOSE	6-12
Closing font file without using CLOSE	6-12
Saving font files you create	6-12
OUTPUT	6-12
Ending the font editing session	6-13
END	6-13
7. LPS print processing	7-1
Overview	7-1
Laser printing systems (LPS)	7-2
LPS component types	7-2
LPS hardware components	7-2
Advanced imaging subsystem (AIS)	7-2
LPS software components	7-3
Three LPS user roles	7-6
System administrator role	7-6
Programmer role	7-6
Operator role	7-7
Key terms in job processing	7-8

Job flow overview	7-8
Input sources	7-8
Job flow	7-8
Preparing for a print job	7-8
Job flow steps	7-10
Input processing	7-13
Output processing	7-15
Operator interaction with input and output processing	7-16
LPS processing status	7-18
Types of LPS processing	7-20
Utility, command, compiler, and editor processing	7-20
Running utilities	7-20
Executing command files	7-20
Running compilers	7-21
Running editors	7-22
Online printing systems	7-22
Channel-attached LPS	7-22
System generation	7-22
Online 3211/4245 mode	7-23
Print description language (PDL) statements	7-23
Copy-sensitive copy modification entries (CMEs)	7-24
Report separation	7-24
Universal character set buffers (UCSBs)	7-25
Forms control buffer (FCB)	7-26
Online record length	7-27
Points to note	7-27
Operating the system	7-29
Placing the system online or offline	7-29
Starting an online job	7-29
Monitoring online operations	7-29
Printing all completed pages	7-30
Ending an online job	7-30
Multicopy disk saturation	7-30
Handling MOUNT FORM message	7-32
Host-induced idle time	7-33
Online recovery	7-33
Online dump	7-34
Starting and ending dump sessions	7-34
Dump format	7-34
Points to note	7-35

Downloading files from the host to the LPS	7-36
Valid download file types	7-36
HOSTCOPY	7-36
Card-image format files	7-36
Font and logo tape file format	7-37
File transmission	7-38
DJDE FILE	7-39
Card-image file processing	7-40
LPS-labeled file processing	7-41
Delimited records for LPS-labeled files	7-42
Small print files and online report control	7-42
LEASH	7-42

8. HIP processing on the LPS 8-1

Initiating communications	8-1
Operator commands	8-2
HIP	8-2
HIP ABORT	8-3
HIP CONCATENATE	8-4
HIP CHANGE	8-5
HIP DRAIN	8-5
HIP END	8-6
HIP EXIT	8-6
HIP HOST	8-6
HIP JOBS	8-7
HIP ONLINE	8-7
HIP RESET	8-8
HIP SHOW BUFFER	8-8
HIP SHOW HOSTS	8-8
HIP SHOW JOBS	8-9
HIP SHOW VERSION	8-13
HIP START	8-14
HIP STOP	8-15
HIP library commands	8-16
ACCEPT	8-17
ACFILES	8-17
ACTREPORTS	8-18
BLOCKSIZE	8-18
BUFFERSIZE	8-19
CONCATENATE	8-20
DRAINTIMER	8-21

	HOST	8-21
	LOGENTRIES	8-22
	NAME	8-22
	QUEUESIZE	8-23
	STARTCOMMANDS	8-24
	XPFJOB	8-25
	XPSJOB	8-25
	Nonprint transfer file control	8-26
	File recovery	8-26
9.	Graphics on the LPS	9-1
	Input for graphics	9-1
	Processing modes	9-1
	Random mode	9-2
	Move mode	9-2
	Block mode	9-2
	Batch mode	9-2
	Tape formats	9-3
	Noninterleaved	9-3
	Document interleaved	9-3
	Page interleaved	9-4
	Batch mode	9-4
	Online formats	9-4
	Noninterleaved	9-4
	Document and page interleaved	9-4
	Batch mode	9-5
	Document interleaved graphic file transfers	9-5
	Management of image files	9-5
	Scaling	9-6
	PDL graphics options	9-7
	FDL GRAPHIC statement	9-8
	Operational considerations	9-9
	Recovery	9-9
	Accounting	9-9
	Graphic messages	9-10
	Disk saturation	9-11
	Imaging dump utility	9-11
	Image Dump or Print	9-11
	Image Fix	9-16

Performance	9-18
Random mode	9-18
Online	9-18
Document interleaved file creation	9-18
Restrictions	9-19
Graphic feature restrictions	9-19
10. Setting up print jobs	10-1
Compare V3C with other Xerox software versions	10-1
Compare job processing	10-1
Identify how print jobs run on V3C	10-1
Compare file compilation	10-2
Compare OSS command execution	10-2
Identify how V3C handles other OSS commands	10-2
Converting Xerox DocuPrint 180 files to print on the 4650 LPS	10-3
SCALE	10-3
Controlling stock input to print jobs	10-4
High-capacity feeder (HCF)	10-4
What clusters are	10-4
Cluster features	10-4
Cluster processing overview	10-5
Cluster classification	10-6
What clusters do for the programmer and operator	10-6
Where clusters are stored	10-7
How applications use clusters	10-8
Simple and OTEXT applications	10-8
Stockset applications	10-8
Mixing applications	10-9
Defining clusters and stocksets with PDL and DJDE	10-10
Basic considerations	10-10
Steps for creating clusters	10-11
RFEED (record feed)	10-13
Keeping stockset changes to a minimum	10-14
Using clusters with ordered or tab stocks	10-14
Cluster commands	10-15
Using cluster commands	10-16
Cluster command examples	10-18
Keeping clusters in the database	10-18
Operator tasks using clusters	10-18
Irregular screen displays	10-19
Changing stock in trays	10-19

Sample operator procedure	10-20
A current tray in two clusters	10-21
Assigning the input trays with FEED	10-22
FEED (select input tray)	10-22
The FEED override sequence	10-24
Assigning an input tray for purge sheets	10-24
PFEED (purge feed)	10-24
Using stocks of varying sizes	10-26
Printing system pages	10-26
Supported paper sizes	10-26
SET CONTAINER (changing the size of a container)	10-27
SET TRAY (changing paper size for trays)	10-28
Feeding paper from the short edge	10-30
Feeding sequenced stocks	10-31
OSTK (ordered stock)	10-31
Setting up output from print jobs	10-32
Stock size and stacker bins	10-32
Assigning the stacker bins	10-32
SELECT	10-32
BLIMIT (bin limit)	10-33
SEPARATE	10-34
UNLOAD (lowering trays and bins)	10-34
BINFULL (setting maximum output to bin)	10-34
TRANS (running transparency jobs)	10-35
Backward compatibility	10-35
11. Running print jobs	11-1
Printing samples of system files	11-1
SAMPLE (during a print job)	11-2
SAMPLE (form, font, or logo)	11-3
SAMPLE (multiple copies of a form)	11-5
SAMPLE (extended .IMG options)	11-6
HARDCOPY	11-10
Controlling offline input for print jobs	11-15
SPACE (position tape or cartridge by report)	11-15
SPACE (position report printing by page)	11-16
MOVE (position tape or cartridge by block or file)	11-18
Rewind a tape or cartridge	11-19
Unload a tape or cartridge	11-19

Controlling online input for print jobs	11-20
ONLINE	11-20
DRAIN (print completed pages)	11-20
ENDJOB (end online job)	11-20
OFFLINE	11-20
Printing user files	11-21
Controlling print processing	11-21
START (start a print job)	11-21
START (print a tape or cartridge dump)	11-25
STOP (stop processing)	11-26
CONTINUE (resume printing)	11-27
ABORT (abort job)	11-28
LOCKOUT	11-30
RESET (stop all system processing)	11-31
CLEAR	11-31
FHI (fault, hint, information)	11-31
Checking print job status	11-32
Using the JOBS command	11-32
JOBS (job queue/report status)	11-32
JOBS ALL options	11-36
NULL (X)	11-36
Using the PSTATUS command	11-36
PSTATUS	11-36
Responding to the Attention light or alarm	11-40
ATTENTION	11-40
A. System command summary	A-1
B. Editor command summary	B-1
C. Font editor command summary	C-1
D. Command file command summary	D-1
E. Character code assignment tables	E-1
IBM BCD table (6-bit representation)t	E-1
Honeywell 200/2000 BCD table (6-bit representation)t	E-2
Honeywell 6000 BCD table (6-bit representation)t	E-3
Fieldata translation table	E-4
UNIVAC ASCII character set	E-5
Standard ASCII character set	E-6
Standard EBCDIC character set	E-7

F. Page formatting	F-1
Page orientations	F-1
Landscape orientation	F-2
Portrait orientation	F-3
G. Crash recovery and dump	G-1
H. Offline notes	H-1
Input unpacking examples	H-1
Valid host computer/label specification table	H-2
Host system JDLs on the system software tape	H-3
I. Online notes	I-1
Processing 3211/4245 commands	I-1
Universal Character Set Buffer (UCSB)	I-2
Forms Control Buffer (FCB)	I-3
Character correspondence in UCSB associative field	I-4
Emulated command codes	I-5
Online dump example	I-6

Introduction

The Xerox DocuPrint 180 LPS Operations Reference discusses the functional characteristics of the Xerox DocuPrint 180 LPS operating software in order to support the LPS operator, the application programmer. Its purpose is to:

- Provide an overview of laser printing system operations.
- Describe individual commands for system operation, management, and job control.
- Describe editor and operator command files.
- Explain the special considerations applicable to user interface, communication, and graphics features.

About the reference set

This document is part of a reference set designed to help you receive maximum benefit from your Xerox DocuPrint 180 laser printing system.

To help you select the appropriate document for your needs, the following section identifies the documents in the set and describes the information contained in each.

Xerox DocuPrint 180 Laser Printing System document set

The Laser Printing System documentation set consists of the following:

Xerox DocuPrint 180 LPS Product Reference

This publication includes the following information:

- Product overview
- Hardware and software
- LPS connections
- User considerations
- LPS comparisons

Xerox DocuPrint 180 LPS Operator Guide

This publication includes the following information:

- System overview
- PC UI procedures
- Paper facts and procedures
- Operating procedures
- Maintenance
- Problem solving
- Supplies
- Meter reading and reporting

Xerox DocuPrint 180 LPS PDL Reference

This publication includes the following information:

- Print Description Language components and processes
- Input processing functions
- Output processing functions
- PDL/DJDE command summary
- Page formatting guidelines
- Character code assignment tables
- PDL/DJDE programming information with step-by-step instructions

Xerox DocuPrint 180 LPS System Generation Guide

This publication includes the following information:

- Configuration options
- Commands
- OSS software installation, upgrade, and modification procedures

Xerox DocuPrint 180 LPS Operations Reference

This publication includes the following information:

- Command syntax for operator and system administrator procedures
- LPS defaults
- LPS resources
- Command summaries
- Communication and graphics on the LPS
- Command files

Xerox DocuPrint 180 LPS Forms Creation Guide

This publication includes the following information:

- Basic concepts for creating forms
- Coding and compiling for LPS Forms Description Language
- Sample form setup command sets
- Tips for successful forms creation

Xerox DocuPrint 180 LPS PC UI Reference	This publication includes the following information: <ul style="list-style-type: none">• PC UI procedures• Hierarchy of PC UI windows
Xerox DocuPrint 180 LPS Operator Command Summary Card	This publication provides a quick reference of commonly-used operator commands.
Xerox DocuPrint 180 LPS Message Guide	This publication includes the following information: <ul style="list-style-type: none">• OSS and other messages• Meaning and recovery procedures
Xerox DocuPrint 180 LPS Installation Planning Guide	This publication includes the following information: <ul style="list-style-type: none">• LPS basic components and options• Tasks that must be accomplished before installation• Preinstallation requirements• Installation process• Post installation activities
Xerox LPS Tape Formats Manual	This publication includes the following information: <ul style="list-style-type: none">• Characteristics of different formats• File organization• Data formats• Carriage control conventions
Xerox LPS Standard Font Library Font User Guide	This publication includes the following information: <ul style="list-style-type: none">• Font naming conventions• Listing of standard fonts• Data sheets• Glossary of typography terminology
Helpful Facts About Paper	This publication includes the following information: <ul style="list-style-type: none">• Selection and guidelines• Storage and handling• Specifications for different printers

Who should use this manual

This manual serves the information needs of three types of readers who use the Xerox DocuPrint 180 LPS. The two sections below describe the three reader groups and the Xerox DocuPrint 180 LPS.

LPS system administrators, programmers, operators

This manual is written to help you use your laser printing system (LPS) effectively. You can find helpful information in this manual if you are:

- A system administrator managing a Xerox DocuPrint 180 LPS
- A programmer writing print applications for a Xerox DocuPrint 180 LPS
- An operator running print jobs on a Xerox DocuPrint 180 LPS.

The reader who must use the information in any section is always the most important reader of that section. Therefore, the manual addresses “you” directly, regardless of whether the material is pertinent for the system administrator, the programmer, or the operator.

Xerox DocuPrint 180 LPS

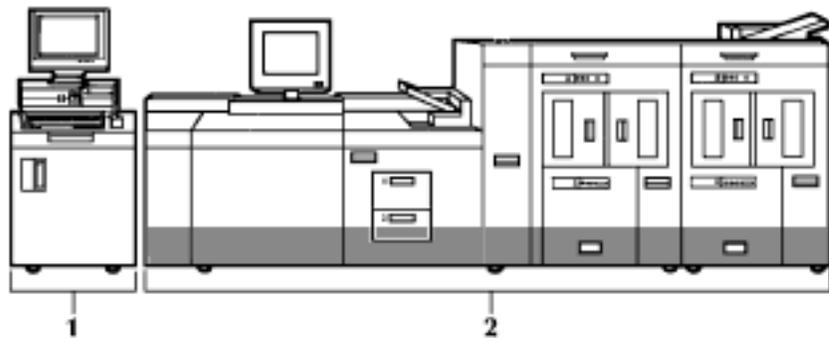
The Xerox DocuPrint 180 laser printing system is a high-speed, high-performance printing system. It processes and prints data from a variety of sources. It enables host mainframe computers and network-connected devices (such as workstations and graphic scanners) to produce publications that incorporate text in a variety of fonts with accompanying graphics, logos, forms, and signatures. Management information systems (MIS) and data processing (DP) environments use the Xerox DocuPrint 180 LPS as a high-performance printer that also provides built-in storage and management of forms, fonts, and other document resources.

In general, the Xerox DocuPrint 180 LPS operating system software is compatible with 4850/4890 LPS V4.0, 9790/8790 LPS V3.9, 4050 LPS V3.8, 4135 LPS V1.5, 4635 V3A, and DP96 software.

The Xerox DocuPrint 180 LPS consists of two separate units illustrated in figure I-1:

- System controller
- Printer.

Figure 1. **Xerox DocuPrint 180 LPS**



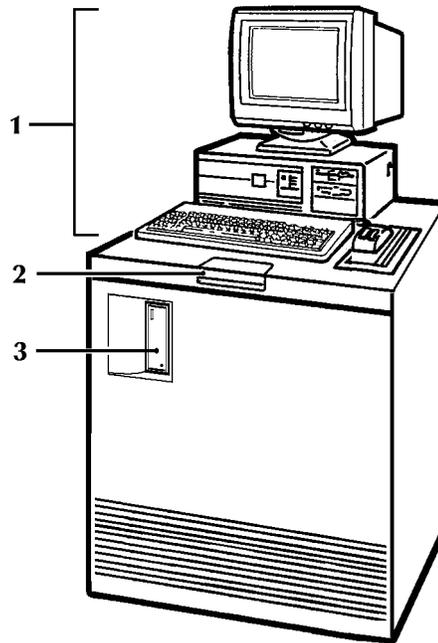
- 1 **System controller**
- 2 **Printer**

Xerox DocuPrint 180 system controller components

The system controller is the LPS subsystem that allows you to interact with the printer. Use either the personal computer user interface (PC UI) or the system controller terminal to enter commands and select options that control the operation of the printer.

Figure 2 depicts the components of the system controller.

Figure 2. **System controller components**



- 1 **System user interface (PC UI)**
- 2 **Operator control panel**
- 3 **.25-inch cartridge tape drive**

System user interface

The system user interface consists of either a PC (containing a processor, screen, a keyboard, a mouse, a mouse pad, and an optional mouse tray) or a character terminal (containing a keyboard and screen only).

The menu, window, and mouse interface of the PC UI are documented by the *Xerox DocuPrint 180 LPS PC UI Reference* and by the *Xerox DocuPrint 180 LPS Operator Guide*.

The operator commands entered at the message window of the PC UI are the same commands entered at the terminal. These commands are documented by this manual.

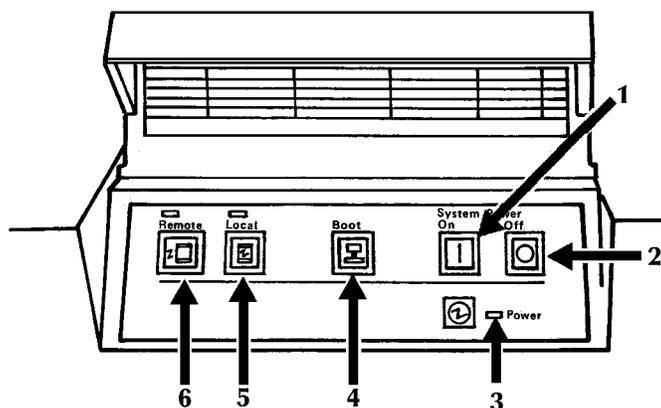


Note: The operator commands documented by this manual run the same LPS tasks that the PC UI operator can also execute with the menu, window, and mouse interface.

Operator control panel The operator control panel allows you to power the printer on and off, boot the operating system software, and switch the LPS between remote and local operation.

Figure 3 depicts the operator control panel

Figure 3. **Operator control panel**



- 1 Power On switch
- 2 Power Off switch
- 3 Power On indicator
- 4 Boot switch
- 5 Remote switch and indicator
- 6 Local switch and indicator.

Xerox DocuPrint 180 tape and cartridge devices

The Xerox DocuPrint 180 system uses three types of devices. These devices are referenced by the labels TD0, TD1, and TD2 since the physical device names are assigned with the appropriate TDn at sysgen or at mini-sysgen.

For systems with an open-reel tape:

- TD0 is the default for the open-reel tape drive
- TD1 is the default for the 1/4 inch cartridge tape drive
- TD2 is the default for the 18/36 track cartridge tape drive

For systems without an open-reel tape:

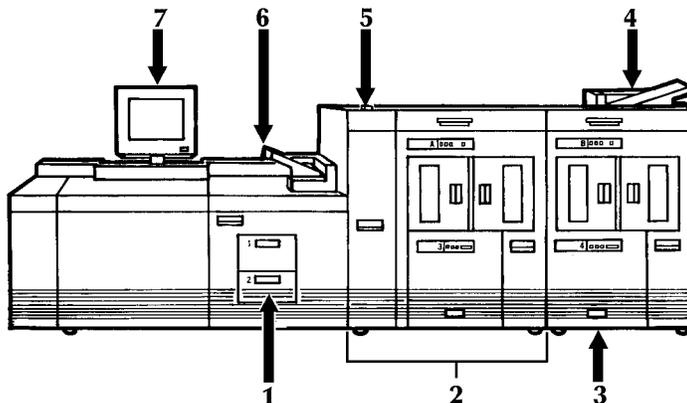
- TD0 is the default for the 1/4 inch cartridge tape drive
- TD1 is the default for the 18/36 track cartridge tape drive

Refer to the *Xerox DocuPrint 180 LPS System Generation Guide* for more information.

Xerox DocuPrint 180 printer components

The printer is the LPS subsystem that processes the electronic images and produces the printed report. Figure 4 depicts the components of the Xerox DocuPrint 180 printer.

Figure 4. Xerox DocuPrint 180 printer components



- 1 Processor feeder trays
- 2 Inverter feeder/stacker module (containing inverter, high-capacity feeders, and high-capacity stacker bins)
- 3 Feeder/stacker module (containing high-capacity feeders and high-capacity stacker bins)
- 4 Purge tray
- 5 Attention light
- 6 Sample tray
- 7 Printer control console

For more detailed information refer to the *Xerox DocuPrint 180 LPS Operator Guide*.

How to use this manual

The following section describes how to use this manual most effectively.

If the LPS is new to you

If your experience with the LPS is limited, read all the information in this introduction and the overview of LPS processing presented in the “LPS print processing” chapter.

Know the key terms and conventions

Become familiar with the key terms, command syntax conventions, and masking conventions presented in the “Key terms and conventions” section of this introduction. Refer to this section whenever necessary.

The first time this manual uses a term, the term appears in italics and is followed by a definition.

Use the “Contents of this manual” section below to pinpoint quickly the chapter that provides the information you seek.

Use the table of contents, index, and glossary

Use either the table of contents or the index to identify which pages or sections of this manual provide the information you seek. Note that each chapter also begins with its own table of contents.

Use the glossary to look up the acronyms or terms that are unfamiliar to you.

Refer to the related publications list

Identify sources of additional information on your LPS system and related material by referring to the “Related publications” listing that precedes the table of contents in this manual. The listed publications address topics such as:

- Print description language
- Forms description language
- LPS network and host options
- Fonts
- Training
- System generation
- Tape formats.

Key terms and conventions

Terms and conventions used in this manual are described below.

Key terms

The following LPS terms are considered equivalent in this manual:

- Enter key (<ENTER>)
- Print tray (output tray, stacker tray, bin).



Note: Print tray refers to a stacker tray; sample tray refers to the tray used only for sampled output.

Syntax conventions

The following are some syntax conventions used in this manual.

KEYWORD variable

Command keywords are shown in uppercase letters; variable information that you determine is shown in lowercase italics.

Only the characters “A” to “Z” and “0” to “9” are valid characters for use in file names. You can use the period (“.”) only as a delimiter between the file name and the file type. If you work on an Ethernet workstation, ensure that your document file names follow these rules. Otherwise print errors occur when you send files to the LPS.

Avoid using any PDL, FDL, DJDE reserved word or file name, or any other utility keyword as a file, form, font, or logo name. Doing so causes unpredictable results.

COMMAND or COM

Enter command keywords in full or abbreviate them to the first three letters.

Table 2 lists the syntax conventions and their use.

Table 2. **Syntax conventions**

Syntax convention	Explanation
<i>inkref or dots</i>	Variable names or values are represented in italics.
a b c	Choices are separated by vertical bars.
{a b c}	Required choices are enclosed in braces.
[a b c]	Optional choices are enclosed in brackets.
<u>b</u>	Default options are underlined.
...	(Ellipsis) indicates repetition of an element, for example: <i>(form-id₁[,form-id₂][,...])</i>

Masking conventions

A mask is a generic file name used to specify a multiple number of files. Within the mask, you can substitute one or more question marks (?) or asterisks (*) for valid alphanumeric characters. Existing file names are compared to the mask. An asterisk or question mark equals all characters. If the last character of the mask is an asterisk, any subsequent positions equal the asterisk. If the last character of the mask is a question mark, subsequent positions must be blank (for example, files with names longer than the mask are not equal).

? example. File mask AB??.FNT represents all .FNT files with names:

- AB.FNT
- AB(3rd character).FNT
- AB(3rd&4th characters).FNT

* example. Where x represents any letter or any digit between 0 and 9, file mask AB*.FNT represents all .FNT files with names:

- AB.FNT
- ABx.FNT
- ABxx.FNT
- ABxxx.FNT
- ABxxxx.FNT

Contents of this manual

The *Xerox DocuPrint 180 LPS Operations Reference* is divided into the following tabbed sections and chapters.

Related publications	Lists associated Xerox publications.
Table of contents	Lists the contents of this manual by chapter headings and section and subsection headings. Ends with separate lists of the tables and figures in this manual.
Introduction	This introduction identifies the readers of this manual and the LPS systems they use. It describes the contents of this manual, and explains how to use it.
Chapter 1. Setting LPS parameters	This chapter explains procedures for LPS booting, and presents the commands that establish LPS operating parameters.
Chapter 2. Managing LPS resources	This chapter explains how to maintain optimal disk usage, back up system files, and modify font and logo files.
Chapter 3. Tracking system activity	This chapter explains how the LPS records system activity data and how you can access, modify, print, and back up this data.
Chapter 4. Editing source files	This chapter describes the file editor available on the operator console for creating and modifying files.
Chapter 5. Using command files	This chapter describes command files, editor command files, and special command file commands.
Chapter 6. Editing font files	This chapter explains the utility used by font knowledgeable users to create font files by using specified font files which exist at an installation as source files.
Chapter 7. LPS print processing	This chapter briefly introduces laser printing systems. It identifies their components, explains the processing flow of a print job, defines key job processing terms and describes the programmer and operator roles.
Chapter 8. HIP processing on the LPS	This chapter provides an overview of HIP communication modes.
Chapter 9. Graphics on the LPS	This chapter describes the system features, processing modes, and the various media formats supported for graphics.
Chapter 10. Setting up print jobs	This chapter explains how to control paper stock to print jobs, and how to set up the output options for print jobs. Explains the programmer and operator roles involved.

Chapter 11. Running print jobs This chapter explains how to print samples, how to control tape input, host online input how to run a print job and how to check print job status.

Statement command summaries and appendices

The following are the appendices contained at the end of this manual:

- System command summary
- Editor command summary
- Font editor command summary
- Command file command summary
- Character code assignment tables
- Page formatting
- Crash recovery and dump
- Offline notes
- Online notes.

A glossary and an index are provided at the back of the manual.

1. Setting LPS parameters

This chapter describes the parameters that may be set for the LPS.

Communicating with the LPS

The commands you enter at the DocuPrint 180 Laser Printing System (LPS) Personal Computer User Interface (PC UI) control functions such as:

- Loading the operating system
- Switching between online and offline processing modes
- Initiating printing
- Building and modifying source files
- Requesting sample prints of data, forms, fonts, and logos
- Obtaining accounting summaries.

While print jobs are running, messages are displayed concerning the status of the jobs and, if necessary, the unexpected occurrences that require operator intervention.

The Xerox DocuPrint 180 LPS has two user interfaces:

- **PC User Interface.** Use the PC UI to handle:
 - Job programming and control features
 - System utilities and diagnostics
 - Paper management and feeder/stacker control
 - Console logs
 - Audit logs and reports
 - Window editor

The menu, window, and mouse interface of the PC UI are documented by the *Xerox DocuPrint 180 LPS PC UI Reference* and by the *Xerox DocuPrint 180 LPS Operator Guide*.



Note: The operator commands documented by this manual run the same LPS tasks that the PC UI operator can execute with the menu, windows, and mouse interface.

- **Printer control console.** Use the printer control console to handle:
 - Paper management and feeder/stacker control
 - Jam clearance
 - Printer diagnostic functions

When you communicate with the printing system:

- Abbreviate all commands to the first three characters or spell them out fully.
- Do not use a command entered in one language (for example, English) interchangeably with the equivalent command in another language (for example, French).
- Pressing a function key invokes the specified action. Do not also press the <ENTER> key.
- Keyed-in commands are executed when you press the <ENTER> key.
- Command delimiters are blank (space), end of line, comma, and slash.
- Use only uppercase letters A to Z, and digits 0 to 9 in source file names.
- System error and information messages are preceded by OS and a four-digit code. The complete set of system messages is documented in the *Xerox DocuPrint 180 LPS Message Guide*.
- Messages without the OS four-digit code prefix are also documented in the *Xerox DocuPrint 180 LPS Message Guide*.
- A message acknowledging your request is displayed to provide you with positive feedback. In system and operator dialog examples, these messages are underlined to differentiate them from system messages.
- The error message `OS2710 Invalid command re-enter` displays when you enter an erroneous command or keyword within a command.

Installation and system generation

The initial considerations and procedures for installing a Xerox laser printing system (LPS) and the operating system software (OSS), that can be tailored to a site-specific configuration, are documented in the *Xerox DocuPrint 180 LPS Installation Planning Guide* and *Xerox DocuPrint 180 LPS System Generation Guide*. (Refer to the “Related publications” section preceding the table of contents in this manual.)

Your LPS service representative oversees the installation of the DocuPrint 180 LPS.

User interface PC UI

Using the Xerox DocuPrint 180 LPS has a PC UI, the major steps are:

- Back up customer files.
- Install MS-DOS 6.2?.
- Install Microsoft Windows 3.1.
- Install Xerox DocuPrint 180 PC UI software.
- Install Xerox DocuPrint 180 UI dialog software.
- Install Xerox DocuPrint 180 printer software.

Powering on and booting the LPS

This section discusses the power-on and boot process you use after the first sysgen has been run.

PC UI

The Xerox DocuPrint 180 LPS has Power On switches for the User Interface (UI) computer and monitor and for the printer control console. It also has a Local/Remote switch for the printer. Before setting the main Power switch to On, check that the printer Local/Remote switch is set to Remote. Then set the main Power switch to On, and set the UI computer and monitor Power switches to On.



Note: If the UI does not operate, reset the UI by doing one of the following:

- Pressing the CTRL/ALT DEL keys simultaneously
- Pressing the Boot/Reset button on the PC
- Turning the UI power switch off, then on.

After the PC UI has completed its software initialization, the Xerox DocuPrint 180 PC UI displays the Boot menu, assuming that the Auto boot function was not enabled during sysgen. Select `Operating System` to boot the LPS from the system disk or `System Disk` to access terminal emulation mode `TEM` (which allows you to perform various system tasks, including `SYSGEN`).

The Xerox DocuPrint 180 boot process involves running system status tests that include checks of main memory, advanced imaging subsystem memory, and the presence of system devices. The full desktop appears after the system boot is completed.

Power loss

If the Xerox DocuPrint 180 LPS loses power or the communications link during online job processing, enter `C` or press the Power Off button. The LPS responds by running its job recovery procedures and displaying the job queue information it displayed prior to the power loss. If the LPS does not respond this way, call your LPS service representative for assistance.

System verification messages

If problems occur while OSS is performing its system verification tests, the Xerox DocuPrint 180 LPS displays system verification messages.

Table 1-1 shows system verification messages.

Table 1-1. **System verification messages**

Message	Explanation
Operating in degraded mode	Problem detected. Other messages will follow.
Devices not accessible name,name,	A configured system device is not in the system configuration. The missing devices are indicated by the following names: DISK (disk) FLOPPY (floppy) TAPE (tape) CARTP (quarter-inch tape) HOST (online channel interface) 3490TP (1/2 inch tape)
Main memory missing: BankS) xx, xy, xz,...	Memory specified during the system configuration phase is missing. The banks are displayed in 16K word sections.
Hardware version mismatch' update configuration	The configured Advanced Imaging Subsystem (AIS)version and the hardware AIS version number do not match.
AIP Firmware error	A firmware error was detected. There is a fault in some part of the AIP firmware.
AIP Download error	There is a missing firmware file, a firmware file cannot be read, a problem with the contents of the firmware file, or a download error occurred causing the checksum to be incorrect.
Problem with AIPOS.sys' file missing or invalid	Problems with accessing or reading the AIS firmware file.
Missing text and metric memory	Problem accessing text and metric memory.
Missing bitmap memory	Bad bank, cannot access bitmap memory.
Missing Ink1 memory, Missing Ink2 memory, Missing PBA1 memory, Missing PBA2 memory, Missing PBB1 memory, or Missing PBB2 memory	Indicate that there is a bad bank, cannot access memory, or memory is not available (not sysgened).
If any of the next four messages is displayed, re-sysgen the system and update the configuration. If this should fail, contact your systems specialist or Xerox Customer Service for resolution.	
Graphics memory not available' Update configuration	Graphics memory is not present according to the hardware, but is sysgened in.
Ink memory not available' Update configuration	Ink memory is not present according to the hardware registers, but is configured into the system.
PBB memory not available' fatal error	Page buffer B memory is not present according to the hardware registers, but is configured into the system.
PBm2 memory not available' fatal error	Highlight page buffers memory is not present according to the hardware registers, but is configured into the system.

Table 1-1. **System verification messages** (continued)

Message	Explanation
SYSTEM NOT OPERATIONAL	<p>Devices were missing, or main memory is below 128K words. The system probably is unable to function even in a degraded mode if this message is displayed. If this message is not displayed, the system may be able to print, but its operation is degraded.</p> <p>You can invoke any system function, although some functions may not execute or may cause a system crash. For example, the lack of AIS memory does not impact a file editing session, but that lack of AIS memory would abort printing.</p> <p>The above messages indicate that the system is not fully functional. Use of the system at such a point is questionable. Contact the LPS service representative for assistance in making your LPS operational.</p>

Printer communication link

The system stops all processing and restarts if a system controller problem occurs.

The Xerox DocuPrint 180 LPS has an interface between the system controller and the printer, part of which is called the client layer. The system initializes the client layer to establish communication between the system controller and the printer after system initialization, after communication is lost, or at the request of technical personnel (see the section "PSC (printer subsystem controller)" later in this chapter).

The system's ancillary Printer message (AIM) processor task establishes communication with the printer, executes client layer initialization, and displays status messages. If a problem initializing the client layer occurs, AIM attempts to establish communication two more times and to initialize the client layer. An attempt lasts between 20 seconds and two minutes. If these attempts fail, AIM prompts you with `Retry ClientLayer initialization? (Y/N)`.

If you enter N, AIM displays `Enter PSC` when you wish to attempt connection to Printer.

Thus you choose between using the system controller for nonprint tasks and loading the communication software again with the PSC command.



Note: If the PSC command fails two times, reinitialize the printer by turning the power off, then on again.

Table 1-2 lists sample messages.

Table 1-2. **Sample messages**

Message	Explanation
Attempting to establish communication with the printer...	The system controller is connecting with the printer.
Communication with the printer is established	System and printer communication is successfully established.
Initializing ClientLayer protocol	Client layer initialization is beginning.
ClientLayer is established	The client layer has been initialized without errors. The system is functional.
Printer ClientLayer is not responding	The printer does not answer client-layer requests.
Printer failed to send configuration	The printer does not send all the printer configuration messages during client layer initialization.
Failed to establish communication with the printer	Three attempts to bring up the link layer have failed.
!Link to printer is down	Communication with the printer was lost during execution of a nonprint task. Displays when the nonprint task completes.
!Loss of communication with the printer	The system controller is not receiving messages from the printer.
Retry ClientLayer Initialization? Y/N	Several attempts to initialize the client-layer have failed. You can request initialization again or terminate the process to run nonprint tasks on your Xerox DocuPrint 180 LPS.
!Sequence errorESS is attempting reconnection to printer.	The system controller has attempted to send a message when the line has failed. AIM is attempting to reestablish communication.
!Link downESS is attempting reconnection to printer.	The link has dropped. AIM is attempting to establish communication.
!Device not readyESS is attempting reconnection to printer.	The printer is not acknowledging a message, the CSI controller is not responding, or the printer is aborting its own messages. AIM is attempting to establish communication.

PSC (printer subsystem controller)

Use this command to download firmware to the specified printed wire board assemblies (PWBA) in the system controller.

Syntax PSC [AIP[,CSI]][,SCSI][,CLI]

Parameters Table 1-3 describes the tasks initialized by utilization of the PSC command.

Table 1-3. **Parameters of the PSC command**

Parameter	Description
AIP	Downloads the AIPOS.SYS firmware and performs diagnostics to check and initialize the various types of memory used by AIS. The messages <code>Initializing AIP</code> and <code>Downloading firmware (version ###)</code> display, followed by memory test messages.
CSI	Sends the software for printer and controller communication and reinitializes the client-layer. The <code>PS1061 Initializing CSI</code> message displays.
SCSI	Displays the current SCSI firmware version.
CLI	Reinitializes the client-layer.

The SCSI firmware is downloaded to the SCSI printed wire board (PWB) automatically.

If the download succeeds, no more messages display. If an error occurs, refer to the *Xerox DocuPrint 180 LPS Message Guide*.

Examples The CSI firmware can be downloaded at any time by entering:

PSC CSI

On the Xerox DocuPrint 180 LPS the printer must be idle to run the PSC task. If the HIP task is in process when you request PSC, the system prompts you to confirm the PSC request. If HIP is running when you confirm the PSC request, then the HIP communication line is disconnected. In addition, printer messages sent but not yet processed may be lost.

Restriction You can restrict access to PSC with RCU.

Configure a finishing or feeding device

The following describes how to configure a finishing or feeding device.

Create the FCG.LIB file

Before you can configure a finishing or feeding device, you must modify a text file called FCG.LIB containing the array of bytes (timing values, constants, flags, and so on) that are loaded into the LPS non-

volatile memory (NVM) by the configuration process. The array is known as the “personality profile” of the finishing device because it indicates to the LPS how to communicate with the finishing or feeding device.

The FCG.LIB file must have one entry for each finishing or feeding device. Each entry consists of a label, an equal sign (=), and a set of values. You create the label, which should help you to identify the device and indicate its configuration if possible. For example, a saddle-stitch profile might be given the label SADLSTCH.

- Entries must begin with an alphanumeric character (0-9, A-Z).
- Each entry provides the NVM values for one finishing or feeding device.
- Each entry must have a label that begins the record.
- Although the label may consist of more than eight characters, the LPS processes only the first eight characters.
- The equal sign separates the label from the array values.
- Commas separate each array value from preceding and following values.
- Only whole numbers can be entered for parameter values (no decimals).
- There are no defaults for any parameter. All values must be specific for the entry to be accepted.
- Each entry is terminated by a semicolon (;).

The FCG.LIB file can also include comment lines. Comment lines must begin with a non-alphanumeric character.

Finishing device entry syntax

label=OUT180, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11;

label name attached to the profile entry.

OUT180 distinguishes the entry as a 4135, 4635 or Xerox DocuPrint 180 output device profile.

Parameters Table 1-4 shows the parameters of the Finishing device entry.

Table 1-4. **Parameters of the Finishing device entry**

Parameter	Description
p1=	<i>Time between Sheets</i> Specifies the minimum time the device requires between the trail edge of one sheet to the lead edge of the next sheet. Range is 0-32767 ms.
p2=	<i>Sheet Jam Time</i> Specifies the maximum time the LPS waits for a sheet delivery signal after a sheet exits before declaring a jam. Range is 0-13000 ms.
p3=	<i>Time between Sets</i> Specifies the minimum time between the trail edge of the last sheet of one set to the lead edge of the first sheet of the next set. Range is 0-32767 ms.
p4=	<i>Set Jam Time</i> Specifies the maximum time the LPS waits for a set delivery signal after a set exits before declaring a jam. Range is 0-13000 ms.
p5=	<i>Time to Cycle Up</i> Specifies the amount of time the LPS should wait after the cycle up signal is sent before the bulk input feeder is capable of sending a sheet. Range is 0-32767 ms.
p6=	<i>Attributes</i> Allows additional control of the SF1 and SF2 function signals. Valid values are 0: ESS control, 4: Always on, and 8: Always off.
p7=	<i>Statuses Supported</i> Expects a status from the finishing device as listed below: S0: Offline or Not Ready, value is 1. S1: Faulted, value is 2. S2: Full, value is 4. S3: Sheet delivered, value is 8. S4: Set delivered, value is 16. Range is 0-31. If all statuses are supported, the value is 31. If no statuses are supported, the value is 0.
p8=	<i>Delivery Signal Type</i> Identifies the required delivery signal as triggered by the lead edge or the trail edge. Range is 0-1: Lead Edge triggered = 0 Trail Edge triggered=1.
p9=	<i>Delivery Start Adjust</i> Adjusts the timing of the start of the delivery signal. Range is -100 to +300 ms for the lead edge, -100 to +75 ms for the trail edge.

Table 1-4. **Parameters of the Finishing device entry**

Parameter	Description
<i>p10=</i>	<i>Delivery Pulse Width</i> Adjusts the timing of the termination of the delivery signal. Only applies to the trail edge of the sheet; not applicable to the lead edge. Range is 30—110 ms.
<i>p11=</i>	<i>End of Set Offset</i> Informs the LPS of the amount of time required between the sheet delivery and set delivery signals. Range is 0-255.

Example GNTST = OUT180, 1000, 1500, 2000, 3500, 0, 2, 31, 0, 0, 0, 0;

Feeding device entry syntax *label=IN180, p1, p2, p3, p4, p5;*

label name attached to the profile entry.

IN180 distinguishes the entry as a 4135, 4635 or Xerox DocuPrint 180 input device profile.

Parameters Table 1-5 shows the parameters of the Feeding device entry command.

Table 1-5. **Parameters of the Feeding device entry command**

Parameter	Description
<i>p1=</i>	<i>Time to Feed sheet</i> Specifies the time required by the bulk feeder between the feed sheet command being received and the required sheet's arrival at the first LPS sensor. Range is 0-500 ms.
<i>p2=</i>	<i>Time to Cycle Up</i> Specifies the amount of time the printer should wait after the cycle up signal is sent before the feeder is capable of delivering the first sheet of paper. Range is 0-32767 ms.
<i>p3=</i>	<i>Attributes</i> Specifies which specific stocks are supported by the bulk feeder, as listed below: D1: Drilled paper D2: Ordered stock Range is 0-7.
<i>p4=</i>	<i>Statuses Supported</i> Receives a status from the feeding device, as listed below: S5: Bulk Input Medium Low S6: Not Ready S7: Sheet fed Range is 0-7.
<i>p5=</i>	<i>Feed Signal Adjust</i> Specifies the duration of the feed sheet command the LPS gives to the bulk feeder. Range is 10-100 ms.

Example GENFD=IN180, 200, 800, 7, 3, 45;

Procedure The procedure follows:

- Step 1.** Open this file within the text editor by entering: EDIT FCG.LIB.
- Step 2.** Key in the entry record for the finishing device to be configured or modify the array values if necessary.
- Step 3.** Save the file.
- Step 4.** Exit the Editor.
- Step 5.** Submit the values for the printer specified by the entry by entering FCG *profile-label*.

System response:

1. The LPS verifies that no print job is being processed. If the LPS is processing a print job, the LPS displays the message OS1970 FCG exiting: Unable to run during a print job.
2. If no print job is being processed, the LPS accesses the FCG.LIB file. If this file does not exist, the LPS displays the message OS1971 FCG exiting: FCG.LIB not present on system.
3. If FCG.LIB is found, the LPS checks for the specified entry label in the file. If the entry label is not in the file, the LPS displays the message OS1972 FCG exiting: Unable to locate entry label. If the specified entry label appears in the file more than once, the LPS displays the message OS1973 FCG exiting: Duplicate profile entries.
4. If the specified entry label is found, the array values for the entries are checked to verify that they are integers within range. If they are not, the LPS displays the message:
OS1978 FCG exiting: Data value is out of range.
5. Finally, the number of values in the entry is checked to verify that no more than 11 values are specified. If an incorrect number of parameters are specified, the LPS displays the message OS1976 FCG exiting: Invalid profile encountered.
6. If an error occurs during step 4 or 5, the LPS displays the message Profile not loaded.
7. If no error occurs during step 4 or 5, the LPS copies the array values to the printer's, and the LPS displays the message OS1974 FCG exiting: Profile for <profile-name> successfully loaded.

FCG (display the profile labels)

Use this command to display the profile labels in the FCG.LIB file.

Logon Level Levels 2 through 5.

Syntax FCG

FCG (configure a finishing or feeding device)

Use this command to load into printer memory the personality profile the LPS must use to communicate with the finishing device or bulk feeder. The personality profile is an entry in a text file called "FCG.LIB." Once this file is created, it is preserved across sysgens.

Logon Level Levels 2 through 5.

Syntax FCG {*profile-label*}

Parameters Table 1-6 shows the parameters used in FCG to configure a finishing or feeding device.

Table 1-6. **Parameters of the FCG (configure a finishing or feeding device) command**

Parameter	Description
<i>profile-label</i>	Specifies the entry in the FCG.LIB file to be sent to the printer.

Reference For more detailed procedural information, refer to the *Xerox DocuPrint 180 LPS Operator Guide*.

FCG SHOW (check finishing or feeding device settings)

Use this command to check the settings for the finishing or feeding devices configured for the LPS.

Logon level Levels 2 through 5.

Syntax FCG SHOW

Restrictions SHOW cannot be abbreviated to SHO.

Reference For procedural information, refer to the *Xerox DocuPrint 180 LPS Operator Guide*.

FCG procedures

Use this task to:

- Send the configuration values for a feeding or finishing device to the printer
- Display the current device configuration values.

Configure task through command line

Table 1-7 shows the command to send feeding or finishing device configurations to the printer.

Table 1-7. **Parameters of the FCG SHOW (check finishing or feeding device settings) command**

Parameter	Description
FCG profile-label	
<i>profile-label</i>	The name assigned to the feeding or finishing device configuration.

To display all profile labels for the feeding or finishing device, enter the following command:

FCG

To display the current device configuration values, enter the following command:

FCG SHOW

Notes:

- You can enter the **FCG SHOW** only on the PC UI.
- **SHOW** cannot be abbreviated to **SHO**.

Verifying LPS status

The system automatically displays messages describing paper tray status, output bin status, and required operator actions in the system message area. These messages are OSXXXX messages such as OS2080 Paper tray x is empty and OS2210 Bin x full.

Verifying system version and time

The following describes how to verify system version and time.

OCS (system version display)

Use this command to verify OSS version information.

Syntax OCS

Status lines Figure 1-1 is an example of the OCS status display.

Figure 1-1. **OCS status display**

```
OCS
  System version xx; IOT Software version  xx;
  OCS version xxx; Command table version  xxxx;
  System class x; Disk ID:  xxxx
  OS1000 Ready For Commands
```

Time and date displays

The date and time are always displayed on the PC UI in the upper left corner of the screen. The status of the input device, output device, and remote device appear on the bottom of the screen. Refer to the *Xerox DocuPrint 180 LPS Operator Guide* for additional information.

Verifying system status

You can increase the efficiency of service calls by providing the representative with the 6-digit dispatch code generated by the OSS problem utility. Run this utility:

- Before placing a service call
- When the system directs you to key in the PROBLEM command (since it detects a problem it cannot correct).

PROBLEM

Use this command to access the dispatch code.

Syntax PROBLEM

Procedure The display shown in figure 1-3 appears.

Figure 1-3. **Initial PROBLEM display**

```
PROBLEM C00  PROBLEM ANALYSIS RUNNING
PROBLEM ANALYSIS RESULTS' DISPATCH CODES
  ##.##.##,message)
  1. TO REPORT ADDITIONAL PROBLEMS
  2. NO MORE PROBLEMS TO REPORT.
INDICATE SELECTION BY ENTERING NUMBER.
```

A sample dispatch code is 10.00.18 SYSTEM REPORTED
PRINTER PROBLEM.

If you want to report additional problems, enter the number 1. Then the system prompts you to identify the type of problem (figure 1-4).

Figure 1-4. **Prompt for PROBLEM type**

```
WHICH OF THE FOLLOWING TYPES OF PROBLEMS DO YOU WISH
TO REPORT?
  1. COPY QUALITY PROBLEMS
  2. FREQUENT JAMS
  3. OUTPUT TRAY PROBLEMS
  4. PAPER TRAY PROBLEMS
  5. TAPE PROBLEMS
  6. OTHER SYSTEM PROBLEMS
  7. NO OTHER SYSTEM PROBLEMS
INDICATE SELECTION BY ENTERING NUMBER.
```

When you enter 7, the system provides a code that you should record and report to your LPS service technician (figure 1-5).

Figure 1-5. **Field service code display**

```
CALL THE FIELD SERVICE DISPATCHER AND REPORT THESE
NUMBERS
    ##.##.##
OPERATORINDICATE THE ACTION YOU HAVE TAKEN.
  1. SERVICE CALL HAS BEEN PLACED.
    WAITING FOR SERVICE
  2. SERVICE CALL HAS BEEN PLACED.
    RETURN TO THE OPERATING SYSTEM
  3. SERVICE CALL HAS BEEN COPIED FOR LATER CALL.
    RETURN TO THE OPERATING SYSTEM
  4. NONE OF THE ABOVE.
    RETURN TO THE OPERATING SYSTEM
INDICATE SELECTION BY ENTERING NUMBER
```

Before entering 1, 2, 3, or 4, ensure that you record the 6-digit dispatch code to report to your service representative.

If you enter 2, 3, or 4, you return to the operating system and may resume printing. If you enter 1, the system cannot be used until it has been repaired.

Setting LPS parameters

The following section describes how to set parameters for various functions of the LPS.

Setting system parameters

The following describes how to set parameters for specific functions of the LPS.

SETTIME (set time or date)

Use this command to set the system time or date or both.

Syntax SETTIME [*mm/dd/yy*] [*hh:mm[:ss]*]

Parameters Table 1-8 shows the parameters used in setting time and date for the LPS.

Table 1-8. **Parameters of the SETTIME (set time or date) command**

Parameter	Description
<i>mm</i>	The month, represented by either a number between 1 and 12, for example, 3 for March, or a three-character abbreviation determined by the language specified in the last full sysgen, for example, MAR for March. If English is the language, the abbreviations are: JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC.
/	Represents a date-field delimiter, which can be a slash (/), hyphen (-), or space.
<i>dd</i>	The day of the month.
<i>yy</i>	The last two digits of the year.
<i>hh</i>	The hour, using a 24-hour clock (military time).
:	The time-field delimiter, which is a colon (:) or space.
<i>mm</i>	The minute of the hour.
<i>ss</i>	The seconds field. Optional.

Restrictions You can enter this command only when the system is idle (such as offline with no background tasks running, and not printing) and only if you are logged in at class 2 or higher. (Refer to the "Setting file access and task execution privileges" section of this chapter.)

ALIGN (set page print alignment)

Use this command to align the laser image with the paper. The alignment values for each printer must be set at least once, but they remain relatively constant. The alignment values you enter are retained by the system until you change them.

Syntax ALIGN[/300 | /600] [[scans],[dots,][TEST]]

Parameters Table 1-9 shows the parameters used in setting the page and print alignment.

Table 1-9. **Parameters of the ALIGN (set page print alignment) command**

Parameter	Description
/300	Displays or sets the current alignment in 300 dpi.
/600	Displays or sets the current alignment in 600 dpi.
<i>scans</i>	The number of scan lines (across a portrait page or down a landscape page) in dots. The range is -75 to 296 inclusive. The default is 1.
<i>dots</i>	The number of dots along each scan line (up a portrait page or across a landscape page) in dots. The range is 0 to 1499 inclusive. The default is 0. The displayed value is rounded to the nearest multiple of 4.
TEST	Prints a TEST form.

If no parameters are specified, the current alignment values are displayed.

For printing, the LPS rounds the dot position value to the nearest multiple of 8. If this correction falls outside the permissible range, the LPS must be adjusted mechanically to bring it into the range of the ALIGN command limitations.

The system extrapolates alignment values for all other paper sizes after you set the alignment for one paper size.

You can also specify alignment values for the diagnostic mode when LOCKOUT is in effect. This feature is designed for the service technician. Refer to the DALIGN section in chapter 11.

Restrictions The following are the restrictions to setting parameters for page print alignment.

- Do not use the TEST parameter when the system is online. If you do, the OS7400 Invalid JDE/JDL for ON-LINE processing message displays.
- Do not request an alignment sample when a print job is in progress since the system will make the alignment settings without printing the test form.

Procedure An alignment form is provided with the system software to ensure proper alignment of output. Using the Align command with the TEST parameter, shows alignment values (scan lines and dots), the current date and time, and disk id. The file name of this form is TEST, and it resides in the FRM directory. If the TEST parameter is specified and the LPS is offline, the test form is printed automatically; this is the equivalent of using ALIGN and SAMPLE TEST.FRM.

It is best to compile and use the test alignment form prior to running other print jobs. Always use the alignment form to adjust the LPS when you run jobs using preprinted forms. In addition, always sample the test form after any LPS mechanical adjustments are made.

A quick way to test alignment is to sample the form, and then fold it in half once on the dot orientation and once on the scan line orientation. This provides you with an easy measurement of how many dots or lines to add or subtract. After correcting the alignment with the ALIGN command, resample the form and check for any further variation that may be needed. For all but the most critical applications, it is sufficient to get the alignment to the point where the center line is visible on each side of the fold.

Examples The following illustrates how to run a test, change a parameter and retest.

- **ALIGN/300 80,24,TEST**
Alignment is 80 scan lines and 24 dots
- **ALIGN**
Alignment is 80 scan lines and 24 dots
- **ALIGN ,16**
Alignment is 80 scan lines and 16 dots
- **SAMPLE TEST.FRM**

If resetting the alignment with ALIGN does not provide the desired result, modify the PDE BEGIN statement in the job's JSL file and recompile it. Refer to the *Xerox DocuPrint 180 LPS PDL Reference*.

DISPLAY (set display format for messages)

Use this command to specify the format (ASCII, hexadecimal, or both) in which printer-related messages are displayed. If you request ASCII, the hexadecimal codes for fault, hint, and information messages are mapped into user friendly ASCII messages. If you request hexadecimal, these messages are displayed only in hexadecimal. If you request both, they are displayed in ASCII and in hexadecimal.

Syntax DISPLAY {ASCII | HEX | BOTH}

ASCII is the set condition from sysgen, but it will remain as changed.

ENERGY

The default timeout value for the energy saver mode is 90 minutes. The power saving mode of the printer powers off most printer base machine functions, but leaves the printer control system and the printer monitor functioning. The Attention light and alarm are idle in the energy saving mode.

Use this command to establish another timeout value for the printer. The printer uses this value to calculate the time it may remain idle before switching into the power saving mode. The timeout value is saved across boots and power-offs. The system exits the energy saver mode when the printer is given a message by the system controller.

Syntax ENERGY [*timeout*]

Parameters Table 1-10 shows the parameters of the ENERGY command.

Table 1-10. **Parameters of the ENERGY command**

Parameter	Description
<i>timeout</i>	The number 0 or a number within the range 10 through 65,535 minutes. 0 indicates that a timeout should never occur. If you omit the timeout value, the system displays the current timeout value.

Setting parameters for print jobs

The following section describes setting parameters for printing tasks on the LPS.

Reallocating space to the print file

The following describes how to reallocate space for print files.

REA (reallocate print file)

Use this command to set the size of the print file on the system disk. The recommended print file size is at least 400 cylinders. The recommended print file size is created during system generation, but you may need to increase its size based on your installation's requirements. The print file size can be increased or decreased in size any time the system is idle.

The command informs you of the number of cylinders allocated and the number of cylinders actually usable by input and output. The system maintains and updates a permanent record of all cylinders deemed unusable for printing if bad sectors are detected while running a print job. These unusable cylinders are removed from use until the disk is reformatted during a full system generation.

Syntax `REALLOCATE [ddd | ? | !]`

The parameter set, `[ddd | ? | !]`, describes the print file which is discussed in this section.

This command invokes the reallocation processor, which determines current space availability for the print file. If any parameter is entered, no questions are asked during the reallocation process. The warning messages that inform you not to disturb the process are still displayed. The size of the current print file is not displayed.

REALLOCATE proceeds by calculating the current size of the file, then deletes it. The desired size is calculated by using the previous size if the new one is not specified. The REALLOCATE then recreates the print file.

Print file parameters Table 1-11 shows the parameters for setting the size of the print file.

Table 1-11. **Parameters of the REA (reallocate print file) command**

Parameter	Description
<i>ddd</i>	Represents a decimal number, which is the number of cylinders requested for the new print file. If <i>ddd</i> is larger than the maximum contiguous space available, it makes the largest size print file it can. If <i>ddd</i> is less than the minimum size for your system configuration, the reallocation process exits and displays the REQUESTED PRINT FILE IS TOO SMALL message.
?	Creates up to the default size for the print file if one does not exist. If a print file already exists, a print file of the same size is created. The default size is 400 cylinders. This command is specifically for use in the SYSGEN.CMD file.
!	Creates a default size print file even if one already exists.

If the print file size is less than the minimum configuration or is too small to handle the job, the printer cycles up and down. This is not desirable.

Procedure If you enter REALLOCATE without parameters, the system executes the complete interactive process.

After you invoke REALLOCATE, the following message displays:

```
DO you want to reallocate a different size (yes/no)?
```

If you do not wish to change the current print file size, enter NO. The **** Resuming normal operation **** message displays, and processing exits without changing the print file size.

If you enter YES, the prompt to enter the size displays.

```
REA VERSION D01 NUMBER OF CYLINDERS IN THE CURRENT
PRINT FILE IS aaa (bbbb sectors)
```

aaa and bbbb are the number of cylinders and sectors respectively in the print file currently on the disk.

```
Number of cylinders available for print file is ccc
(dddd SECTORS)
```

The value shown is the largest contiguous free space on the disk and is presented in cylinders and sectors.

```
How many cylinders should the print file contain?
zzz
```

Enter a value of at least 400. This value must be at least 50 cylinders less than the total free space on the disk. A value of more than 400 is beneficial in the event that the system experiences disk saturation situations. Otherwise, the 400 value should be sufficient. If more than the maximum cylinders available is requested, a print file of the maximum number of available cylinders is created.

n cylinder print file allocated with m usable cylinders

- n is the number of cylinders requested and allocated to the print file.
- m is the number of usable cylinders within the print file (allocated cylinders minus the cylinders flagged as unusable within the print file).

**** Resuming normal operation ****

At this point, the reallocation processor has completed its function, and the system is ready for normal operation.



Caution: Interrupting the reallocation process sometimes renders sections of the disk unusable.

If REA completes by displaying `PRINT FILE TOO SMALL`, no print file has been created. In this case use the `COMPRESS DP1:` command to create sufficient free disk space. Then use REA again. If REA fails a second time, back up or delete user files on the disk, or move them to another disk.

REA and disk management

Manage cylinders in order to maintain the required minimum print file size. Doing so assures printing without input or output processing lockup.

Procedure The message `Requested Print File is Too Small` appears if the value you enter is smaller than the minimum number of cylinders.

If the minimum size print file cannot fit on the disk:

Step 1. Enter COM DP0: (or COM DP1:, or COM DP2: , or COM DP3:).

Use this command to compress the files on the disk, creating contiguous free space.

Step 2. Enter the REA command again.

If the minimum size print file can fit on the disk:

Step 1. Enter FCH DP0: (or FCH DP1:, or FCH DP2: , or FCH DP3:).

This displays the disk addresses and number of blocks available and the regions of the disk.

Step 2. Check the value reported for the Largest Number of Blocks to identify the space available for the print file.

Step 3. Multiply the minimum number of cylinders by 400 to identify the minimum number of blocks needed.

Step 4. Enter FIL/EX DP0: (or FIL/EX DP1: or FIL/EX DP2: or FIL/EX DP3:).

This allows you to identify the files that can be deleted from the disk, and to decide the combination of files to delete to provide enough free space.

Step 5. Delete files you have selected.

Step 6. Enter COM DP0: (or COM DP1:, or COM DP2: , or COM DP3:).

Use this command to compress the files on the disk, creating contiguous free space.

Step 7. Enter FCH DP0: (or FCH DP1:, or FCH DP2: , or FCH DP3:).

Use this command to verify that the Largest Number of Blocks value is greater than the minimum print file size.

Step 8. Delete more files and compress the disk again if the Largest Number of Blocks value is too small. (You may delete the HIPBF.SYS file if HIP is not loaded on the LPS and your site has backed up the file for the LPS.)

Step 9. Enter the REA command again.

REA and online report spooling

You can reallocate small print files by restricting input processing to control report spooling. Online users can specify the number of reports the input task can process ahead of output. Refer to the "LEASH" section in chapter 7 for additional information.

Setting job parameters

The following section describes the different parameters that can be set for individual printing jobs.

FONTS

Use this command to optimize the reallocation of task memory by specifying the maximum number of fonts the system attempts to keep resident in memory during a print job. This value can have a significant impact on system performance for some jobs.

Syntax FONTS *number*

Parameters Table 1-12 shows the parameter setting for FONTS.

Table 1-12. **Parameters of the FONTS command**

Parameter	Description
<i>number</i>	<p>A positive, non-zero integer less than 129, which specifies the maximum number of active fonts per print job. No single page (or both sides of a duplex sheet, combined) printed can invoke more fonts than specified. Ensure that this number includes all fonts and logos used on forms and variable data.</p> <p>The fonts for literals, logos, and the variable data in the form, if a form is used, must be included in the specified number.</p> <p>You can specify a larger number of fonts than will ever be used on a single page, thus allowing font changes on a page-to-page basis without throughput degradation (loss of printing speed) if all the fonts can be loaded into memory. If <i>number</i> is omitted, the system displays the value currently in effect. The default is 32 fonts.</p>

The FONTS number remains in effect until the FONTS command is reissued or a new operating system is generated.

A print job may actually use more than the specified number of fonts as long as no more than this number of fonts is called for on a single page or on both sides of a duplex page, combined. If, during a print job, the current FONTS value is exceeded while processing a page (such as the total number of fonts required for form printing plus those required for variable data printing), the system aborts that job and displays the message OS2885 Max number of fonts exceeded. Enter new value. Restart job.

FORMS

Use this command to optimize the allocation of task memory by specifying the maximum number of forms the system attempts to keep in forms memory during print job processing. This value can have a significant impact on system performance for some jobs.

Syntax FORMS *number*

Parameters Table 1-13 shows the parameter setting of the FORMS command.

Table 1-13. **Parameters of the FORMS command**

Parameter	Description
<i>number</i>	A number between 1-64 inclusive. It sets the maximum number of active forms per print job. If you omit <i>number</i> , the system displays the value currently in effect.

A number larger than 1 allows form changes on a page-to-page basis without throughput degradation if all the forms can be loaded into memory.

Restriction The FORMS command remains in effect until you reissue the FORMS command or a new operating system is generated.

GRAPHIC

Use this command to optimize the allocation of task memory by specifying the maximum number of graphics the system attempts to keep in graphic memory during print job processing. This value can have a significant impact on system performance for some jobs.

Syntax GRAPHIC *number*

Parameters Table 1-14 shows the parameter of the GRAPHIC command.

Table 1-14. **Parameters of the GRAPHIC command**

Parameter	Description
<i>number</i>	An integer between 16 and 128. If you omit <i>number</i> , the system displays the value currently in effect.

Restrictions The following are the restrictions to setting the GRAPHIC parameter:

- No single page printed can invoke more than 16 graphics. Ensure this number includes all graphics used on a form for this page if a form is used.
- The GRAPHIC command remains in effect until you reissue it or a new operating system is generated. A print job can use more than the specified number of graphics as long as no more than 16 are called for on a single page.

Adjusting print image darkness

The following describes the adjustment of the darkness of the print image.

DARKNESS

Use this command to make the printed images lighter or darker. When the command is entered without any parameters, the current setting is displayed.

Syntax DARKNESS [*level*[,TEST]]

Parameters Table 1-15 shows the parameter of the DARKNESS command.

Table 1-15. **Parameters of the DARKNESS command**

Parameter	Description
<i>level</i>	A number between 1 and 9 inclusive. Use a higher number to darken the image; use a lower number to lighten the image.
TEST	Prints a sample with the darkness level you specified.

Lightening high-density images

The following explains lightening high-density images.

EDGE

Use this command to decrease the image density on the leading edge of the page, and thus limit how often paper jams. The change is applied the next time the printer cycles up. When the command is entered without any parameters, the current setting is displayed.

Syntax EDGE [NORMAL | STRESS | OFF]

Parameters Table 1-16 shows the parameters that may be set for EDGE .

Table 1-16. **Parameters of the EDGE command**

Parameter	Description
NORMAL	Invokes light screening.
STRESS	Invokes heavy screening.
OFF	Leaves the image density on the edge of the paper unchanged.

no parameter displays the current setting.

OFF the setting after sysgen, but it remains as changed.

Recommendation EDGE prevents fuser jams best when ALIGN is set at 1,0.

Finishing Exerciser Utility

The following section describes the finishing exerciser utility.

FEX

Use this command to generate multiple data set test cases through a third-party finishing device connected to the printer.

Syntax FEX

Restriction FEX cannot run during a print job.

Process The Finisher Exerciser Task will present the operator with a set of questions to allow FEX to construct a temporary test case on the Xerox DocuPrint 180 LPS. FEX will invoke this test case to generate a print job that is designed to assist the operator in the alignment of a third party finishing device that is attached to the system.

At the user interface type the following (in the message window if a PC UI):

FEX <ENTER>

The following response will be displayed.

```
OS0733 Finishing Exerciser Utility (FEX)
starting...
```

FEX will now check for the existence of the required JDL file (FEXDMP.JDL). If it cannot be located on the system disk, the following message will be displayed.

```
OS2920 FEX: Error, FEXDMP.JDL not found.
```

```
Recompile FEX.JSL
```

FEX will now ask for information that is required to develop the test case. The following prompts will be displayed.

```
OS2922 Enter the TOTAL number of sets to create:
```

```
OS2923 Enter the number of pages per set:
```

```
ENTER FEEDER #:
```

Please select a papersize for this job from the following menu:

- 1) US Letter (8.5 x 11 in)
- 2) US Legal (8.5x 14 in)
- 3) US Legal LEGAL KEYWORD
- 4) US B size (17 x 11 in)
- 5) A4 (8.27 x 11.69 in)
- 6) A3 (16.54 x 11.69 in)
- 7) (7 x 10 in)
- 8) Sysgen papersize

What size is loaded for this job?:

```
OS2924 Enter the FRM name (leave off the .FRM):
```

DO YOU WANT SF1 FUNCTION TO ALL SHEETS? (Y/N)

DO YOU WANT SF2 FUNCTION TO ALL SHEETS? (Y/N)

OS2835 Enter 'D' for Duplex printing, 'S' for Simplex printing:

FEX now generates the test case as requested and invokes the LPS to print the test job.

Setting file access and task execution privileges

This section explains the procedures for limiting access to files and task execution.

Protecting access to files

The following describes how access to files may be restricted or limited.

LOGON (restrict file access)

Use this command to restrict access to files of a given type (such as those cataloged in a certain file directory) for a particular action (record editing, deleting, and copying of files). It also allows your installation to authorize or restrict certain system level commands. (Refer to the RCU command.) Five user classes are provided, each with varying degrees of access to files (as illustrated in table 1-23) and operator command usage.

Syntax LOGON [1]
LOGON [2 | 3 | 4 | 5 [,password]]

Parameters Table 1-17 shows the parameters of the LOGON command.

Table 1-17. **Parameters of the LOGON (restrict file access) command**

Parameter	Description
1,2,3,4, or 5	Specifies the user classification. Table 1-23 illustrates how this classification is related to the file access and functions that are performed.
<i>password</i>	A 1 to 15 character string that consists of any characters (including leading, nested or trailing blanks) on the keyboard. Levels 2 through 5 have a default password that you can change.

Changing passwords A user already logged into class level 2, 3, 4, or 5 can change the password for that level by keying in LOGON with no parameters. The system then prompts for the new password.

If the preferred form LOGON n is used, you are requested to key in the password, which is not displayed on the screen as it is keyed, nor is any indication given as to the length of the password.

When the LOGON n password form is entered, the password is displayed on the terminal as entered by the operator.

System level passwords are case sensitive. The default system level passwords are in uppercase. Therefore, if you use lowercase to key in any system level password, the password will not be recognized. Login passwords you create are also case sensitive. Therefore, subsequent entries of any new password must be entered in the same case for the system to consider it valid.

Changing logon levels

A password is always required to access a logon level higher than the current level.

A password, however, may not be required to move from a higher level to a lower level. For example, you are not required to enter a password to access logon level 1, since level 1 is not password protected. And, since logon levels 2 and 5 are designed for customer use, you can also move from logon level 5 to level 2 without a password. Also, until you set a password for level 2, you can move from any higher level to level 2 without a password.

Moving from level 4 or 3 to level 2, or level 5 to 3 or 4, does require a password.

SECURE (restrict file access)

Use this command to protect the following file types against unauthorized access: CMD, CME, FNT, FRM, FSL, IMG, JDL, JSL, LGO, PDE, TST, STK, and LIB.

Syntax SECURE *file-name.file-type*

Parameters Table 1-18 shows the parameters of the SECURE command.

Table 1-18. **Parameters of the SECURE (restrict file access) command**

Parameter	Description
<i>file-name</i>	Specifies a one to six character name of a disk file.
<i>file-type</i>	Specifies one of the directory types listed above.

The following editor commands are affected by the secured file feature: COPY, DELETE, FID, FILE, FIX, FLOPPY, GET, LIST, MERGE, REVIEW, RNAME, and SAVE.

Restrictions A secured file cannot be copied to tape.

Secured files, access, and manipulation

The system supports the security feature by maintaining the security characteristics of files that it manipulates (such as a renamed secured file retains its security characteristics). An unsecured file that replaces a secured file makes the file unsecured. However, if a secured font is referenced in a forms source library (FSL) file compiled at logon level 5, the FSL file and resulting FRM file are not automatically secured.

It is your responsibility to secure the FSL and FRM files. Likewise, if you copy a secured file to a new file or over an existing unsecured file, the resulting file is unsecured.

You must be in logon level 5 to access any secured file. The only exception to this is the FIX command. (Refer to the section "FIX (modify font or logo file)" in Chapter 2.) Fixing of fonts and logos is unaffected by security. You can fix a secured font or logo at any logon level.

Security breach

While printing a job, an attempt at unauthorized file access aborts the operation running at the time of the security breach. The message SECURITY VIOLATION also displays after the message PROBLEM WITH FILE. . .

The contents of a secured file are left in the working storage area when it is manipulated by the editor. You must issue a CLEAR command when terminating an edit session to prevent unauthorized accesses. This type of unauthorized access is not detected or reported.

Limiting task execution privileges

The following describes the limiting of task execution privileges.

RCU (restrict command usage)

Use this command to authorize or restrict the usage of system commands and tasks (PDL, FDL, OSDS, and so on). The system commands and task names are authorized/restricted for users by relating them to the classes previously defined in the LOGON command. To use a specific system command, you must be logged on to a class level equal to or greater than the user-defined (or default) restriction level.

PC UI The PC UI generally functions in “object mode,” in which you use the mouse to select objects, menu items, or entry fields. System tasks, however, are run by the PC UI in two other modes: the terminal emulation (TEM) mode, and the teletype (TTY) mode. Both the TEM and TTY modes are character modes: your input interface with the system controller is the keyboard, your output interface is the display. The TTY mode runs inside an object mode window that you can resize, scroll, and move around. The TEM mode completely covers the object mode screen that returns when you exit the TEM mode.

The mode used depends on the task. The editor, for example, runs in the TTY mode.

Syntax RCU *command*[,*class*]

Parameters Table 1-19 shows the parameters under RCU areas.

Table 1-19. **Parameters of the RCU (restrict command usage) command**

Parameter	Description
<i>command</i>	The system command or task name to be authorized or restricted. There are certain commands that cannot be restricted.
<i>class</i>	The lowest logon class allowed to use the command. If class is not specified, the current class level for that command displays.

Restriction You cannot restrict the use of a command to a level higher than the current logon class. Also you cannot change the class level of a command unless you are authorized to use that command.

Table 1-20 lists the file access classes for the different functions. Each of the columns is described after the table.

Table 1-20. **File access classes**

Function		Record editing			File delete			File copy to disk			File secure	User file
Type v	Class >	1	2,5	3,4	1	2,5	3,4	1	2,5	3,4	5	—
CMD		N	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
CME		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
DAT		N	N	Y	N	N	Y	N	N	Y	N	N
FIS		N	N	N	N	Y	Y	Y	Y	Y	N	N
FNT		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
FRM		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
FSL		N	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
ICT		N	N	N	N	Y	Y	Y	Y	Y	N	N
IDR		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
IMG		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
IPF		N	N	N	N	Y	Y	Y	Y	Y	N	N
IPM		N	N	N	N	N	Y	N	N	Y	N	N
JDL		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
JSL		N	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
LGO		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
LIB*		N	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
LOG		N	N	N	N	N	Y	N	Y	Y	N	N
MSC		Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
OSD		N	N	N	N	N	N,Y	N	N	Y	N	N
PCH		N	Y	Y	N	Y	Y	Y	Y	Y	N	N
PDE		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
SAF		N	N	N	N	N	N	N	N	Y	N	N
STK		N	N	N	N	Y	Y	Y	Y	Y	Y	Y
SYS		N	N	Y	N	N	Y	N	N	Y	N	N
\$Y\$		N	N	N	N	N	N	N	N	N	N	N
TMP		Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
TPF		N	N	N	N	N	Y	N	N	Y	N	N
TSK		N	N	N	N	N	N	N	N	Y	N	N
TST		N	N	N	N	N	Y	Y	Y	Y	Y	Y
XCS		N	N	N	N	Y	Y	Y	Y	Y	N	N

* Some LIB files may be edited; others may not be edited.

Table 1-20 definitions are given below.

Type	Specifies the file directory in which files are cataloged. Each file directory contains a specific type of file (for example file directory JSL contains job source language files). Refer to the "Files you can edit" section of the "Editing source files" chapter for a definition of each directory name.
Class	Specifies the logon class required to perform a particular function.
N or Y	Specifies whether a particular function (record editing, deleting, copying) may be performed.
Record editing	Specifies whether records in a file may be edited. If access mode is N, neither a GET (editor command) nor any record editing commands are accepted.
File delete	Specifies whether a file may be deleted with an editor or OSS DELETE command.
File copy	Specifies whether a file may be copied to system disk with an editor or OSS COPY command.
File secure	Specifies whether a file may be limited to access by only a logon level 5 user.
User file	Specifies whether the file can be backed up with COPY TAPE WRITE LABEL ALL.

2. Managing LPS resources

This chapter explains the management of the various resources of the LPS.

Maintaining optimal disk usage

This section describes the various operations available for maintaining optimal disk usage.

Files on system disks

There is one disk drive provided as base configuration of the controller with up to three additional disk drives available optionally. The system provides various utilities (SFT, DSR, and PSC) and commands (COMPRESS, PURGE, and FCHECK) which you can invoke to maintain system functions and resource usage at optimal levels.

Operating on system files

Disk files are grouped into directories by the system on the basis of their file extensions. Each file extension is an abbreviation which denotes the type of file. Use the OSS and Editor commands described in this section to:

- List the directory by file type
- Delete files from the directory
- Copy files into the directory.

Table 2-1 lists the types of files and their extensions.

Table 2-1. **System files you can operate on**

File extension	File type
CMD	Command file
CME	Copy modification entry file
DAT	System data file
FIS	Font interchange standard
FNT	Font file
FRM	Form object file
FSL	Form source language file
ICT	Ink catalog
IDR	Ink directory
IMG	Digitized image file
JDL	Job descriptor object file
JSL	Job source language file
LIB	Utility file
LGO	Logo file
LOG	System integrity log file
MSC	Temporary user file
OSD	OSDS diagnostic file
PCH	System patch file
PDE	Page descriptor entry file
SAF	SAFES diagnostic file
STK	Stockset file
SYS	System control file
\$Y\$	System file
TMP	Temporary user file
TPF	Temporary patch file
TSK	System task image file
TST	RTEXT object file

Files can be saved to or restored from formatted LPS-labeled tapes. To write files to tape, the system must have a tape unit. Before an uninitialized tape is written, it must be initialized using TAPE VOLINIT. After files have been written, the TAPE REWIND or TAPE UNLOAD is used to rewind the tape. The COPY command, described in the "Editing source files," reads or writes the files.

Only uppercase letters "A" to "Z," and the digits "0" to "9" can be used in source file names.

Table 2-2 lists the file manipulation and maintenance commands that are initiated either from system level or from the editor.

Table 2-2. **File commands available from Editor and OSS**

Command	Purpose
COPY	Copies a file
DELETE	Deletes a file
FID	Creates a prototype command file
FILE	Displays a directory
FLOPPY	Backs up or restores system disk files to LPS floppy
LIST	Lists a directory
REVIEW	Reviews file directories
RNAME	Renames a file
TAPE	Initializes, positions, rewinds, unloads a tape, or lists the files on a tape

Necessary free space on system disks

Several tasks need sufficient disk space to operate properly. The editor creates large .TMP files when editing a file or preparing to print. FDL and PDL need space to create forms, JDLs and listing files. The sysgen task requires space to create new files and to replace files that are larger than the previous version. The consequence of the sysgen task running out of space on disk 0 is potentially serious, possibly requiring a format and resysgen to recover. The disk on which each processor places files varies with the number of disks in the system.

Table 2-3 shows which disks have critical usage by various tasks.

Table 2-3. **Disks used by various tasks**

Task name	File types	One disk in the system	Two disks in the system	Three disks in the system	Four disks in the system
EDITOR	TMP	DP0	DP1	DP2	DP3
FDL	FRM and TMP	DP0	DP0 and DP1	DP2	DP2 and DP3
PDL	JDL and TMP	DP0	DP0 and DP1	DP2	DP2 and DP3
SYSGEN	SYSTEM FILES	DP0	DP0	DP0	DP0

The amount of space needed varies according to the task. A minimum of 2000 contiguous free sectors and a total of 3000 free sectors is recommended for each disk indicated in the table. An FCHECK can be done periodically to determine disk space availability.

File biasing by disk configuration

Use the file locations identified by tables 2-4, 2-5, 2-6, and 2-7 to enhance FDL compilation time.

Table 2-4. **File biasing - 4 disk system**

File class	Disk 3	Disk 2	Disk 1	Disk 0
1	X	X	X	First
2	Last	Third	Second	First
3	Second	First	Last	Third
4	First	Second	Third	Last

Table 2-5. **File biasing - 3 disk system**

File class	Disk 2	Disk 1	Disk 0
1	X	X	First
2	X	Second	First
3	First	Last	Second
4	First	Second	Last

Table 2-6. **File biasing - 2 disk system**

File class	Disk 1	Disk 0
1	X	First
2	Last	First
3	Last	First
4	First	Last

Table 2-7. **File biasing - 1 disk system**

File class	Disk 0
1	First
2	First
3	First
4	First

Table 2-8. **File class and type legend for tables 2-4 through 2-7**

File class	File type
X	not allowed on the disk drive
1	.TSK, .OSD, .SAF
2	.SYS, .LOG, .\$\$
3	.CME, .FNT, .FRM, .JDL, .ICT, .IDR, .IMG, .IPF, .IPM, .LGO, .PDE, .STK, .XCS
4	.CMD, .DAT, .FSL, .JSL, .MSC, .PCH, .TMP, .TPF, .TST, .FIS, .LIB

Restrictions

The following are restrictions to file biasing:

- If a file type is not in Class 1, 2, or 3, then it is in Class 4.
- If the disk you specify has no space left to create the file, then the second disk you specify is used. If no space is found on any disk, the system returns Device Full status.
- You can override the file biasing scheme by specifying the disk drive. For example: COPY TAPE READ LABEL DP1:ABC.FNT will put the file on disk 1 if space is available. If no space is available on DP1:, even if space is available on other disks, the system displays the message Device full.

Checking usage of disk space

The following explains the procedures used in determining the amount of available disk space.

FCHECK (file check)

Use this command to display or print data about file fragmentation on a disk. It lists unusable sectors identified by the sector check process during a full system generation. When bad sectors and fragmented files accumulate, you should purge and then compress the disk.

Syntax FCHECK[/PR][/EXP | /EXG] [disk-id]

Parameters Table 2-9 shows the syntax parameters of the FCHECK command.

Table 2-9. **Parameters of the FCHECK command**

Parameter	Description
/PR	Information is printed. If this is not entered, the information will be displayed at the PC UI
/EXP	Displays or prints the primary defect list (PList). Service representatives use /EXP for disk diagnostic purposes only.
/EXG	Displays or prints the grown defect list (GList). Service representatives use /EXG for disk diagnostic purposes only.
<i>disk-id</i>	The individual disk drive (specified as DP0:, DP1:, DP2:, or DP3:). DP0: is the default.

Display The displayed or printed data appears as shown in figure 2-1.

Figure 2-1. Data reported by FCHECK or FCHECK/PR

```

DISK ID: user disk id
Disk fragmentation check for disk 0 (DPO:)
  Starting      Number of
  LBN           available blocks
  2445          154
  3584          1
  5161          80
  6515          34217
  40733         7505
  48548         172

Number of sectors in groups of 3 sectors or less = 1
Number of sectors in groups of 5 sectors or less = 1
Number of available spaces on disk = 67
Number of available sectors = 42129
Largest available space = 34217
Total number of sectors = 48720

Bad block display for disk 0 (DPO:)

  Disk Serial number: 00102802
  Vendor Identification: SEAGATE
  Product Identification: ST1480
  Product Revision Level: 5826

Remaining Spares Display for disk 0 (DPO:)
Cylinders with all spares available: 2
Cylinders on disk: 1476

Number of cylinders with alternate sparing: 3

Number of cylinders with 0 spares remaining: 11
Number of cylinders with 01 spares remaining: 65
Cylinders with all spares remaining: 1397

Primary Defect List (PList):  format = cylinder/head/sector
0124/0002/0060      0125/0002/0060      0127/0005/0036      0127/0006/0060
      .
      .
      .
0251/0005/0058      0253/0001/0058      0254/0001/0058      0259/0007/0035

Grown Defect List (GList):  format = cylinder/head/sector
0005/0002/0056      0126/0002/0038      0156/0006/0033      0167/0004/0049
      .
      .
      .
1250/0001/0016      1261/0006/0043      1411/0005/0046      1444/0002/0031
1445/0002/0031

All Defect List (PList+GList):  format = cylinder/head/sector
0005/0002/0056      0124/0002/0060      0125/0002/0060      0126/0002/0038
      .
      .
      .
1250/0001/0016      1261/0006/0043      1411/0005/0046      1444/0002/0031
1445/0002/0031

```

The PList is the equivalent of MBAIS information which lists the original bad blocks from the manufacturer. At the time of format, the disk uses the PList to know which blocks are bad.

The GList is the equivalent of BADBLK information. The SCSI system disks reassigns bad block automatically. The GList contains the list of bad blocks automatically reassigned by the disk or the Bad Block Utility (BBU). The blocks reassigned by the disk or by the BBU are not removed from the system bitmap.

When FCHECK or FCHECK/PR is entered, only the All defect list is displayed or printed. When FCHECK/EXP or FCHECK/PR/EXP is entered, the PLIST is shown followed by the All defect list. When FCHECK/EXG or FCHECK/PR/EXG is entered, the GLIST is shown followed by the All defect list. When FCHECK/EXP/EXG or FCHECK/PR/EXP/EXP is entered, the PLIST is shown followed by the GLIST, then the All defect list.

In addition, the manufacturer's serial number is displayed or printed.

The logical block number (LBN) and the number of sectors available is displayed for each group of contiguous, available sectors.

The Number of sectors in groups of 3 sectors or less is an approximation of the number of sectors on the disk less likely to be used than groupings of more than 3 sectors. These sets of sectors are used for small files (for example, small .JSL, .FSL, .PDE files).

The Number of sectors in groups of 5 sectors or less (includes the count of groups of 3 or fewer sectors) is an approximation of the maximum number of sectors on the disk less likely to be used than groupings of more than 5 sectors. These sets of sectors are used for small files (for example small JSL, FSL, PDE files).

The Number of available sectors on disk is the total amount of disk space that is available for files.

The Largest available space on the disk is the size of the largest possible file that can be created.

The Bad block display for disk is a list of the bad sectors that were logged by the format processor's sector check during a full system generation.

You can restrict access to FCHECK with RCU.

Reorganizing usage of disk space

The following explains procedures used in manipulating and reorganizing disk space.

COMPRESS

Use this command to verify file structure, compress files, and compress directories.

File verification

COMPRESS verifies the integrity of the system's file structure by ensuring that:

- All used sectors are allocated to a file.
- All sectors allocated to files and bad blocks are marked as used.
- Each sector is allocated to only one file.
- Directories are properly linked and terminated.
- All sectors marked as used but not allocated to a file are marked as available.

During this process, the `VERIFYING SYSTEM FILES` message displays.

If multiallocated sectors are detected, COMPRESS builds the `CPR000.TMP` file. This file lists the first occurrence of multiallocated sectors. (Refer to figure 2-2.) In this example, the `FDL.TSK` file starts at sector 2032 for 178 consecutive sectors, ending at sector 2210. However, the `BAD BLOCK GROUP` file starts at sector 2189, thus creating the multiple-allocation problem.

Figure 2-2. **Disk compress error report**

DISK COMPRESS ERROR REPORT				12 JUL 98	10:35:14	000010
						000020
FILE IN ERROR	UNIT	START LBN	SIZE			000030
						000040
FDL.TSK	0	2032	178			000050
BAD BLOCK GROUP	0	2189	1388			000060

When such a situation occurs, COMPRESS terminates processing, indicating it has encountered multiallocated sectors. You should then edit and print the error log. Analysis of the particular files involved, including an assessment of which file is actually present and which file is malformed, provides sufficient information to delete the bad file, thus resolving the conflict. For instance, if a form and a font both occupy the same sector, sampling of both files indicates which is good; the bad file should then be deleted. COMPRESS can then be run.

You can use the START command to print the CPR000.TMP file. To select TRAY as the output destination use:

```
START CPR,TASKS$,,,DISC:CPR000.TMP
```

To select bin as the output destination use:

```
START CPRBIN,TASKS$,,,DISC:CPR000.TMP
```

Verification syntax COMPRESS [disk-id, [disk-id...]]V

Parameters Table 2-10 shows the parameters of the COMPRESS verification syntax.

Table 2-10. **Parameters of the COMPRESS verification syntax**

Parameter	Description
<i>disk-id</i>	Specifies which disk (DP0:, DP1:, DP2:, or DP3:) is verified. DP0: is the default.
V	Specifies that the disks are verified but not compressed.

File compression This procedure creates an ordered list of files by their location on the disk. Permanent system files are removed from the list. COMPRESS attempts to fill each free area on the disk with a file of the same size from a higher logical block number (LBN). If no such movable exact size file exists, all files between the file area and the next free area with a higher LBN are moved downward so that the first free area becomes used. This process is repeated until the number of free areas has been reduced to the minimum. The following messages are displayed when their functions are being performed:

```
Compressing files
**CRITICAL OPERATIONS UNDERWAY. DO NOT DISTURB**
**DISK WILL BE UNUSABLE IF PROCESS INTERRUPTED**
```

Compression syntax COMPRESS [disk-id [,disk-id...]]

Table 2-11 shows the parameters of the COMPRESS compression syntax.

Table 2-11. **Parameters of the COMPRESS compression syntax**

Parameter	Description
<i>disk-id</i>	Specifies which disk (DP0:, DP1:, DP2:, or DP3:) is compressed. DP0: is the default.

File compression restrictions COMPRESS displays the entered command syntax while it is running. COMPRESS stops on an ABORT command only if a noncritical operation is in process. If stopped in this manner, the disk maintains integrity and is only partially compressed.

Aborting COMPRESS is not recommended.

COMPRESS runs at logon class 2 or above. The system must be idle and the Host Interface Processor (HIP), the Xerox Print Service (XPS), and the Dynamic Document Interface (DDI) must be idle when COMPRESS is run.

Directory compression

Directory compression may occur during the file compression procedure (but not during a verification-only procedure). On an individual directory basis, the program determines if the number of empty entries in a directory allows one or more sectors to be eliminated from that directory. The entries are compacted and empty sectors are taken off the chain and released. Each directory sector is then moved to the lowest available sector on the disk. During this phase of the program the system displays the COMPRESSING DIRECTORY SECTORS message.

Disk compression restrictions

If the disk drive has a large number of files on it, you must invoke the disk save and restore (DSR) utility instead of the COMPRESS utility. The system displays a message to inform you when to use DSR.

PURGE

Use this command to clear all unused disk areas of their old contents and replace them with a meaningless pattern. Purge operates on the entire print file, as well as on the other disk space.

Syntax PURGE [D | P | F | *disk-id*]

Parameters Table 2-12 shows the parameters of the PURGE command.

Table 2-12. **Parameters of the PURGE command**

Parameter	Description
D	Purges the entire disk including the print file. D is the default if you enter PURGE without an option.
P	Purges only the print file.
F	Purges the entire disk excluding the print file.
<i>disk-id</i>	Purges the specific disk (DP0:, DP1:, DP2:, or DP3:).



Note: You can abort PURGE at any time.

Backing up and copying system files

The following explains the various functions used in backing up and copying system files.

Setting default tape device

The following explains how to set a default tape device.

Identify current TAPE & CARTRIDGE device (SUB DEV)

Use this command to identify the disk drive on the controller or peripheral cabinet which is currently associated with the TAPE keyword, or to identify the disk drive associated with the CARTRIDGE keyword. The system commands which automatically work with the drives currently associated with these keywords are:

- DSR
- MOVE
- REWIND
- SFS
- SPACE
- SST
- TAPE
- COPY TAPE
- START.

Logon level Levels 1 through 5.

Syntax SUBSTITUTE DEVICE

Parameters None.

SUBSTITUTE DEVICE (general information)

The following covers names used in association with substitute devices.

- Restrictions** The following are the restrictions to substituting a device.
- You can change device assignments only when the LPS is idle.
 - If the LPS has only one open-reel tape, one cartridge tape, or both an open-reel tape and a cartridge tape, use the general physical device names listed in table 2-13.

Table 2-13. **General physical device names**

General physical device names	Devices
CTS	Any single cartridge tape 1/2 inch 3480 cartridge (MTA or SCSI interface) 1.4 inch cartridge (QIC)
REEL	Any single SCSI open-reel tape

- If the LPS has two or more open-reel tapes or two or more cartridge tapes, use the specific physical device names listed in table 2-14.

Table 2-14. **Specific physical device names**

Specific physical device names	Devices
SREEL	Horizontal open-reel tape (SCSI interface)
FCTS	1/2 inch cartridge (if only one 1/2 inch on LPS)
SFCTS	Horizontal 1/2 inch cartridge (SCSI interface)
SCTS	1/4 inch cartridge (QIC)

Defaults

If you use the general device names on an LPS with two or more open-reel tapes or cartridge tapes, the keywords default to the tape devices as follows (and refer to figure 2-15):

- **CTS** is assigned to the physical devices on the LPS in the following order:
 - 1/4 inch cartridge (QIC)
 - Horizontal 1/2 inch cartridge (SCSI interface)
- **REEL** is assigned to the physical devices on the LPS in the following order:
 - Horizontal open-reel tape (SCSI interface)

Table 2-15. **Device keywords**

Keyword	Devices
TAPE	Default keyword used by LPS applications (for example, Editor, DSR, offline print jobs) when the keyword is omitted.
CARTRIDGE	Default keyword used by LPS applications (for example, Editor, DST, offline print jobs) when CAR is specified in the command line.

Assign a controller device to TAPE or CARTRIDGE (SUB DEV)

Use this command to assign a tape or cartridge drive on the controller to the TAPE or CARTRIDGE keyword. The system commands which automatically work with the drives currently associated with these keywords are:

- DSR
- MOVE
- REWIND
- SFS
- SPACE
- SST
- TAPE
- COPY TAPE
- START.

This command was instituted for compatibility with other software releases.

Logon level Levels 1 through 5.

Syntax SUBSTITUTE DEVICE [REEL | SREEL | CTS | FCTS | SFCTS | SCTS] FOR {TAPE | CARTRIDGE}

Parameters Table 2-16 shows the syntax parameters of the SUBSTITUTE DEVICE command.

Table 2-16. **Parameters of the SUBSTITUTE DEVICE command**

Parameter	Description
REEL	Associates keyword with the only open-reel tape drive on the controller.
SREEL	Associates keyword with the horizontal (SCSI interface) open-reel tape drive on the controller.
CTS	Associates keyword with the only cartridge drive on the controller.
FCTS	Associates keyword with the only 1/2 inch cartridge drive on the controller.
SFCTS	Associates keyword with the horizontal (SCSI interface) 1/2 inch cartridge drive on the controller.
SCTS	Associates keyword with the 1/4 inch (QIC) cartridge drive on the controller.

Defaults The following are the default syntax settings:

- The default for TAPE is REEL.
- The default for CARTRIDGE is CTS.

Examples When the three SUBSTITUTE DEVICE commands listed below are issued, subsequent OSS commands as listed below function as described.

SUB DEV REEL for TAPE	Assigns the physical device for REEL to the keyword TAPE.
SUB DEV SCTS for CAR	Assigns the 1/4 inch (QIC) device to the keyword CARTRIDGE.
SUB DEV	Displays the current SUB DEV assignments for the keywords TAPE and CARTRIDGE.
DSR or DSR TAPE	Saves disk files to the physical device assigned to the keyword TAPE.
COP CAR WRI LAB TMP	Writes files to the physical device assigned to the keyword CARTRIDGE.
STA TWOUP,RSX	Prints using the physical device assigned to the keyword TAPE.

Know the Xerox tape and cartridge format

The Xerox tape (and cartridge) format is described by the following parameters:

- Each file has a variable number of records.
- Each record is 512 bytes long.
- Records may be blocked in groups of 1 to 16 records.
- Files are separated by one end-of-file mark.
- The tape or cartridge has the standard Xerox label.

Prepare the tape or cartridge

The following covers preparation of a tape or cartridge.

TAPE [CARTRIDGE | TDn] VOLINIT

Use this command to format a tape or cartridge in the Xerox labeled format and to specify the data recording density.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] VOLINIT {,1600 | ,6250}

Parameters Table 2-17 shows the syntax parameters of the TAPE [CARTRIDGE | TDn] VOLINIT command.

Table 2-17. **Parameters of the TAPE [CARTRIDGE | TDn] VOLINIT command**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TDn	Specifies a tape device number.
VOLINIT, {1600 6250}	Initializes the tape or cartridge at 1600 or 6250 bpi.

References For background or procedural information refer to your *Xerox DocuPrint 180 LPS Operator Guide*.

Copying files

Use the copy commands to copy files from disk to disk, tape to disk, or disk to tape.

The Xerox DocuPrint 180 LPS may use open-reel tape, 1/2 inch cartridge, or 1/4 inch cartridge. In these commands:

- TAPE refers to TD0.
- CARTRIDGE refers to TD1.
- TD n refers to the previously assigned TD n device.

The Xerox DocuPrint 180 LPS open-reel tape drive handles only tapes with 15 or more feet of leader preceding the beginning of tape (BOT).

COPY (disk to disk)

Use this command syntax to create a backup copy of a disk file on disk.

Syntax COPY [*disk-id*] *input-file-id* [*disk-id*] *output file-id*

Parameters Table 2-18 shows the parameters of the copy (disk to disk) command.

Table 2-18. **Parameters of the COPY (disk to disk) command**

Parameter	Description
<i>disk-id</i>	Specifies the input file disk or the output file disk. Must be DP0: or DP1: or DP2: or DP3:.
<i>file-id</i>	Specifies the input or output file identified by <i>file-name.file-type</i> . If the output file type is omitted, the file type defaults to the file type of the input file. (Masking cannot be used for this application.)

Example COPY FORMA.FSL FORMX
CREATING FILE FORMX.FSL

Restriction The output file name must be different from the input file name unless you copy the file to a different disk with the disk-id parameter. If the output file name is the name of an existing file on the disk, then the contents of the existing file are replaced by the contents of the input file.

COPY (disk to labeled tape or cartridge)

Use this command to add new files after the existing files on a tape or cartridge. If there are no other files, the new files are written, starting at BOT.

Syntax COPY [TAPE | CARTRIDGE | TD n] WRITE [/BF: n] LABEL [$disk-id$] { $input-file-id$ | $output-file-id$ | $file-id$ | $file-type$ | ALL}

Parameters Table 2-19 shows the syntax parameters of the COPY (disk to labeled tape or cartridge) command.

Table 2-19. **Parameters of the COPY (disk to labeled tape or cartridge) command**

Parameter	Description
TAPE CARTRIDGE	Keywords specifying that files must be written to an LPS-labeled tape. TAPE is TD0; CARTRIDGE is TD1.
TD n	A tape device number.
/BF: n	Groups n disk blocks into one tape block. n can have a value of 1 to 16, and the size of the tape block, in bytes, resides in the tape label. The default blocking factor is 1, and the label indicates the blocking factor.
$disk-id$	Specifies the system disk (DP0:, DP1, DP2, or DP3:) on which the file resides.
$input-file-id$ [$output-file-id$]	Specifies the name of an input disk file and the name to be given the output tape file, in the form <i>file-name.file-type</i> . If no <i>output-file-id</i> is specified, the disk <i>file-name</i> is used for the tape file.
$file-type$	Specifies the name of a directory (CMD, CME, FNT, FRM, FSL, IMG, JDL, JSL, LGO, MSC, PDE, TMP, TST, XCS, LIB, STK, TSK, SYS, PCH, LOG, SAF, OSD, DAT, TPF, and FIS) in which all the files of that directory are written to tape.
ALL	Specifies that all user files are to be copied to tape. Files will be written to tape from the CMD, CME, FNT, FRM, FSL, IDR, JDL, JSL, LGO, PDE, TMP, TST, MSC, IMG, LIB, and STK directories.



Note: If the LPS has only a 1/4 inch cartridge, then the sysgen assigns it to TD0 which is associated with the keyword TAPE. This assignment can be changed to CARTRIDGE with the SUBSTITUTE DEVICE command.

The restoration of user files that were saved from COPY TAPE WRITE LABEL ALL command from a disk containing another version of operating system software (older or newer) to a disk containing the current level of software may result in incompatible user file type existing on the disk because of software differences.

COPY (disk to unlabeled tape)

Use this command to copy source files (such as .FSL or .JSL files) to tape. No initialization (that is, TAPE VOLINIT) is necessary to write to an unlabeled tape. A 1/4 inch cartridge, if unlabeled, must be initialized.

Syntax COPY [TAPE | CARTRIDGE | TD n] WRITE [EBCDIC | ASCII]
{*disk-id*} *file-id*

Parameters Table 2-20 shows the syntax parameters of the COPY (disk to unlabeled tape) command.

Table 2-20. **Parameters of the COPY (disk to unlabeled tape) command**

Parameter	Description
TAPE CARTRIDGE	Keywords specifying that files must be written to tape. TAPE is TD0; CARTRIDGE is TD1.
TD n	A tape device number.
EBCDIC or ASCII	Specifies the format of the tape to be written. EBCDIC is the default.
<i>disk-id</i>	Specifies the system disk (DP0:, DP1, DP2, or DP3:) on which the file resides. If no <i>disk-id</i> is specified, DP0 is the default.
<i>file-type</i>	Specifies the name of the disk file to be written to tape. (Masking cannot be used for this application.)



Note: If the LPS has only a 1/4 inch cartridge, then the sysgen assigns it to TD0 which is associated with the keyword TAPE. This assignment can be changed to CARTRIDGE with the SUBSTITUTE DEVICE command.

When writing files to an unlabeled tape, there is no position checking. The tape write always starts at the current tape position and is terminated by writing two EOFs, backspacing over one of them when another file is written. This results in one EOF between each file, with a double EOF at the end of all the files.

COPY (labeled tape or cartridge to disk)

Use this command to copy files from a tape or cartridge to disk.

Syntax COPY [TAPE | CARTRIDGE | TD n] READ[LABEL[REVIEW]
[*disk-id*] {*input-file-id* [*output file-id*] | *file-id* | NEXT | ALL | *file-type*}

Parameters Table 2-21 shows the syntax parameters of the COPY (labeled tape or cartridge to disk) command.

Table 2-21. **Parameters of the COPY (labeled tape or cartridge to disk) command**

Parameter	Description
TAPE CARTRIDGE	Keywords specifying that files must be written to an LPS-labeled tape. TAPE is TD0; CARTRIDGE is TD1.
TD n	A tape device number. 0 indicates open-reel; 1 indicates 1/4 inch; 2 indicates 1/2 inch.
[READ] LABEL	Specifies that the input file comes from the tape device number.
REVIEW	Prompts the user when the file on the tape would replace an existing file of the same name on the disk and requests whether to replace the file. If the operator responds with Y, the file on tape replaces the disk file. If the response is N, the tape skips to the next file. Entering X aborts the entire copy process.
<i>disk-id</i>	Specifies the system disk (DP0:, DP1:, DP2:, or DP3:) on which the file resides.
<i>input-file-id</i> <i>output-file-id</i>	Specifies the name of an input tape file to be copied to disk and the name to be given the output disk file, in the form <i>file-name.file-type</i> . Masking cannot be used with this application. If no <i>output-file-id</i> is specified, the file written to disk has the same name as the file tape.
NEXT	Specifies that the next file on the tape will be copied to disk.
ALL	Specifies that all the files on tape are to be copied to disk. Files will be written to disk from the CMD, CME, FNT, FRM, FSL, IMG, JDL, JSL, LGO, MSC, PDE, TMP, TST, LIB, STK, TSK, SYS, PCH, FIS, LOG, SAF, OSD, DAT, TPF, and XCS directories.
<i>file-type</i>	Specifies the name of a directory on tape to be written to disk.



Note: If the LPS has only a 1/4 inch cartridge, then the sysgen assigns it to TD0 which is associated with the keyword TAPE. This assignment can be changed to CARTRIDGE with the SUBSTITUTE DEVICE command.

Use the COPY TAPE WRITE LABEL ALL command only to save files for later restoration on a Xerox DocuPrint 180 LPS running the same OSS version so that the user files (.FSL, .JSL, .FNT, .FRM, .JDL, and .PDE files) are not corrupted. Use the COPY TAPE READ LABEL *file-type* command to restore user files saved with the COPY TAPE WRITE LABEL ALL command.

Example

COPY TAPE LABEL OCRA.FNT

Searching for file OCRA.FNT

Creating file OCRA.FNT

(if the file is a new file and does not exist on the disk)

REPLACING FILE OCRA.FNT

(if the file already exists on disk and is overwritten)

The preceding command finds and copies the file OCRA from the tape to disk. Its disk file name is OCRA, and it is cataloged in the FNT file directory. If another file exists on this tape (after the one just copied), the command, COPY TAPE LABEL NEXT copies the next file from tape and stores it in a *file-name.file-type* as specified in the tape label information.

The command, COPY TAPE [LABEL] ALL copies all the files from the Xerox-labeled tape, stores them with the same file name as on tape and catalogs them by file type. After copying the files from an LPS-labeled file tape to disk, use the TAPE REWIND or TAPE UNLOAD command to rewind the tape.

COPY (unlabeled tape to disk)

Use this command syntax to copy files from an unlabeled tape to disk.

Syntax 2 COPY [TAPE | CARTRIDGE | TD*n*] READ [EBCDIC | ASCII]
[*disk-id*] *file-id*

Parameters Table 2-22 shows the syntax parameters of the COPY (unlabeled tape to disk) command.

Table 2-22. **Parameters of the COPY (unlabeled tape to disk) command**

Parameter	Description
<u>TAPE</u> CARTRIDGE	Specifies that the input file comes from tape. TAPE is TD0; CARTRIDGE is TD1.
TD <i>n</i>	Specifies tape device number.
READ	Specifies a copy from tape to disk. This copy command defaults to read even if not specified.
<u>EBCDIC</u> ASCII	Specifies the recording format of the input tape. EBCDIC is the default.
<i>disk-id</i>	Specifies the system disk to which the file will be copied.
<i>file-type</i>	Specifies the file to be written to disk, in the form <i>file-name.file-type</i> . (Masking cannot be used for this application.)



Note: If the LPS has only a 1/4 inch cartridge, then the sysgen assigns it to TD0 which is associated with the keyword TAPE. This assignment can be changed to CARTRIDGE with the SUBSTITUTE DEVICE command.

Restrictions The following are restrictions to copying files from an unlabeled tape to disk.

- The input tape must be unlabeled, unblocked, EBCDIC or ASCII, open-reel, and 1600 bpi or 6250 bpi.
- The maximum record length is 150 characters, although only the first 72 characters of each record are entered into the disk file.
- A sequence number is placed in bytes 75 to 80 of the disk file record.
- Records shorter than 72 bytes are blank filled to 72 bytes.
- The largest file that can be created is 10,000 records.

Example COPY TAPE UNSYS.JSL

This command copies a file from an unblocked, unlabeled tape to disk.



Note: If the *file-id* used in the COPY TAPE command already exists on a disk, the file directory pointer is erased first. Then, when the tape read is successful, the new file is copied to disk.

Manipulating files

The following deals with the various ways files may be manipulated.

DELETE

Use this command to delete a file from the system.

Syntax DELETE [*disk-id*]*file-mask*[, *file-mask*,...,*file-mask*]

Parameters Table 2-23 shows the syntax parameters of the DELETE command.

Table 2-23. Parameters of the DELETE command

Parameter	Description
<i>disk-id</i>	Specifies the system disk (DP0:, DP1:, DP2:, or DP3:) which contains the file to be deleted.
<i>file-mask</i>	Specifies a file or set of files as <i>file-name.file-type</i> .

To delete a file from the system, the user must have an appropriate level of access. Refer to the "Setting file access and task executing privileges" section in chapter 1 for further information. Masking can be used with this application.

Example DELETE UNIVAC.JSL,FORMX.FSL
UNIVAC.JSL DELETED
FORMX.FSL DELETED



Caution: System (SYS) files should never be deleted.

FID (build file list)

Use this command to build a list of files. When the FID command is invoked, a prototype command file is created which contains one record for each file of a user-specified file type. FID may also be initiated from the system level.

Syntax FID [*disk-id*]*file-id file-type*

Parameters Table 2-24 shows the syntax parameters of the FID (build file list) command.

Table 2-24. **Parameters of the FID (build file list)**

Parameter	Description
<i>disk-id</i>	Specifies the system disk (DP0:, DP1, DP2, or DP3:) on which the file is created.
<i>file-id</i>	The <i>file-name</i> and <i>file-type</i> of the prototype command file to be created. .CMD is the default <i>file-type</i> . The <i>file-name</i> must be specified.
<i>file-type</i>	The specific file directory type from which the individual records of the command file are to be built. Each record of the command file consists of a semicolon in position 1, followed by a file name and type, followed by an asterisk (for example ;GBAR.FSL*). These records may then be edited to produce a command file for manipulating a subset of the current fonts in the FNT directory).

FILE (display file list)

Use this command to display the file names in each disk file directory. Specific file IDs within the catalog or the entire catalog may be displayed.

Syntax FILE {[/*EX*] [*disk-id*] [*file-id*] [*file-id*,...]} | [*disk-id*] [*file-id*] /SORT *filename* [*file-type*] | [*disk-id*] [*file-id*] /*filename* [*file-type*], [*leading string*], [*trailing string*] }

Parameters Table 2-25 shows the parameters of the FILE (display file list) command.

Table 2-25. **Parameters of the FILE (display file list) command**

Parameter	Description
/EX	Displays all the file characteristics of a file id. The display format is a title line identifying the information in each column. Note that when /EX is selected, the leading and trailing strings are not displayed.

FILE NAME	UNIT	FSIZE	RSIZE	FFBY	EOF	DISK ADDR
XRK.IMG	2	6	512	512	6	472522
KI1.IMG	2	8	512	0	8	472530
KI2.IMG	2	21	512	0	21	472540

FILE NAME	A ten-character field indicating file name and file-type.
UNIT	The system disk (DP0:, DP1:, DP3:) on which the file resides.
FSIZE	A ten-character decimal field indicating the number of blocks allocated to the file.
RSIZE	A six-character decimal field indicating the number of bytes in a record.
FFBY	A six-character decimal field indicating the first free byte in the last block of the file.
EOF	A ten-character decimal field indicating the end-of-file block number.
DISK ADDR	A ten-character octal field indicating the starting address of the file on the disk.

If the /EX option is not entered, only the file IDs are displayed. Note that when /EX is selected, the leading and trailing strings are not allowed.



Note: Extended file information is not displayed for licensed fonts and logos or secured files.

Parameter	Description
<i>disk-id</i>	Identifies which system disk (DP0:, DP1:, DP2:, or DP3:) contains the file id(s) to be printed.
<i>file-id</i>	Refers to the name of a file and the file directory in which it resides, in the form <i>file-name.file-type</i> . If no <i>file-id</i> is specified, the default files are all the files in the directory. (Masking can be used with this application.)
/SORT	Alphabetically sorts a list of files by type to be printed or displayed. The number of files that can be sorted with the FILE[/SORT] command is limited by the availability of dynamic memory. When this limit is reached, the message <i>File/ list too long, only first xxxx File names will be sorted.</i> is displayed. Sorting should be done by file types for large disk or tape volumes.
<i>leading string</i>	A string of any characters (except commas) which is added before each file name in the list that is created.
<i>trailing string</i>	A string of characters (except commas) that is added after each file name in the list that is created. The total number of leading and trailing characters cannot be greater than 50.
<i>/file-name[file-type]</i>	A keyword specifying a file created on disk which contains a record for each file listed in the following order: <i>Lead str file-id Trailstr</i> . If <i>file-type</i> is omitted, the file is set up in the directory .TMP.
<i>,file-name[file-type]</i>	Specifying a file created on disk which contains the alphabetically sorted list of tape files. If the <i>file-type</i> is omitted, the file is set up with the <i>file-type</i> .TMP.

The total number of files displayed as well as the total number of disk sectors used by these files is accumulated and displayed by file type.

The Total of *n* sectors in *m* files message displays as each parameter on the file command is processed.

Parameter	Description
<i>n</i>	The sum of the FSIZE field for all the files of a <i>file-type</i> requested by a parameter on the FILE command.
<i>m</i>	A decimal number that represents the number of files for the given <i>file-type</i> .
The Grand total of <i>n</i> sectors in <i>m</i> files message is displayed when all the parameters on the FILE command are satisfied.	
<i>n</i>	The sum of the FSIZE field for all the files requested by the parameter string of the FILE command.
<i>m</i>	A decimal number that represents the total number of files requested by the parameter string of the FILE command

If during the search of the system file directories, file catalogs are found to contain no files, this information is saved until all the parameters of the FILE command have been completed. The No files in type *file-type* message displays.

Example FILE DP0: *.SYS/SORT, SYSFIL.TMP

This command creates a file called "SYSFIL.TMP" which lists all the SYS files on the DP0: disk. The filenames in SYSFIL.TMP are sorted in alphabetical order, and SYSFIL.TMP is displayed on the screen. If the screen display of the sorted records is interrupted, then the created file "SYSFIL.TMP" lists only the displayed file names.

FIX (modify font or logo file)

Use this command to modify licensed and standard fonts and logos and to format the font/logo matrix for the configured imaging hardware.

Syntax FIX [*file-id.file-type* | *file-id* [,*file-id*...]]

Parameters Table 2-26 shows the syntax parameters of the FIX command.

Table 2-26. **Parameters of the FIX (modify font or logo file) command**

Parameter	Description
<i>file-id</i>	Refers to the name of a file and the file directory (must be FNT or LGO) in which it resides, in the form <i>file-name.file-type</i> . (Masking can be used for this application.)
<i>file-type</i>	Specifies the file directory type (must be FNT or LGO) on which the fix process is performed.

When no parameters are specified, the system defaults to all fonts and all logos in the font and logo file catalogs.

REVIEW (review files)

Use this command to review each file in a directory with the option of deleting it.

Syntax REVIEW [*disk-id*]{*file-id* [,*file-id*,...]}

Parameters Table 2-27 shows the syntax parameters of the REVIEW command.

Table 2-27. **Parameters of the REVIEW command**

Parameter	Description
<i>disk-id</i>	Refers to the system disk (DP0:, DP1:, DP2:, DP3:) that contains the specified directory.
<i>file-id</i>	Refers to the name of a file and the file directory in which it resides. The form used is <i>file-name.file-type</i> as previously defined. To delete a file from the system, the operator must have an appropriate level of access; refer to the "Setting file access and task execution privileges" section in chapter 1. (Masking can be used for this application.)

After the operator requests the set of file names in a file type displayed, the file names are displayed one at a time. The operator has the option of deleting a file by entering D or exiting the display process by entering the X command.

Any response other than D or X (for example pressing the space bar) results in the current file being retained and the next file name being displayed. Any file name containing a period or dollar sign cannot be deleted and is displayed with the message *PROTECTED*.

RNAME (rename a file or directory)

Use this command to change the name and/or directory type of a file. After the file name has been changed, the original name is deleted from the file directory.

Syntax RNAME *old-file-id new-file-id*

If the file-type part of the new file-id is not specified, the file type is the same as the old file. (Masking does not apply to this command.)

TYPE

Use this command to display a source file without editing the file.

Logon level Levels 1 through 5.

Syntax TYPE [*disk-id*]*file-id*

Parameters Table 2-28 shows the syntax parameters of the TYPE command.

Table 2-28. **Parameters of the TYPE command**

Parameter	Description
<i>disk-id</i>	Identifies the system disk (DP0:, DP1:, DP2:, or DP3:) which contains the <i>file-id</i> to be displayed.
<i>file-id</i>	Refers to the name of a file and the file directory in which it resides. The form used is <i>file-name.file-type</i> as previously defined.

Each source file has line numbers associated with its data records. These line numbers are supplied by the editor during file creation. Editable files created by the COPY TAPE READ command from open-reel tape have line numbers added to them at the time the file is created. Source files downloaded from the online subsystem through HOSTCOPY must have line numbers added with the editor (refer to the MERGE command).

Only the uppercase letters A to Z, and the digits 0 to 9 can be used in source file names.

List tape or cartridge files

The following explains the procedure for listing tape or cartridge files.

TAPE [CARTRIDGE | TDn] LIST

Use this command to display or to create a disk file of file-names and sort them alphabetically or add leading or trailing strings to facilitate future command file manipulation of that tape or cartridge.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] LIST [,PRINT]/EX {[[/SORT,*file-name* [*.file-type*]] | /*file-name*[*.file-type*],[*leading string*],[*trailing string*]]}

Parameters Table 2-29 shows the syntax parameters for creating tape or cartridge file lists.

Table 2-29. **Parameters of the TAPE [CARTRIDGE | TDn] LIST command**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TDn	Specifies a tape device number.
LIST[,PRINT]	Lists the <i>file-ids</i> of the labeled tape, beginning at the current tape position. If PRINT is omitted, the tape directory is displayed.
/EX	Prints all the file characteristics of a given file id. Note that when /EX is selected, the leading and trailing strings are not allowed. If the /EX option is not entered, only the file IDs are printed.
/SORT	Alphabetically sorts a list of file names of the files on tape. The number of files that can be sorted is limited by the availability of dynamic memory. When this limit is reached, the message FILE/LIST TOO LONG. ONLY FIRST xxxx FILENAMES WILL BE SORTED. is displayed. Sort by file types for large disk or tape volumes.
, <i>file-name</i> [<i>file-type</i>]	Specifies the file created on disk that contains the alphabetically sorted list of tape files. The default <i>file-type</i> is .TMP.
/ <i>file-name</i> [<i>file-type</i>]	Specifies the file, created on disk, which contains a record for each file listed in the following order: Lead str <i>file-id</i> Trailstr. The default <i>file-type</i> is .TMP
<i>leading string</i>	A character string (except commas) that is added before each <i>file-name</i> in the list that is created.
<i>trailing string</i>	A character string (except commas) appended to each <i>file-name</i> in the list that is created. The total number of leading and trailing characters cannot be greater than 50.

Example TAPE LIST,PRINT /SORT, *.FRM FRMFIL.LST

This command creates a file called "FRMFIL.LST" which lists all the .FRM files on the mounted tape. The file names in FRMFIL.LST are sorted in alphabetical order, and FRMFIL.LST is printed. If the screen display of the sorted records is interrupted during a pause by the X command, then the created file "FRMFIL.LST" lists only the displayed filenames.

Listing begins at the current tape position and continues to the end of the tape. (The listing may be stopped by entering the X command when the pause occurs on the display.) If the PRINT option is specified, the operator is asked to enter a tape volume ID. The ID can be 1 to 30 characters. If more than 30 characters are entered, only the first 30 characters are used. The following title line appears on each page of the printed listing:

```
Tape directory    date    time    Tape ID:    id string
```

The following header line precedes the tape file information on both the screen and printed listings:

```
File No.    File Name    File Size    'BF'    Date
```

Table 2-30. File header description for FMFIL.LST listing

File header	Description
FILE NO.	The decimal number that indicates the numerical position of the file on the tape.
FILE NAME	The <i>file-id</i> .
FILE SIZE	The number of 512 byte disk sectors.
- BF -	The number of 512 byte disk sectors in 1 tape block.
DATE	The date the file was written to tape.

If a valid tape label cannot be found, a number appears in the FILE NO. field but the other fields are blank. This is the case when listing a system software tape (SST). The first two files of the SST do not have labels (with some unlabeled tapes, other meaningless characters may appear). A string of asterisks appearing in the date field means no date was specified in the label or that it could not be converted properly.

To print the file, exit from the editor (using the END command). You may spool editor print files. The current working storage file is intact if you reenter the editor after the file is printed. The listing is in portrait mode.

Locate tape or cartridge files

The following describes how to locate a file on tape or cartridge.

TAPE [CARTRIDGE | TDn] FIND (find file)

Use this command to position the tape or cartridge preceding the label record of the file. You can then issue commands to manipulate this file.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] FIND [,file-id]

Parameters Table 2-31 shows the syntax parameters of the TAPE [CARTRIDGE | TDn] FIND (find file) command.

Table 2-31. **Parameters of the TAPE [CARTRIDGE | TDn] FIND (find file) command**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TDn	Specifies a tape device number.
FIND, file-id	Positions an LPS-labeled tape or cartridge at the specified file.

TAPE [CARTRIDGE | TDn] NEXT (move to next file)

Use this command to position the tape or cartridge in order to manipulate the next file on the tape or cartridge.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] NEXT

Parameters Table 2-32 shows the syntax parameters of the TAPE [CARTRIDGE | TDn] NEXT (move to next file).

Table 2-32. **Parameters of the TAPE [CARTRIDGE | TDn] NEXT (move to next file)**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TDn	Specifies a tape device number.
NEXT	Displays name of the next file on the tape or cartridge.

Restriction The system ignores a second NEXT command until it executes another TAPE [CARTRIDGE] command.

TAPE [CARTRIDGE | TDn] SKIP (skip to file)

Use this command to position the tape or cartridge after the specified number of files.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] SKIP [,n]

Parameters Table 2-33 shows the syntax parameters of the TAPE [CARTRIDGE | TDn] SKIP (move to 1st or 2nd... file) command.

Table 2-33. **Parameters of the TAPE [CARTRIDGE | TDn] SKIP (move to 1st or 2nd... file) command**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TD <i>n</i>	Specifies a tape device number.
SKIP	Positions the tape or cartridge after the <i>n</i> th file.
<i>n</i>	A positive integer which specifies the number of tape marks the tape or cartridge is past.

TAPE [CARTRIDGE | TDn] EOF or NOEOF or ENDFILE

Use this command to control the writing of tape marks between files for LPS labeled and unlabeled tapes.

Logon level Levels 1 through 5.

Syntax TAPE [CARTRIDGE | TDn] {NOEOF | EOF | ENDFILE }

Parameters Table 2-34 shows the syntax parameters of the TAPE [CARTRIDGE | TDn] EOF or NOEOF or ENDFILE command.

Table 2-34. **Parameters of the TAPE [CARTRIDGE | TDn] EOF or NOEOF or ENDFILE command**

Parameter	Description
TAPE CARTRIDGE	TAPE is TD0; CARTRIDGE is TD1.
TDn	Specifies a tape device number.
NOEOF	Disables the writing of tape marks between files. This command must precede the list of file names to be written to tape.
EOF	Writes two tape marks and then backs up over one of the tape marks.
ENDFILE	Writes the file ENDFIL.END and ends the tape with two tape marks. It then backs up over one of the tape marks. Also concatenates files without tape marks between logical files. There is no restriction on files or file-types that can be copied.

Restrictions The following are the restrictions to writing tape marks.

- If tape mark writing is disabled using NOEOF and you must enable it again using EOF or ENDFILE, the tape will not contain any tape marks to signify the end of the tape. A tape must normally end with two tape marks.
- If NOEOF is selected and the end of tape is reached while writing a file, the LPS marks the tape two times in that area, and the last file will be corrupted.
- The EOF and NOEOF options are available only from the EDIT> prompt (a text editing session) or from an editor command file.

Examples The following are examples of usage of the syntax parameters.

```
Use of NOEOF and ENDFILE
EDIT> TAPE NOEOF
EDIT> COPY TAPE WRITE LABEL ABC.IMG
EDIT> COPY TAPE WRITE LABEL ABD.IMG
EDIT> COPY TAPE WRITE LABEL BCD.IMG
EDIT> TAPE ENDFILE
```

Three .IMG files are copied to tape with no tape marks written between the .IMG files. TAPE ENDFILE enables the writing of tape marks, writes the ENDFIL.END file to tape, writes two tape marks, and backs up over one of them.

Use of NOEOF and EOF

```
EDIT> TAPE NOEOF
EDIT> COPY TAPE WRITE LABEL ABC.IMG
EDIT> COPY TAPE WRITE LABEL ABD.IMG
EDIT> COPY TAPE WRITE LABEL BCD.IMG
EDIT> TAPE EOF
```

Three .IMG files are copied to tape with no tape marks written between the .IMG files. TAPE EOF writes two tape marks and backs up over one of them.

Recommendation

Use the ENDFILE option to prepare a tape to be used by the HOSTCOPY command.

For example, you can create a back up tape with all user files concatenated into one file (multiple logical files in one physical file), thereby eliminating the need for multiple job steps when executing HOSTCOPY to download these files.

This command is provided for compatibility with the HOSTCOPY procedure, which requires that the end of the last physical file on the tape be identified with a logical file called ENDFIL.END. This file results in the message END OF TAPE FOUND. IS ANOTHER FILE TO BE SENT to be displayed during HOSTCOPY.

Backing up system files to tape or cartridge

The following explains the procedures for backing up system files to tape or cartridge.

DSR (disk save and restore)

The disk save and restore (DSR) process utility operates on the entire system in two steps: it creates a tape backup of all system files, and it restores those files to disk, erasing all previous disk contents. Use this command after sysgens and software updates to back up the entire system on tape, and after a system crash or the loss of system files to restore the entire system. As a non-print task, DSR runs only if the system is idle

DSR is run by two separate programs: DSR.TSK and DSR.SAF. DSR.TSK resides in the operating system and executes the save function using system controller disk, tape, and terminal services. DSR.SAF executes the restore function which rebuilds the system controller. It is a standalone program which you boot from the backup tape. DSR.SAF is a byte-by-byte copy program which runs its own disk, tape, and terminal service routines.

The disk or disks whose files are copied to tape are the *source disks*. The disk or disks to which the files are restored are the *object disk(s)*.

- PC UI window mode** If you save a DSR tape from the PC UI window mode, ensure that you restore the tape when the DocuPrint 180 LPS is running in the PC UI window mode.
- Character mode** If you save a DSR tape from the character mode (either from the PC UI or from the terminal), ensure that you restore the tape when the 4635 LPS is running in character display mode.

Starting DSR

The following explains how to start DSR.

DSR save syntax DSR [TAPE | CARTRIDGE | TDn] [1600 | 6250][,*disk-id* [*disk-id...*]][,L]

DSR save parameters Table 2-35 shows the DSR save syntax parameters.

Table 2-35. **Parameters of DSR save**

Parameter	Description
<u>TAPE</u>	Specifies TD0.
CARTRIDGE	Specifies TD1.
TDn	Specifies the previously assigned TDn tape device.
1600 <u>6250</u>	Specifies a tape recording density. The default is always 6250 bpi, even though the tape has been initialized at 1600 with the TAPE VOLINIT command. This parameter has no effect if the medium is cartridge.

Table 2-35. Parameters of DSR save (continued)

Parameter	Description
<i>disk-id</i>	Indicates the disk (DP0:, DP1:, DP2:, or DP3:) whose contents are written to tape. If a DPn: is omitted, all disks are backed up.
L	Lists file information on the display as the system files are copied to tape. When several disks are backed up, this list can be extensive. No list is the default.

DSR save restrictions

The DSR save restrictions are:

- If HIP is not active and jobs are pending in the queue, DSR runs, but it loses the HIP queue because HIP uses addresses unavailable during compression.
- If a specified drive does not exist, the `DSxxxx Invalid Parameter...DSR aborted. message` is displayed, and DSR terminates. (A complete list of messages related to DSR is in the *Xerox DocuPrint 180 LPS Message Guide*.)
- If the save tape is not mounted with the tape drive and online at task invocation, DSR displays the message `DSxxxx Mount and ready DSR tape`. Once DSR senses the tape drive is online and at BOT, it proceeds with the save process.

DSR restore

The restore processor DSR.SAF is the first file written to tape by the save process, DSR.TSK. DSR.SAF is a bootable, standalone task.

You initiate the restore task by loading the tape and pressing the Boot switch. Next, on a terminal the system displays the prompt `READY $` and you enter `BT`. On a PC UI the system displays a boot menu and you select the appropriate tape device. Then, the system loads DSR.SAF from the tape device. DSR.SAF displays a restore menu from which you select an option which it executes.

If the print file `PRFIL1.SYS` is copied to the object disks by DSR.SAF, the `REALLOCATE` task runs automatically to generate working space for the print file.

DSR restore restrictions

The DSR restore restrictions are:

- A sysgen is not required by the DSR restore task, but ensure that you format and check the object disk for bad blocks before running DSR.SAF.
- The disk's entire file structure is saved and restored; DSR does not process individual files.
- Access DSR through logon level 2. (Refer to the "Protecting access to files" section in the "Setting LPS parameters" chapter.)
- DSR does not distinguish between machine-specific files (for example `MCHID$.SYS`, the machine I.D. file) and files which can be copied from one system to another.

(Use the single file transfer utility, `SFT`, to prevent billing problems when you move files between systems.)

File protection

To invoke DSR.SAF, load the tape and enter Y after the following prompt:

```
DSR Vxx (date of version)
DSR RESTORE IN COMPRESS MODE
ARE YOU SURE ('Y' OR 'N')
```

DSR.SAF first verifies that the source and object disk are the same type. If they are different, DSR displays `DISK TYPE INCOMPATIBILITY` and aborts the restore process. Next DSR compares the source disk id with the object disk id. If their disk IDs are different, DSR prompts you to confirm that the object disk files should be overridden by the source disk files.

```
SOURCE DISK ID: diskid
OBJECT DISK ID: diskid
CONFIRM BEFORE OVERWRITING (Y,N):
```

If the object disk has no id, DSR displays the following messages:

```
SOURCE DISK ID: diskid
OBJECT DISK HAS NO ID
CONFIRM BEFORE OVERWRITING (Y,N):
```

If you enter Y, the restore task executes. DSR reconstructs the file directory chains, loading all directory sectors of the same file type contiguously at the lowest possible disk address.

Once the directories are reconstructed, the file images are consolidated into groups of similar file types. The restoration process can cause disk fragmentation. Particularly, if a disk being restored is near capacity, one or more files may not be restored.

Disk space for restore

You can restore any one disk, a combination, or all disks saved to tape. The following displays after you confirm the restore request.

```
SELECT DISK(S) TO RESTORE
0.  RESTORE  DP0:
1.  RESTORE  DP1:
2.  RESTORE  DP2:
3.  RESTORE  DP3:
4.  RESTORE  ALL THE ABOVE DISKS
5.  ABORT RESTORE PROCESS
ENTER OPTION(S):
```

You can select one or more items by keying in the corresponding item number. For example, `ENTER OPTION(S): 0, 1, 3.`

If you select option 5, DSR displays the message, `RESTORE PROCESS IS ABORTED. DSR EXIT.` and exits.

Since the bad block information on the object disk ultimately dictates where files are (and are not) restored, DSR verifies whether the disk has a file listing its bad blocks. If no bad block is found, DSR displays: `UNABLE TO LOCATE BAD BLOCK DATA. REFORMAT DISK` and aborts the restore process.

Once the DSR restoration process is complete, the following message is displayed.

```
*****
* NOTE:  ONE OR MORE FILES WERE NOT RESTORED*
          DUE TO INSUFFICIENT DISK SPACE
*****
```

Multivolume files

Sometimes the data stored on a disk volume does not fit on a single tape reel written at 1600 bpi. Therefore, DSR supports multivolume save and restore. DSR writes a volume label at the beginning of each tape. In the label is a sequence number. If end of volume (EOV) is encountered in the save process, DSR writes a tape mark, rewinds and unloads the tape, increments the next sequence number, and displays: MOUNT AND READY DSR VOLUME n, where n is the next sequence number. During the restore process, if DSR encounters EOV and a single tape mark, it rewinds and unloads the tape, and displays: MOUNT AND READY DSR VOLUME n. When the next volume is mounted, DSR verifies that the correct volume is mounted. If it is not, DSR unloads the tape and displays: VOLUME SEQUENCE ERROR. MOUNT DSR VOLUME n.

Device error handling

DSR responds to all device errors encountered during the save and restore processes. The degree of error recovery differs between the save and restore processes. Refer to the *Xerox DocuPrint 180 LPS Message Guide* for a more complete listing of DSR error/recovery messages.

Aborting the save process

If an abnormal device error persists after a number of retries, DSR displays DISK ERROR xxx...DSR ABORTED or TAPE ERROR xxx...DSR ABORTED, where xxx is a three-character mnemonic representing the error returned by the device driver to DSR. Since no files are lost by terminating an in-progress save to tape, DSR aborts the save process and exits.

Incomplete restoration

If a file being restored from tape is too large to fit in the largest free contiguous space on the system disk, the FILE: <filename.typ> NOT RESTOREDINSUFFICIENT DISK SPACE message displays.

If this occurs, then the following message displays when the restore has completed.

```
*****
* NOTE:  ONE OR MORE FILES WERE NOT RESTORED*
          DUE TO INSUFFICIENT DISK SPACE
*****
```

Aborting the restore process

The restore task cannot be aborted without the risk of losing system files. Therefore DSR.SAF does not provide you with the option of aborting the restore.

If a tape device error is encountered during the restore process, DSR retries the tape operation several times. If an error is identified as irrecoverable, DSR displays `TAPE ERROR XXX . . . DSR ABORTED. REFORMAT DISK` and aborts the process.

Special file handling

The following lists specialized file types and how they are handled during DSR save and restore.

Licensed fonts and logos

Licensed fonts and logos are encrypted as they are written to tape during the save process and are decrypted as they are restored to disk.

When copying licensed fonts and logos from disk to tape, use the `COPY TAPE WRITE LABEL` command.

Bad block files

DSR accesses the bad block information resident on the object disk at the time of the restore process in order to avoid writing files to known corrupted regions on the disk. The bad block files on the object disk are not altered during the restore process and are incorporated into the new file structure.

Print file and its history

The print file and print permanent history file are rebuilt after the restore process is completed.

Recovery files

The two recovery files `PFILR$.SYS` and `RECTK$.SYS` are copied to tape and restored by DSR.SAF.

Core image file

The core image file is written to tape by the save process and is transferred to the object disk during the restore process.

Copying a DSR file to the LPS

The following describes how to copy a DSR file to the LPS.

SFT (single file transfer utility)

The single file transfer (SFT) utility allows the transfer of single files from a DSR tape to a system disk drive. The file control processor (FCP) and disk space availability dictate the destination of the transferred file.

SFT procedure

Step 1 To invoke the SFT task, at the OS1000 READY FOR COMMANDS prompt, enter:

```
RCU SFT
SFT
```

The screen displays the prompt shown in figure 2-3.

Figure 2-3. **SFT welcome prompt**

```
Welcome to SingleFile Transfer program.
1.)Transfer a single file to disk.
2.)Select a device.
3.)Exit SFT program.
```

The system then prompts you with the Make a selection: message.

If you enter 1, the system responds with: Mount and Ready DSR tape and press <RETURN>.

If you enter 2, the system displays the prompt shown below.

Figure 2-4. **Device selection prompt**

```
Device Selection Table
1.)TD0
2.)TD1
3.)TD2
Make a selection:
```

After you select one of the devices, the system again displays the following prompt, shown in figure 2-3.

If you enter 3, the tape unloads automatically and the system displays the Exit SFT task... message.



Note: If you enter 1, but the DSR tape is not mounted and ready, the system responds with the message: Verify tape is Ready and online and press <RETURN> to continue or <X> to exit.

If you enter X, the message Exit SFT task appears on the display, the system exits the SFT task and returns to the OS1000 READY FOR COMMANDS prompt.

Step 2 Once the tape is mounted and ready, the system displays the message: Please enter file name.

If you enter X, the message Exit SFT task appears on the display, the tape automatically unloads, and the system exits the SFT task, returning to the OS1000 READY FOR COMMANDS prompt.



Note: The system responds with various error messages when the file names are incorrect or are entered incorrectly. The messages then prompt you for further action.

Once you enter the file name and type, the system prompts: Is XXXXXX.XXX the file you requested (Y/N)?

If you enter Y, the system displays Searching for XXXXXX.XXX.

If you enter N or press any key, the system responds with Would you like to continue SFT program (Y/N)?

When you enter Y, you are requested to enter another file name.

When you enter N or press any key, the system unloads the tape, exits the SFT task, and returns to the system prompt.

Step 3 When the file you specified is found on the DSR tape, the messages shown in figure 2-5 are displayed.

Figure 2-5. **SFT successfully completed**

```
Creating XXX.XXX on disk.
File XXXXXX.XXX created on disk. (This occurs
after the file is written to the disk.)
Exit SFT task...
```

After exiting SFT, the tape automatically rewinds and the system prompt appears on the display.

If the requested file is not found on the DSR tape, the messages shown in figure 2-6 are displayed.

Figure 2-6. **SFT did not find file**

```
Reached end of DSR tape...
File XXXXXX.XXX not found on tape.

Exit SFT task...
```

Restrictions The following are the restrictions to the SFT.

- Only one file can be copied from tape to disk.
- When writing a file to disk, SFT deletes the file on the disk (if it already exists) and writes the file from the DSR tape to the first location on the disk which can accommodate the file's size.
- Specify the full file name: SFT does not support wildcards.
- Do not copy .OSD, .\$Y\$, .SAF, .SYS, .TSK, or .LOG files.
- If two files have the same name on the DSR tape, SFT writes only the first file to disk.

- If a system disk is full, SFT writes the specified file to the next disk.
- The system disks must be structurally intact and consistent.

Messages

When restoring a single file from a multivolume DSR tape, if the end of the first volume is reached and the requested file is not found, the system prompts you to mount the next volume and press RETURN.

If an incorrect volume is loaded, the system messages you that the wrong tape has been loaded and prompts you to mount the correct tape volume.

If you attempt to restore a protected file type, the system responds with the following message, shown in figure 2-7.

Figure 2-7. **SFT and protected files**

```
Check file type entered.  File types $Y$, SAF,
TSK, LOG, and SYS are protected from copying to
disk.  Would you like to continue SFT program (Y/
N)?
```

If you enter Y, you are requested to enter another file name. If you enter N or press another key, the tape unloads and SFT terminates.

If a file on the tape has the same name as a file on the disk, the following message appears, shown in figure 2-8.

Figure 2-8. **SFT and duplicate file names**

```
Problem renaming XXXXXX.XXX on disk.
Exit SFT task due to error...
```

If no file storage space is available, the system responds with the messages: Disk(s) full... Exit SFT task due to error...

When the tape drive goes offline, the system displays the message: Tape drive is not operational...

If you mount a tape not in DSR format, the system responds with the message: Invalid DSR tape... Mount volume XX and press <RETURN> to continue.

If a hardware malfunction occurs during the restoration process, the system displays the message: Fatal hardware error... Exit SFT task due to error...

Backing up disk files to PC UI

The following section covers procedures for backing up disk files to PC UI.

Know LPS disk file attributes

Files on your LPS disks maintain attributes such as record size, line numbers, and character encoding format information. When LPS files are transferred to the PC UI hard disk, a File Description Block (FDB) is automatically created and maintained. The PC UI requires the FDB information in order to transfer, convert, or edit files from the LPS disks. LPS files backed up to floppy and imported to the PC UI do not have the FDB. The PC creates FDB data for these floppy files by prompting the operator for information and by referencing a profile of file attributes typically associated with the imported file's file-extension.

Convert the console log

You can convert the console log into an ASCII text file. FDB data is created automatically for the converted console log. By converting the log you can examine and manipulate the actual text with your own software. Because the converted log may be quite large, you may need to copy it to PC UI floppies using the MS-DOS BACKUP command, from the MS-DOS command line.



Caution: FDB data should never be copied to PC UI floppy using the MS-DOS BACKUP command, as it may corrupt the PC UI FDB data if restored.

Convert files with supported formats

You can convert files with the supported formats if FDB data for the source file is available. FDB data is created automatically for the converted file. The supported formats are:

- PC ASCII
- UNIX compatible ASCII
- LPS format with limited line and file lengths
- EBCDIC format used by mainframe computers.

If the converted file fits on one PC UI floppy, its FDB data is also copied by the system. However, if multiple PC UI floppies are required, it is necessary to copy the file to PC UI floppies with the MS-DOS BACKUP command after conversion, from the MS-DOS command line.



Caution: FDB data should never be copied to PC UI floppy using the MS-DOS BACKUP command, as it may corrupt the PC UI FDB data if restored.

Delete LPS or PC UI floppy files

You can delete files on the LPS or PC UI floppy. If one of these files is deleted, the FDB data is automatically erased.

Copy LPS or PC UI floppy files

You can copy files from the LPS or PC UI floppy to the PC UI. If one of these files is copied, the FDB data is automatically copied.

List LPS or PC UI floppy files

You can review on the PC UI a list of files from the LPS or PC UI floppy.

Transfer files between the LPS and PC UI

You can transfer files from the LPS to the PC UI and vice versa. Files transferred from the LPS have FDB data created automatically. This data reflects the LPS file attributes. In order to use these files, you may have to convert to another format, for example, PC ASCII, after transfer.

Files transferred to the LPS use FDB data to determine how to store the file on the LPS. For correct usage by the LPS, you may have to convert to another format, for example, LPS, before transfer. During the file transfer operation you can:

- Overwrite the file on the destination system.
- Rename the file on the LPS if the LPS is the destination system.

The LPS does not have to be idle in order to run file transfer.

Restrictions These are the restrictions for transferring files between the LPS and PC UI.

- One file at a time is transferred from the LPS to the PC UI, and vice versa.
- LPS files transferred to the PC UI are no longer secure.

Managing files on floppy disks

Use floppy disks to back up and restore system disk files. Use these commands to format and initialize the floppy disk, save or restore files, delete or add files, and display or print the file directory contained on the floppy disk. FLOPPY may be initiated at system level or from the editor. Unless otherwise noted, user files can be copied to the floppy disk from any of the following directories on the system disks: CMD, CME, FNT, FRM, FSL, IMG, JDL, JSL, LGO, MSC, PCH, PDE, TMP, TST, XCS, LOG, ICT, IDR, FIS, TPF, DAT, OSD, SAF, LIB, STK, SYS, and TSK.

Restriction A floppy disk must be formatted and initialized the first time it is used.

FLOPPY FORMAT

Use this command to format and initialize a floppy disk for use on your LPS. The floppy disk has 1404 available sectors when formatted. The sector configuration is 78 cylinders, 2 tracks per cylinder, 18 sectors per cylinder, and 512 bytes per sector. The format mode is double-sided, high-density.

Syntax FLOPPY FORMAT

Procedure description

The format process searches for a valid master directory, which contains a volume id of 1 to 30 characters. If a master directory is not found and the system determines that the floppy disk is not an LPS format configuration, the `This diskette is in an unknown format Should formatting continue Y/N: message` is displayed.

If you enter Y, the floppy disk is formatted according to the LPS configuration. If you enter N, the format process ends.

If a master directory is not found and the floppy disk is configured in the LPS format, the system displays `Diskette volumeID not available should formatting continue Y/N:`

If you enter Y, the floppy disk is formatted. If you enter N, the format process ends.

When the master directory is found, the message `Diskette volumeID: volumeid Should formatting continue Y/N:` is displayed. A Y formats the floppy disk; an N ends the format process.

FLOPPY SAVE

Use this command to copy one or more files from the system disk to floppy disk.

Syntax FLOPPY SAVE [REVIEW][*disk-id*] {*input-file-id*[*output-file-id*] | *file-id* | *file-type* | ALL }

Parameters Table 2-36 shows syntax parameters of the FLOPPY SAVE command.

Table 2-36. **Parameters of the FLOPPY SAVE command**

Parameter	Description
REVIEW	Prompts you to confirm overwriting a floppy file by a system file. If you enter Y, the file on the floppy disk is overwritten. If you enter N, the specified process skips to the next file on the system disk.
<i>disk-id</i>	Identifies which system disk (DP0:, DP1:, DP2:, or DP3:) contains the files to be saved. (If no disk id is specified, all disks are searched.)
<i>file-id</i> <i>input-file-id</i> <i>output-file-id</i>	Specifies the file as file-name.file-type. (Masking can be used if different file IDs are not specified for input and output.)
<i>file-type</i>	Specifies one of the following: CMD, CME, FNT, FRM, FSL, ICT, IDR, IMG, JDL, JSL, LGO, LIB, MSC, PDE, STK, TMP, or TST.
ALL	Saves all user files on the floppy disk: CMD, CME, FNT, FRM, FSL, ICT, IDR, IMG, JDL, JSL, LGO, LIB, MSC, PDE, STK, TMP, and TST.

Procedure Enter the appropriate FLOPPY SAVE command. The system displays: Saving XXX.XXX. Data is stored onto the diskette. If more than a single floppy disk is required, the following is displayed:

```
** Floppy drive I/O error on output
** Unable to open output file XXX.XXX
** Disk Saturated
ENTER:      C CONTINUE COPYING
           X ABORT SAVE PROCESS
```

User enters C and the system displays:

Have you inserted new diskette (Y/N)?

User enters Y and the system displays:

```
Diskette volume ID: XXXXXXXX
Should formatting continue (Y/N)?
```

If formatting is required enter Y. If not, enter N. The system displays:

```
ENTER:      C CONTINUE COPYING
           X ABORT SAVE PROCESS
           I INSERT NEW DISKETTE AND CONTINUE
```

User enters C, the diskette is formatted and the user enters the floppy ID at the prompt. The system displays:

```
ENTER:      C CONTINUE COPYING
           X ABORT SAVE PROCESS
           I INSERT NEW DISKETTE AND COUNTINUE
```

User enters C and the FLOPPY SAVE command continues until all require data is stored on floppy.

If you enter X, the copy process terminates.

FLOPPY DELETE

Use this command to erase a file on the floppy disk.

Syntax FLOPPY DELETE *file-id*

Parameters Table 2-37 shows the syntax parameter of the FLOPPY DELETE command.

Table 2-37. **Parameters of the FLOPPY DELETE command**

Parameter	Description
<i>file-id</i>	Specifies the file to be erased. Masking is permitted.

FLOPPY CLEAR

Use this command to delete all files (except \$Y\$) from the master file directory on the floppy disk.

Syntax FLOPPY CLEAR

Restriction FLOPPY CLEAR functions correctly only if an LPS with V3.6 OSS or higher has initialized the floppy.

FLOPPY FILE

Use this command to display the name of each file in each floppy disk directory.

Syntax FLOPPY FILE [*file-id*]

Parameters Table 2-38 shows the syntax parameter of the FLOPPY FILE command.

Table 2-38. **Parameters of the FLOPPY FILE command**

Parameter	Description
<i>file-id</i>	Refers to the name of the file and the directory in which it resides. It has the form <i>file-name.file-type</i> as previously defined. If no file id is specified, the default is to list the names of all the files in each directory. (Masking can be used for this application.)

The total number of files displayed, as well as the total number of disk sectors used by these files, is accumulated and displayed by file type.

The Total of n sectors in m files message is displayed as each parameter of the File command is processed.

Message parameters The parameters for the file number and size message are shown in table 2-39.

Table 2-39. **Parameters of the file number and size messages**

Parameter	Description
<i>n</i>	The sum of the FSIZE field for all the files of a file type requested by a parameter on the FILE command.
<i>m</i>	A decimal number that represents the number of files for the given file type.
The Grand total of <i>n</i> sectors in <i>m</i> files message is displayed when all the parameters on the FILE command are satisfied.	
<i>n</i>	The sum total of the FSIZE field for all the files requested on the parameter string of the FILE command
<i>m</i>	A decimal number that represents the total number of files requested by the parameter string on the FILE command.

If during the search of the system file directories, file catalogs are found to contain no files, this information is saved until all the parameters on the FILE command have been completed and the NO files in type file-types message is displayed.

FLOPPY FCHECK

Use this command to display data related to the fragmentation of files on the floppy disk and to review a list of sectors that were logged as bad by the sector check process during initialization. This information can help you decide whether file maintenance should be performed on the floppy disk (such as saving or restoring files, or reformatting the disk).

Syntax FLOPPY FCHECK

Displayed information The following information is displayed after utilization of the FLOPPY FCHECK command.

- The logical block number (LBN) and the number of sectors available is displayed for each group of contiguous, available sectors.
- The total number of available sectors that occur in groups of 3 or fewer sectors. This is an approximation of the number of sectors on the disk less likely to be used than groupings of more than 3 sectors. These sets of sectors are used for small files (for example small .JSL, .FSL files).
- The total number of available sectors that occur in groups of 5 or fewer sectors (includes the count of groups of 3 sectors or less). This is an approximation of the maximum number of sectors on the disk less likely to be used than groupings of more than 5 sectors. These sets of sectors are used for small files (for example small .JSL, .FSL files).

- The number of available sectors on the floppy disk. This is the total amount of floppy disk space that is available for files.
- The largest available space on the floppy disk. This is the size of the largest possible file that can be stored/saved.
- A list of the sectors that were logged as bad by the sector check process during initialization of the floppy disk.

FLOPPY RESTORE

Use this command to copy files from the floppy disk to a system disk.

Syntax FLOPPY RESTORE [REVIEW] [*disk-id*] {*file-id* | *input-file-id output-file-id* | *file-type* | ALL}

Parameters The syntax parameters are shown in Table 2-40.

Table 2-40. **Parameters of the FLOPPY RESTORE command**

Parameter	Description
REVIEW	Prompts the user when a file from the floppy disk replaces a file of the same name on the system disk. The system displays a message requesting approval to replace the file. If the response is Y, the file on the system disk is replaced by the one from the floppy disk. If the response is N, the specified process skips to the next file on the floppy disk.
<i>disk-id</i>	Identifies to which system disk (DP0:, DP1:, DP2:, or DP3:) the file is restored.
<i>file-id</i> <i>input-file-id</i> <i>output-file-id</i>	Refers to the name of a file to be restored and the file directory in which it resides. The form used is <i>file-name.file-type</i> , for example UNIVAC.JSL. (Masking can be used if different file IDs are not specified for input and output.)
<i>file-type</i>	Any one of the system directories to be restored.
ALL	Restores all files from the floppy disk to the system disk.

FLOPPY LIST

Use this command to print a list of the file name in each directory. Other file-related information is also provided.

Syntax FLOPPY LIST [*file-id*]

Parameters The syntax parameters are shown in table 2-41.

Table 2-41. **Parameter of the FLOPPY LIST command**

Parameter	Description
<i>file-id</i>	Refers to the name of the file and the directory in which it resides. It has the form <i>file-name.file-type</i> as previously defined. If no <i>file-id</i> is specified, the default is to list the names of all the files in each directory. (Masking can be used for this application.)

The total number of files listed as well as the total number of disk sectors used by these files is accumulated and listed by file type.

The message `Total of n sectors in m files` appears as each parameter on the file command is processed.

Message parameters

Table 2-42 shows the parameters of the file number and size messages.

Table 2-42. **Parameter of the file number and size messages**

Parameter	Description
<i>n</i>	The sum of the FSIZE field for all the files of a file type requested by a parameter on the FILE command.
<i>m</i>	A decimal number that represents the number of files for the given file type.
The Grand total of <i>n</i> sectors in <i>m</i> files message appears when all the parameters on the FILE command are satisfied.	
<i>n</i>	The sum total of the FSIZE field for all the files requested by the parameter string of the FILE command.
<i>m</i>	A decimal number that represents the total number of files requested by the parameter string of the FILE command.

If, during the search of the system file directories, file catalogs are found to contain no files, this information is saved until all the parameters on the FILE command have been completed. The `No files in type file-types` message prints on the output page.

Copying the system software tape

The following describes procedures for making a backup of the system software tape.

SST (system software tape or cartridge)

Use this command to create backup Xerox labeled tapes (SSTs) or Xerox labeled cartridges (SSCs).

Logon levels Level 2 or higher.

Restriction You can copy 1 to 1350 files with SST.

Procedure description At the OS1000 READY FOR COMMANDS prompt, enter:

```
RCU SST <CR>
SST <CR>
```

This starts the task. Entering SST will cause the PC UI to go to the Terminal Emulation Mode (TEM). The following message is then displayed:

```
This task duplicates Xerox labeled tapes or SSTs.
The duplicate SSTs can be either offline only,
online only, or offline and online. This task also
allows you to modify the duplicate SSTs by adding
files, deleting files, and changing the order of
the files.
```

The system then prompts with the message:

```
Would you like to continue (Y/N)? <CR>
```

If you enter N, the system will return to the windows mode. If you enter Y, the following menu displays:

```
The options are:
1) Read files from a Xerox labeled TAPE or SST to disk
2) Modify the order of the files with SSTMOD.TMP
3) Write an offline only SST
4) Write an offline and online SST
5) Write an online only SST
6) Write the TAPE files copied in option 1 to another tape
7) Delete files copied from tape to disk in option 1
8) Select Device (default device is TD0)

Use this command to create backup Xerox labeled tapes (SSTs)
or Xerox labeled cartridges (SSCs).

9) Exit
Enter the number of the option you want: 1 <CR>
```

Use **option 1** to copy the files from an LPS labeled tape or SST to the system disk. You are prompted "Is this an SST Y/N?" then to load the tape or SST and then prompted to specify if the tape is an SST or not.

```

Is this an SST (Y/N)? <CR>
Please load the SST or tape you wish to copy, then
press <<ENTER>> to continue or 'X' to return to the
menu: <CR>

Reading tape...
Reading file <filename>
    
```

While the task is reading in the files from the tape, files on the disk are being created. Each new file is given the name ST****.MSC, where **** is a unique four-digit number from 0 to 9999. The Xerox labeled tape is read until the end of volume is reached, and the SST is read until ENDFIL.END is reached. When SST finishes reading the tape, SST unloads the tape.

When this completes, the previous menu again displays.

At this point, you may select **option 2**. The second option allows you to change the list of files just read in from the tape in option 1. To modify the list of files, you must have created a file called SSTMOD.TMP, which contains modifying operators. Create this file using the editor. If this file has not yet been created, you can exit SST using **option 9**, create the file, and rerun SST with **option 2**. (Note: After each SST exit, necessary information regarding the tape that was read in through option 1 is saved.) The three modifying operators are ADD, DELETE, and MOVE:

- **ADD** Adds a file which resides on the disk to the list.
- **DELETE** Deletes the file from the list.
- **MOVE** Moves a file from one position on the list to another.

Add syntax ADD /BF:n<file-name₁> <file-name₂>

Parameters The parameters of the add syntax are shown in table 2-43.

Table 2-43. **Parameters of the add syntax command**

Parameter	Description
/BF:n	Assigns a blocking factor from 1 to 16 to the added file.
<file-name ₁ >	Specifies the file in the list after which <file-name ₂ > is to be added. <file-name ₁ > may be the word BEGIN which adds <file-name ₂ > to the beginning of a list.
<file-name ₂ >	The file residing on the disk that is to be added to the list.

Add restriction The ADD command does not include a date in the tape header.

Delete syntax DELETE <file-name>

Delete parameters The parameter of the delete syntax is shown in table 2-44.

Table 2-44. **Parameter of the delete syntax command**

Parameter	Description
<file-name>	The name of the file in the list to be deleted.

Move syntax MOVE <file-name₁> <file-name₂>

Move parameters The parameters for the move syntax are shown in table 2-45.

Table 2-45. **Parameters of the move syntax command**

Parameter	Description
<file-name ₁ >	May be the word BEGIN, which moves <file-name ₂ > to the beginning of the list.
<file-name ₂ >	In the list of files is to moved to the position after <file-name ₁ >.

If a modifying operator fails, error messages are displayed and you are asked to press the <ENTER> key to continue.

The modified file list is stored in SSTFIL.TMP. Review this file by exiting SST and entering EDIT or TYPE commands.

After option 2 completes, the menu shown above displays again.

Use **option 8** to specify the device type. The default is TD0. If you enter 8, the system displays:

```
1) TD0
2) TD1
3) TD2
Enter the number of the option you want: 1 <CR>
```

If you enter 1, the operations following Select Device read or write to the tape loaded on TD0. You must reset the device configuration if you need option 2, TD1, or option 3, TD2. When option 8 completes, the menu option is displayed again.

Options 3 to 6 require you to load a tape. (When you begin SST, run TAPE VOLINIT on the target tape to make it the same density as the source tape.)

- Use **option 3** to create an offline only SST.
- Use **option 4** to create an offline and an online SST.
- Use **option 5** to create an online only SST.



Note: The online portion of a duplicate SST tape created by SST.TSK contains more .SAF files than the master SST tape.

- Use **option 6** to create a duplicate Xerox labeled tape. SST unloads the tape after completing each of these options. The following messages appear when creating the specified duplicate tape:

```
Please load a tape (with a write ring), then press
<ENTER> to continue or 'X' to return to the menu:
<CR>>

Writing to tape...
Writing file <filename>
```

Option 7 allows you to delete all the files read in during option 1. Run option 7 before selecting option 1 again. The `Deleting files...` `Deleting the temporary file for <file-name>.` message may be displayed.

Option 9 allows you to exit the task. All necessary information regarding the list of files which were read in by option 1 will be saved so that you may resume where you left off if you run SST again. The PC UI will return to the windows mode.

3. Tracking system activity

The following section describes status file data.

Accounting data in the status file

The following section details when and how the system adds data to the status file.

How the system adds data to the status file

The system automatically updates the status file:

- When input processing starts a report, an empty entry in the status file is created.
- When the end of the report is reached.

As the LPS processes print jobs, report information is automatically accumulated and saved into the user's status file on a report basis. Report information is stored individually, not cumulatively, when the input and output tasks complete processing of a report.

The JOBS command provides a display of the status file. The "Saving report accounting information" section provides procedures for transmitting the status file to other media for further processing. The status file data selected for storage gives the user an extended accounting capability and allows tracking of daily system activity on the LPS. This information is used as input to host-based accounting systems for report accounting.

The SFS command reinitializes the status file whenever the LPS is idle.

The following sections explain how status information may be saved, interpreted, and initialized.

Status file creation during sysgen

If a status file does not exist at system generation time, the system displays the message DO YOU WANT REPORT ACCOUNTING? (Y OR N). Selecting Y entry creates a status file large enough to hold 2,000 reports. If a job contains over 2,001 reports, you should clear the status file before running the job. An N entry creates a status file large enough to hold 102 reports.

Initially, all entries in the new status file are cleared. If the response to the report accounting question is Y, the last 25 entries in the 2,001 entry status file are marked to warn the operator that accounting data loss is imminent and that it is time to execute the file-save procedure. The system command SFS also allows the operator to clear and reinitialize the status file.

Backing up status file information

As the status approaches the file-full state, a message warning of this condition is displayed:

```
STATUS FILE NEARLY FULL  
EXECUTE STATUS FILE SAVE PROCEDURE
```

If you ignore the file-full warnings, and the number of reports in the system exceeds the status file size selected, the status of the oldest report is lost. Report processing information is then overwritten by newer report data with no further warnings until you reinitialize the status file for the SFS command.



Note: The status file includes print jobs started by nonprint tasks (for example FDL, PDL, EDIT, and so on).

To facilitate use of the report accounting feature, the message

```
APPROXIMATELY XXX UNUSED ENTRIES IN THE ACCOUNTING  
FILE
```

is displayed in response to the JOBS command. The number of unused report entries in the status file is displayed so you can save the status file before starting a job which may exceed the number of report entries remaining in the status file.

After the status file has used the memory allocated to it, it writes each new record over the oldest existing record. Unused entries are existing entries which have not yet been overwritten. The total unused entries value is determined the first time the status file has no free space left; this value is used by the status file to track the process.

The SFS command can be used to write the status file to an LPS-labeled tape or to a disk file for transfer to a remote host. To initiate transfer of the status file, the host must request the status file be sent with the remote communication interface. After a copy of the status file has been saved, enter the SFS command to reinitialize the file. The format of the status file which has been saved is provided in the following section.

As print jobs are processed by the LPS, usage data is accumulated and saved by the system software.



Note: Accounting information appearing on the accounting page (if it is printed) may not reflect abnormal circumstances occurring during the last 16 pages of a report. The accounting page is formatted for printing before the last data page is printed, and it is assumed that the last few pages will be printed successfully. Information which may be affected includes: pages to bin; pages to tray; pages with graphics; simplex pages printed; duplex pages printed; number of lines printed; output processing time; and primary and secondary report completion codes.

An accounting page is produced automatically whenever the primary and secondary report completion code is non-zero. If the condition occurs within the eight duplex back sides (simplex is not a problem), the accounting sheet may not be printed. If printed, it will not reflect the correct report completion code (RCC) nor will the next report's accounting sheet show this RCC. However, the system status file will always have the correct RCC.

Status file data in the online accounting page

Figure 3-1 illustrates a sample online accounting page.

Figure 3-1. Sample online accounting page

```

XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX

DATE:          14 JAN 98   AT   19:35:32
DEPARTMENT:    M0036:JDL*
JOB ID:        248        REPORT NO. 1
FILE ID:
Input Processing Time:  00:00:07
Output Processing Time: 00:00:13
REPORT COMPLETION CODES: 0      0
Pages to Bin:         1
Pages to Tray:        1
Paper Path Holes:     0          Graphic Pages Printed: 0
Lines Printed:        66          Graphic Exception Code: 0
Online Idle:          0          Tape Mounts:0
Blocks Read:          0
Blocks Skipped:       0
RECORDS READ:         79
DJDE RECORDS READ:   1
Maximum Copy Count:  1
OVERPRINTS:          0
COLLATE:              YES
SF/MF:                MULTI
SIMPLEX/DUPLEX:       SIMPLEX
JDE,JDL USED:         1,M0036
ACCTINFO:
INITIAL FONT LIST:    L00TPB

INITIAL FORM LIST:    HCF129

INITIAL CME LIST:     -NONE

XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX

```

Status file data

The following section describes the parameters listed in a status file data report.

DATE entry

The current date as entered when the system was booted from disk. The time (*hh:mm:ss*) indicates when the input processor began processing the input for a report.

DEPARTMENT entry

This is either the character string entered as part of the ACCT statement (DEPT=) or a system default name. The system default name will be the name of the .JDL file under which the job ran with :JDL appended to it. If the requested department name using DEPT= has not been previously set up in the accounting file, the requested name will be listed on the accounting page and an asterisk will be appended.

JOB ID entry

This entry is the system-generated identifier of the print job. It is incremented automatically by the system each time a job is run.

REPORT NO. entry

The number of the report within the current job.

FILE ID entry

This is the file identifier field of the HDR1 label if it exists. (Refer to the *LPS Tape Formats Manual*.)

INPUT PROCESSING TIME entry

Elapsed time (*hh:mm:ss*) for the input processor to process the report from the input source device and pass the results to the output processor. The functions of the input processor are discussed in the "LPS operating system software" section of the *Xerox DocuPrint 180 LPS Product Reference*. For online systems, this includes any time the input processor is waiting for data because of host-induced data transmittal delays.

OUTPUT PROCESSING TIME entry

Elapsed time (*hh:mm:ss*) for the output processor to print the report as sent to it by the input processor. This time includes processing time of the report plus any time the output processor is waiting for paper jams to be cleared, and so on. The functions of the output processor are discussed in the "LPS operating system software" section of the *Xerox DocuPrint 180 LPS Product Reference*.

Report completion codes

Two codes are listed. The first code is the primary report completion code; the second code is the secondary report completion code. These codes indicate the completion status of the report. Refer to tables 3-1 and 3-2.

Report completion codes refer to abnormal events that occur while the LPS prints a report. If an abnormal event occurs during input processing but the report prints without aborting, the report completion code is zero (0).

The system may combine codes and add them together into a unique code that you must decode. For example, a code of 11 indicates that the system combined exception codes 1, 2, and 8.

These codes also appear in the JOBS command display. This is helpful for reports aborted by the user, for which accounting pages are not printed.

Table 3-1 displays the meanings of primary report completion codes.

Table 3-1. **Primary report completion codes**

Code	Meaning
0	Normal completion
1	Operator-issued SPACE PAGES command
2	End-of-report encountered during SPACE PAGES command processing
4	Data on page exceeded page size
8	Graphic memory size exceeded
16	Font memory size exceeded
32	Report too big for available memory
64	Too much data on a page of duplex set-up error
128	Irrecoverable graphics imaging error
256	Irrecoverable disk or imaging hardware error
512	Account page not printed because of aborted job
1024	Unable to recover to a page boundary (following a system crash)
2048	Successive page recovery used
4096	Secondary report completion code used
8192	Operator-issued ABORT O command
16384	Operator-issued ABORT JOB # command

Table 3-2 displays the meanings of secondary report completion codes.

Table 3-2. **Secondary report completion codes**

Code	Meaning
1	Data stream stock callout was overridden by operator.
2	SIZING option caused report to abort.
8	More than 16 TL/DL blocks were used on one page.
16	CODE = NONE error was detected during printing.
32	Set integrity problem.
64	Graphic off leading edge of page.

PAGES TO BIN entry

A count of all pages printed and delivered to the output bins. Only pages delivered are counted; discarded pages are not counted. Any sheet of paper going through the duplex paper path counts as two pages to bin even if only one side has printing (this includes OPRINFO, RPAGE, ROFFSET, DJDE, and RTEXT routing pages). Maximum page count is 999,999,999.

PAGES TO TRAY entry

A count of pages delivered to the sample tray. This includes the printing of the tape label (if VOLUME PLABEL=YES), as well as Sample, Accounting, and RSTACK delimiter pages delivered to the sample tray.

PAPER PATH HOLES entry

A count of paper path holes caused by user job characteristics. Generally, holes are created when the resources of a page cannot be loaded from the disk before the sheet must be fed by the printer. Holes continue to be inserted until the resources for the page can be loaded.

If a page is very complex and "reduced performance mode" must be invoked by the printer in order to print the page, the inserted holes are also included in the count.

Holes caused by error recovery (imaging errors or jams, for example) are not included in the hole count printed on the accounting page. Holes inserted during adjustment of print quality, and holes inserted for paper scheduling reasons are also omitted from the paper path hole count.

GRAPHIC PAGES PRINTED entry

A count of all pages containing at least one graphic impression delivered to the bin.

LINES PRINTED entry

A count of the lines printed on pages delivered to the print trays (PAGES TO BIN) and to the sample tray (PAGES TO TRAY). This includes the number of lines on the accounting, RTEXT, DJDE, and OPRINFO pages. The count does not include the lines printed for a user-requested page number. Maximum line count is 999,999,999.

GRAPHIC EXCEPTION CODE entry

Composite code identifying the types of abnormal graphic processing conditions encountered by the input processing task.

ONLINE IDLE (SEC) entry

A measure of LPS idle time while waiting for the host system. Refer to the "Host-induced idle time" section in the "LPS print processing" chapter for further details on this online accounting page entry.

TAPE MOUNTS entry

A count of the tape volumes required to process an offline job.

BLOCKS READ entry

Number of data blocks read from the input source device.

BLOCKS SKIPPED entry

Number of blocks skipped as a result of the operator-initiated command MOVE *n* BLOCKS (refer to the "MOVE" section in the "Running print jobs" chapter).

RECORDS READ entry

Number of records read from input source device according to format defined in JDE.

DJDE RECORDS READ entry

Number of DJDE records read from input source device.

MAXIMUM COPY COUNT entry

The maximum number of printed copies requested of the report (may not be the number actually printed if, for example, the job was aborted). The number of copies to be printed is defined by the user in the JDE (COPIES command of OUTPUT statement) and may be overridden by DJDEs or by the COPIES option on the START command.

OVERPRINTS entry

The number of overprint lines in the report. If the IGNORE command of the LINE statement (OVERPRINT command) is coded in the .JDL file, the accounting entry is 0.

COLLATE entry

Entry is YES or NO depending on COLLATE command of OUTPUT statement. The default is to collate output pages. COLLATE may also be modified by a DJDE. COLLATE = YES is forced for any job run in duplex mode.

SF/MF entry

Entry is SINGLE (single report mode) or MULTI (multiple report mode). The default mode is MULTI, which may be overridden by the START command.

SIMPLEX/DUPLEX entry

Simplex or duplex mode. Simplex mode (print on one side of page) is the default. Duplex mode (print on both sides) is selected by the PDL DUPLEX command of the OUTPUT statement.

JDE, JDL USED entry

Names of the job descriptor entry and job descriptor library as specified in the START command. These names may have been modified by a DJDE.

ACCTINFO entry

Text from the delimiter record when the ACCTINFO command of the RSTACK statement is coded in the JDE. The document name is the source of the ACCTINFO field on the accounting page and the Report Name field in the job status display. These fields are printed without filtering space characters or punctuation. If the document name is all blank characters, '.DFLT.' is substituted and shown in the Report Name field on the accounting sheet, as well as on the job status display. If the source of the document name is printing protocol, the ACCTINFO field is truncated to 14 characters on the accounting page or job status display. If the document name comes from Printing Instructions, the ACCTINFO field displays 64 characters on the accounting page and 16 characters on the job status display.

INITIAL FONT LIST entry

Names of the fonts referenced in the JDE. Fonts are defined in the PDE statement (FONTS command) and referenced for use in a JDE on the OUTPUT statement (FORMAT command). If the font list is modified by DJDEs, the message DJDE MODIFIED is printed along with the original list of font names.

INITIAL FORM LIST entry

Names of the forms specified in the JDE on the FORMS command of the OUTPUT statement. The form specified in the BFORM command will not be included in this list. If other FORMS are invoked with DJDEs, the message DJDE MODIFIED is printed along with original list of form names.

INITIAL CME LIST entry

The names printed are the identifiers of CMEs referenced in a JDE. CMEs are defined with the CME statement (along with an identifier) and referenced in a JDE with the OUTPUT statement (MODIFY command). If CMEs are used in DJDE processing, the message DJDE MODIFIED is printed along with the original list of names.

Using accounting and system activity commands

The OSS ACCOUNT command allows you to list, add, and delete user accounts, or to back up and clear user account information.

You may use three OSS REPORT commands to report billing, user, and system activity data.

You must use both the PDL ACCOUNT statement and the OSS ACCOUNT command to update user accounts. (Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for information on the PDL ACCOUNT statement.) Use the ACCOUNT command to create or modify the list of user names (usually department) for which accounting information is maintained.

Generating system activity reports

The following section details how to create system activity reports.

REPORT

Use this command to generate the accounting report needed.

- REPORT USER provides user and system activity information for the specified user account.
- REPORT ACTIVITY provides system activity information for all user accounts.
- REPORT BILLING provides page count information on:
 - total print job output
 - total output produced by the diagnostic mode.

The customer billing, user accounting, and system activity reports print in the system portrait font P0612\$.FNT by default. You can specify another font by creating a DFLT.SYS file which consists of any or all of these records:

- BLRPTF = *fontid* (specifies customer billing font)
- ACRPTF = *fontid* (specifies activity report font)
- SARPTF = *fontid* (specifies system activity font).

Billing page counts

The customer billing procedure for the LPS uses the REPORT command. Instead of counting pages using physical meters, software in the printer maintains page counts in its nonvolatile memory. The customer billing procedure invoked with REPORT reads the meter counts stored in memory and produces either a displayed or hardcopy billing summary.

Billing syntax REPORT BILLING [,DISPLAY | ,PRINT]

Parameters Table 3-3 shows the parameters of the REPORT command.

Table 3-3. **Parameters of the REPORT command**

Parameter	Description
DISPLAY	Displays the billing information on the screen. DISPLAY is the default.
PRINT	Prints the billing report

The system controller provides the following information:

- Date
- Time
- Serial number
- System ID
- Base system configuration
- Machine type
 - CPU memory
 - AIS version number
 - bitmap memory size
 - ink memory size
 - text and metrics memory size
 - SCSI disk units present on system
 - page buffer memory size
 - run memory test at boot flag
 - boot automatically flag
 - display boot message flag
 - printer speed (pages printed per minute)
 - printer resolution
 - 1 to n or n to 1 default order
 - paper size
 - tape device assignment
 - online

- System configuration options
 - SCSI (tape) devices present on system
 - system controller language (for example, US ENGLISH)
 - terminal type
 - graphics type
 - shared disk installed
 - DocuPrintServer (formerly XPSM) installed
 - Communication interfaces
- Number of good impressions successfully delivered
- Number of sheets printed in IOT (printer) diagnostic mode
- Number of purge sheets sent to purge tray
- Number of good impressions printed in 3 pitch mode
- Number of good impressions printed in 4 pitch mode
- Number of good impressions printed in 5 pitch mode
- Number of good impressions printed in 6 pitch mode
- Number of good impressions printed in 7 & 8 pitch modes
- Number of good impressions successfully delivered
- Percentage of duplex prints.

Usage statistics Use this command to print the user accounting report.

User syntax REPORT USER[,CLEAR]

When you specify CLEAR, the summary report is printed and the user ID statistics in the accounting log entries are set to zero. The cleared text is printed on the summary report page. The printer notifies the system controller of successful completion.



Note: Customer usage statistics cannot be displayed on the PC UI or on the terminal.

The summary information printed as a result of the REPORT USER command contains accumulated usage data (pages, lines, and so on) for all print jobs (since the last CLEAR) by department or JDL. Refer to figure 3-2 for a sample of this output.

This data may also be written to tape with the ACCOUNT command. The *font-id* specified for ACRPTF determines the font used for the user accounting report. The *font-id* specified for SARPTF determines the font used for the system activity report. The selected font must be portrait and no larger than 9 points.

Figure 3-2. User accounting report

Date 01/10/98	USER ACCOUNTING REPORT	Page 1
Department/JDL	Printing Summary	Time Summary-HR, MIN-
	Pages = 21	Input = 0 ,1
NODEPT	Lines = 2200	Output = 0 ,2
	Pages to Tray = 0	
	Processing Summary	Tape Summary
	Jobs = 2	Mounts = 0
	Files = 2	Blocks Read = 220
	Reports = 2	Blocks Skipped = 0
Department/JDL	Printing Summary	Time Summary,HR, MIN
	Pages = 105	Input = 0 ,5
OPERATION	Lines = 1700	Output = 0 ,7
	Pages to Tray = 0	
	Processing Summary	Tape Summary
	Jobs = 1	Mounts = 1
	Files = 7	Blocks Read = 1100
	Reports = 7	Blocks Skipped = 0
Department/JDL	Printing Summary	Time Summary,HR, MIN
	Pages = 3	Input = ,0
TASK\$:JDL	Lines = 100	Output = ,0
	Pages to Tray = 0	
	Processing Summary	Tape Summary
	Jobs = 3	Mounts = 0
	Files = 3	Blocks Read = 0
	Reports = 3	Blocks Skipped = 0

System activity statistics Use this command to print or display a system activity and LPS usage report.

Activity print syntax REPORT ACTIVITY [,CLEAR]

If CLEAR is specified, system activity information is printed from the activity log and the entries are set to zero. The cleared text is printed on the system activity page. The summary information printed as a result of the REPORT ACTIVITY command contains accumulated usage data for all print jobs (since the last CLEAR). An example of the output is shown in figure 3-3.

Activity display syntax REPORT ACTIVITY DISPLAY

Figure 3-3 shows a sample system activity report.

Figure 3-3. **System activity report**

	SYSTEM ACTIVITY REPORT	
Date 01/15/98	Page Counts	
Pages = 1065	Paper Path Holes = 2	
	Processing Time	
Input:	Hours = 0	Minutes = 6
Output:	Hours = 0	Minutes = 8
	Processing Counts	
Jobs = 49	Files = 47	Reports = 49
	I/O Statistics	
Tape Mounts = 9	Blocks Read = 2065	Blocks Skipped = 0

Establishing accounts and copying account data to tape

The following section describes how to manage user accounts.

ACCOUNT

Use this command to list, add, and delete user accounts, or to back up and clear user account information.

You must use both the PDL ACCOUNT statement and the OSS ACCOUNT command in order to update user accounts. (Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for information on the PDL ACCOUNT statement.) Use the ACCOUNT command to create or modify the list of user (usually department) names for which accounting information is maintained.

Syntax ACCOUNT {{ADD | DELETE},*department*}

OR

ACCOUNT LIST

OR

ACCOUNT WRITE[,TD*n*][,EBCDIC | ,ASCII][,1600 | ,6250][,CLEAR]

Parameters Table 3-4 shows the parameters of the ACCOUNT command.

Table 3-4. **Parameters of the ACCOUNT command**

Parameter	Description
ADD	Adds a department name to the list of departments under which accounting information is maintained.
DELETE	Deletes a department name from the list of department names. A department name cannot be deleted if data for that department exists in the file.
<i>department</i>	1 to 31 alphanumeric characters (A through Z, 0 through 9, and the character ":"), which specify the department as specified in the DEPT command of the PDL ACCT statement.
LIST	Adds a department name to the list of departments under which accounting information is maintained.
WRITE	Specifies how user and system activity data is written to tape. The data written to tape is the data printed by the REPORT command. The data is written to tape starting at BOT and followed by two EOFs.
TD n	Specifies the tape device the file is written to. If omitted, the default is TD0:
<u>EBCDIC</u> ASCII	Specifies the character mode in which the tape is written. Must be EBCDIC or ASCII. The default is EBCDIC.
<u>1600</u> 6250	Specifies the tape recording density in bits per inch. Must be 1600 or 6250. The default is 1600 bpi.
CLEAR	Erases the accounting data after copying to tape.

ACCOUNT commands can also be invoked from an editor command file (refer to the "Editing font files" chapter).

The system provides two department defaults: NODEPT for print jobs without name entries in the accounting file and TASKS\$.JDL for print jobs run by system tasks such as FDL, PDL, and the editor.

Tape format of data written by ACCOUNT

Use this command to write system usage accounting data to tape. The tape is unlabeled with 80-byte records. Each 80-byte record is identified in its first two bytes by a value indicating the type of information contained in that record. These 2-byte identifiers are 10, 11, 12, 13, 16, 17, and 18. Records with identifier 14 (for future use) are written to tape but contain no currently useful information. The first four records written to tape are record types 10, 11, 12, and 13 which are defined in Tables 3-5 through 3-8.

Table 3-5 outlines the content structure of ACCOUNT record 1.

Table 3-5. Content structure of ACCOUNT record 1

Bytes	Contents
0-1	Identifier (10)
2-10	Date in format DD-MM-YY
13-20	Time in format HH:MM:SS
21-79	Unused

Table 3-6 outlines the content structure of ACCOUNT record 2.

Table 3-6. Content structure of ACCOUNT record 2

Bytes	Contents
0-1	Identifier (11)
2-39	Unused
40-49	Pages
50-59	Reports
60-69	Files
70-79	Jobs

Table 3-7 outlines the content structure of ACCOUNT record 3.

Table 3-7. **Content structure of ACCOUNT record 3**

Bytes	Contents
0-1	Identifier (12)
2-6	Hours (input)
7-11	Minutes (input)
12-21	Unused
22-26	Hours (output)
27-31	Minutes (output)
32-43	Unused
44-49	Mounts
50-59	Blocks read
60-69	Blocks skipped
70-79	Paper path holes

Table 3-8 outlines the content structure of ACCOUNT record 4.

Table 3-8. **Content structure of ACCOUNT record 4**

Bytes	Contents
0-1	Identifier (13)
2-26	Unused
27-31	Number of blocks with CRC error
32-41	Blocks received
42-79	Unused

The information contained in these records is printed by the REPORT ACTIVITY command.

The remaining useful records on the tape are sets of type 16, type 17, and type 18 records, one set for each department-name/JDL entry in the accounting log. Information contained in the records is the same as that printed by the REPORT USER command. The format of the records is shown in tables 3-9, 3-10, and 3-11.

Table 3-9 outlines the content structure of ACCOUNT record 5.

Table 3-9. **Content structure of ACCOUNT record 5**

Bytes	Contents
0-1	Identifier (16)
2-32	Department name
33-34	Unused
35-39	Pages to tray
40-79	Same as record 2

Table 3-10 outlines the content structure of ACCOUNT record 6.

Table 3-10. **Content structure of ACCOUNT record 6**

Bytes	Contents
0-1	Identifier (17)
2-69	Same as record 3
70-79	Lines

Table 3-11 outlines the content structure of ACCOUNT record 7.

Table 3-11. **Content structure of ACCOUNT record 7**

Bytes	Contents
0-1	Identifier (18)
2-79	Same as record 4

Modifying the status file

The following section describes how to modify the status file.

SFS (status file services)

The status file services is used by the LPS analyst and by LPS service personnel.

Use this command to create, clear, display, print, and store status file information. The SFS command can be keyed in or placed in a command file.

Syntax SFS {CLEAR | CREATE | DISPLAY | *file-name* | HARDCOPY | HOST | TAPE | TD*n* | ? | !} [,*parameters...*]

Each primary option (TAPE, HOST, and so on) has an associated set of parameters. The function of each option and its parameters is described in the corresponding section below.

Keywords may be specified using the first three letters. Parameter specification is positional, so an unspecified (default) parameter must be represented by a comma if a following parameter is specified. (A command line cannot end with a comma.) For example, the command line:

```
SFS TAP,,,1600
```

is equivalent to:

```
SFS TAPE,ASCII,256,1600
```

SFS CLEAR

Use this command to clear the status file and reset warning and frequency flags by reinitializing the status file. *Warning flags* indicate that the status file is about to be overwritten. *Frequency flags* determine how often warning messages are displayed.

Syntax SFS CLEAR[,NOF|,OWF | ,DWF | ,NWF:n.m]

Parameters Table 3-12 shows the parameters of the SFS CLEAR command.

Table 3-12. **Parameters of the SFS CLEAR command**

Parameter	Description
NOF	Specifies no warning and frequency flags. No warnings are to be displayed.
OWF	Specifies old warning and frequency flags. Previous flag settings are to be preserved.
DWF	Specifies default warning and frequency flags. The default value for warning flags is 40. The default value for frequency flags is 3.
NWF:n.m	Specifies new warning and frequency flags. <i>n</i> indicates the threshold at which warnings begin. <i>m</i> indicates the frequency at which warning messages are displayed. (For example, if <i>n</i> =25 and <i>m</i> =3, warnings are displayed when 25, 22, 19, 16, and so on, entries can be made before the end of the status file is reached.)

Restriction You can clear the status file only when the system is idle. Before clearing the status file the current status file should be backed up to tape since the system does not warn that the status file has been backed up.

SFS CREATE

Use this command to create a new status file and set its size and its warning and frequency flags, initializing the status file. (Warning flags indicate that the beginning of the file is about to be overwritten. Frequency flags determine how often warning messages are displayed.)

Syntax SFS CREATE[, [OS|DS | NS:*n*][, NOF | ,OWF|,DWF | ,NWF:*n.m*]]

Parameters Table 3-13 shows the parameters of the SFS CREATE command.

Table 3-13. **Parameters of the SFS CREATE command**

Parameter	Description
OS	Specifies the old file size.
DS	Specifies the default file size (2000 entries).
NS: <i>n</i>	Creates a new file of <i>n</i> blocks where $3 \times n$ = the number of possible entries in the status file.
NOF	Specifies no warning and frequency flags. No warnings are displayed.
OWF, DWF, and NWF: <i>n.m</i>	See SFS CLEAR .

SFS DISPLAY

Use this command to display print queue database entries. You receive a prompt that allows you to continue or stop the display

Syntax SFS DISPLAY[, [SHORT | COMPLETE][, num]]

Parameters Table 3-14 shows the parameters of the SFS DISPLAY command.

Table 3-14. **Parameters of the SFS DISPLAY command**

Parameter	Description
SHORT	Displays the status file summary using the short format which lists processed jobs and contains the BLOCK NO., JOB NO., ENTRY NO., ENTRY STATUS fields.
COMPLETE	Displays the status file summary using the long format which displays one entry at a time.
num	<p>Specifies the number of entries to be displayed as follows:</p> <p>n (the first n entries)</p> <p>$-m$ (the last m entries)</p> <p>$n-m$ (all entries between n and m, inclusive; if $n = m$, only one entry is displayed.)</p> <p>$n_1;n_2; \dots;n_{10}$ (from one to ten discrete entries, where n is the nth file entry, or a range of entries in the form $n-m$. Entries must be specified in ascending order.)</p> <p>The default displays all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file.</p>

The Figure 3-4 shows the Status File report (Short).

Figure 3-4. **Status file report (short)**

```
DISK ID:
DATE & TIME:

STATUS SUMMARY                                ENTRY NO.: 9999

                                GENERAL INFORMATION

Report Number/Suffix: 99999/x      Date: 99-xxx-99Time: 99:99:99
Job Number: 99999                  Status:
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                INPUT PROCESSING INFORMATION

Customer Job Number: xxxxxxxx      Customer Report Name: xxxxxxxx
JDE: xxxxxxJDL: xxxxxx            Pages Written to Disk: 99999999
Maximum Copy Count: 99999         Department Name: xxxxxxxxxxxxxx
Tape Mounts: 99999               Graphic Images Read: 99999999
Completion Code: xxxx
(Abort Message: xxxx)
Input Processing Time: 99:99:99
Flag: xxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxxxxxxxxx

                                OUTPUT PROCESSING INFORMATION

Copies Printed: 99999
Pages Printed: 999999999
Pages to Bin: 999999999
Pages to Tray: 99999
Graphic Pages Printed: 999999999
Primary Report Comp Code: 999999999
Secondary Rept Comp Code: 999999999
Lines Printed: 999999999
Simplex Pages Printed: 999999999
Duplex Pages Printed: 999999999/999999999
Output Processing Time: 99:99:99
```

SFS file-name

Use this command to print a status report in a user-defined format and deliver it to the selected print tray.

Syntax SFS *file-name* [, [SIMPLEX | DUPLEX | TEST]] [, [c] [, num]]

Parameters Table 3-15 shows the parameters of the SFS file-name command.

Table 3-15. **Parameters of the SFS file-name command**

Parameter	Description
<i>file-name</i>	Specifies a user-created form file on disk, which has the format and content of a status report. The file name may be 1 to 6 characters (A through Z, 0 through 9). The file extension is .MSC and should not be specified.
SIMPLEX	Prints the status report on one side of the paper.
DUPLEX	Prints the status report on both sides of the paper.
TEST	Prints only the report and column header information to the sample tray.
<i>c</i>	Specifies the number of copies. The default is 1.
<i>num</i>	<p>Specifies the number of entries to be displayed as follows:</p> <p><i>n</i> (the first <i>n</i> entries)</p> <p><i>-m</i> (the last <i>m</i> entries)</p> <p><i>n-m</i> (all entries between <i>n</i> and <i>m</i>, inclusive; if <i>n</i> = <i>m</i>, only one entry is displayed.)</p> <p><i>n₁;n₂; ...n₁₀</i> (from one to ten discrete entries, where <i>n</i> is the <i>n</i>th file entry, or a range of entries in the form <i>n-m</i>. Entries must be specified in ascending order.)</p> <p>The default prints all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file.</p>

Restriction Non-existing entry number [num] in the SFS command will cause the following message to be displayed.

```
SF2115 Invalid values for 'NUMBER OF ENTRIES'
```

SFS HARDCOPY

Use this command to print a status report and deliver it to the selected print tray.

Syntax

SFS HARDCOPY [, [SHORT | COMPLETE] [, [SIMPLEX | DUPLEX] [, c] [, num]]]

Parameters Table 3-16 shows the parameters of the SFS HARDCOPY command.

Table 3-16. **Parameters of the SFS HARDCOPY command**

Parameter	Description
SHORT	Prints the status report using the short format which contains the BLOCK NO., JOB NO., ENTRY NO., ENTRY STATUS fields.
COMPLETE	Prints the status report using the long format.
SIMPLEX	Prints the status report on one side of the paper.
DUPLEX	Prints the status file report on both sides of the paper.
c	Specifies how many copies to be printed. The default value is 1
num	<p>Specifies the number of entries to be displayed as follows:</p> <p>n (the first n entries)</p> <p>$-m$ (the last m entries)</p> <p>$n-m$ (all entries between n and m, inclusive; if $n = m$, only one entry is displayed.)</p> <p>$n_1;n_2; \dots;n_{10}$ (from one to ten discrete entries, where n is the nth file entry, or a range of entries in the form $n-m$. Entries must be specified in ascending order.)</p> <p>The default displays all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file.</p>

Figure 3-5 is a sample of a status file report.

Figure 3-5. **Status file report -- hardcopy form (complete)**

```

DISK ID: [optional]

DATE & TIME: [optional]

                                STATUS SUMMARY                                Page 9999
                                ENTRY No.: 9999

GENERAL INFORMATION

Report Number/Suffix: 99999/x
Job Number: 99999
Date: 99-xxx-99      Time: 99:99:99
Status: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

INPUT PROCESSING INFORMATION

Customer Job Number: xxxxxxxx
Customer Report Name: xxxxxxxxxxxxxxxxxxxxxxxx
JDE: xxxxxxxx
JDL: xxxxxxxx
Maximum Copy Count: 99999
Pages Written to Disk: 999999999
Department Name: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Tape Mounts: 99999
Graphic Images Read: 999999999
Input Processing Time: 99:99:99
Flag: xxxxxxxxxxxx xxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxx
Completion Code: xxxx

OUTPUT PROCESSING INFORMATION

Copies Printed: 99999
Pages Printed: 999999999
Pages to Bin: 999999999
Pages to Tray: 99999
Graphic Pages Printed: 999999999
Primary Report Comp Code: 99999
Secondary Rept Comp Code: 99999
Simplex/Duplex Pages Printed: 999999999/999999999
Lines Printed: 999999999
Output Processing Time: 99:99:99

```

SFS HOST

Use this command to format the status file for transmission to a remote host and write this host-format file to the LPS disk, using the file name SFASEB.MSC. The LPS/host file transfer interface can then be used to transfer the file. (The SFS HOST command does not send the status file.)

Syntax SFS HOST [,*fmt*][,*num*]

Parameters Table 3-17 shows the parameters of the SFS HOST command.

Table 3-17. **Parameters of the SFS HOST command**

Parameter	Description
<i>fmt</i>	Specifies the data format option. Must be EBCDIC or ASCII. The default is ASCII.
<i>num</i>	<p>Specifies the number of entries to be displayed as follows:</p> <p><i>n</i> (the first <i>n</i> entries)</p> <p><i>-m</i> (the last <i>m</i> entries)</p> <p><i>n-m</i> (all entries between <i>n</i> and <i>m</i>, inclusive; if <i>n</i> = <i>m</i>, only one entry is displayed.)</p> <p><i>n₁;n₂ ; ...n₁₀</i> (from one to ten discrete entries, where <i>n</i> is the <i>n</i>th file entry, or a range of entries in the form <i>n-m</i>. Entries must be specified in ascending order.)</p> <p>The default displays all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file.</p>

SFS TAPE

The use of the SFS TAPE command is described below.

Purpose Use this command to save the status file on the tape device assigned to TD0.

Syntax SFS TAPE [, [*fmt*][, [*blk*][, [*den*][, [*num*]]]]]

Parameters Table 3-18 shows the parameters for the SFS TAPE command.

Table 3-18. **Parameters of the SFS TAPE command**

Parameter	Description
<i>fmt</i>	Specifies the data format option. Must be EBCDIC or ASCII. The default is ASCII.
<i>blk</i>	Specifies the block length in bytes. Must be 80, 256, or 512. The default is 256.
<i>den</i>	Specifies the tape density. Must be 1600 or 6250. The default is 1600.
<i>num</i>	<p>Specifies the number of entries to be displayed as follows:</p> <p><i>n</i> (the first <i>n</i> entries)</p> <p><i>-m</i> (the last <i>m</i> entries)</p> <p><i>n-m</i> (all entries between <i>n</i> and <i>m</i>, inclusive; if <i>n</i> = <i>m</i>, only one entry is displayed.)</p> <p><i>n</i>₁;<i>n</i>₂; ...<i>n</i>₁₀ (from one to ten discrete entries, where <i>n</i> is the <i>n</i>th file entry, or a range of entries in the form <i>n-m</i>. Entries must be specified in ascending order.)</p> <p>The default displays all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file.</p>



Note: This tape is intended for processing on the user's host computer and cannot be reread back into the LPS.

SFS TDn

Use this command to save the status file on the identified tape device.

Syntax SFS TDn [, [fmt][, [blk][, [den][, num]]]]

Parameters Table 3-19 shows the parameters of the SFS TDn command.

Table 3-19. **Parameters of the SFS TDn command**

Parameter	Description
TDn	Specifies the physical number of the tape device (TD0, TD1, TD2).
<i>fmt</i>	Specifies the data format option. Must be EBCDIC or ASCII. The default is ASCII.
<i>blk</i>	Specifies the block length in bytes. Must be 80, 256, or 512. The default is 256.
<i>den</i>	Specifies the tape density. Must be 1600 or 6250. The default is 1600.
num	Specifies the number of entries to be displayed as follows: <i>n</i> (the first <i>n</i> entries) <i>-m</i> (the last <i>m</i> entries) <i>n-m</i> (all entries between <i>n</i> and <i>m</i> , inclusive; if <i>n</i> = <i>m</i> , only one entry is displayed.) The default displays all entries. The specified values must be equal to or greater than one, and equal to or less than the maximum number of entries in the status file. Discrete sets of entries may not be specified with the SFS TAPE command.



Note: This tape is intended for processing on the user's host computer and cannot be reread back into the LPS.

SFS ?

Use this command to perform a search for an existing status file. If the status file exists, nothing is done. If no status file exists, a new status file is created and you are prompted to specify warning and frequency flag settings.

Syntax SFS ?

SFS !

Use this command to abbreviate the following command: SFS TAPE,EBCDIC,80,1600

Syntax SFS !

Creating the SFS user form file

An SFS user form file is a user-created file containing commands that describe the content and format of a status report. Its file name is 1 to 6 characters (A through Z, 0 through 9); its file type must be .MSC.

The SFS user form file consists of three statements:

- TITLE statement (optional)
- COLUMN statement
- END statement (optional).

Statements are separated by semicolons (;).

Each statement is made up of one or more commands, described individually below. Commands within a statement are separated by commas (,).

Comments may be inserted anywhere in the file and are delimited by slashes (/).

A form file may be terminated by a comma, a semicolon, an END statement, or the physical end-of-file.

Statement syntax

Keywords (represented in uppercase letters) may be abbreviated to the first three characters. If more than three characters are entered, the entire keyword must be present.

Single quotes are required where shown. There must be at least one blank between the closing single quote of a text string and the opening single quote of the next text string. An apostrophe within a text string is denoted by two consecutive single quotes (' ').

An empty string (KEYWORD = ' ') denotes a blank line.

Title statement

The title statement, if present, must be the first statement in the file. It can consist of three commands:

TITLE = 'text string' ['text string'...]

The entered text strings form the title of the status file report. The default value for TITLE is a blank line. The number of lines in TITLE should not exceed 4. Lines after the fourth one are ignored.

MAX = *integer*

Defines the number of lines in header rows for the status report. The default value is 1. The value for MAX should not exceed 4. If a value greater than 4 is entered, 4 is assumed.

HJUSTIFY = { L | R | C }

Specifies justification (left, right, or center) for the TITLE text. Default for HJUSTIFY is C (center).

Column statement Each column statement consists of one to seven commands:

HEADER = *'text string' ['test string...']*

The entered strings form the header for the column that contains the command. If a column header has more than one line, the MAX command must be present. (If there is no title command, a MAX command may be included in the column command.) The number of lines in HEADER should not exceed the MAX value. If less than the MAX value is coded, blank lines are provided on top. The default for HEADER is the MAX value of blank lines.

WIDTH = *integer*

Defines the width for the column that contains the command. A WIDTH command affects the column that contains it and all following columns, until another WIDTH command is encountered. The first column defined following the title statement should contain a WIDTH command. Columns with an invalid WIDTH are ignored.

CONTENT = *three-character-string*

Specifies the status file content of the column. (A list of status file fields, showing the valid three-character mnemonic, is located in table 3-11.) The CONTENT command is required for each column. Columns with an unspecified content are ignored.

LENGTH = *integer*

Specifies the number of characters in the CONTENT field that are to be shown on the report. The number of characters actually printed is whichever value is smallest: the LENGTH value, WIDTH value, or real length of the status file field. (Actual field lengths are listed in table 3-11.) The default for LENGTH is whichever value is smaller: the WIDTH value or the real length of the status file field.

HJUSTIFY = { L | R | C }

Specifies justification (left, right, or center) for the HEADER text, affecting the column that contains the command and all the following columns, until another HJUSTIFY command is encountered. Default for HJUSTIFY is C (center).

CJUSTIFY = { L | R | C }

Specifies justification (left, right, or center) for the CONTENT, affecting the column that contains the command and all the following columns, until another CJUSTIFY command is introduced. The default for CJUSTIFY is C (center).

PART = { L | R }

Specifies the portion of the status file field that CONTENT is to show (left or right) when the specified LENGTH is less than the real length of the status file field. The default for PART is R (right). PART affects the column that contains the command and all the following columns, until another PART command is encountered.

End statement The end statement consists of only one command:

END

Specifies the logical end of the SFS form file. Whatever follows the END keyword is discarded. The END command may also be used as the last command of the last column command with the same result.

Tape format of data written by SFS

The SFS command writes the status file to the LPS-labeled tape, or to a remote host (with the LPS/Host File Transfer Interface). If written to tape, the first record is the required LPS standard label, followed by 512-byte blocks as defined in Table 3-20.

Table 3-20 lists status file definitions.

Table 3-20. **First status file record**

Bytes	Contents
0-167	Report entry (described below)
168-335	Report entry
336-503	Report entry
504-505	Unused
506-507	File format id (internal use)
508-509	Forward pointer (described below)
510-511	Backward pointer (described below)

Table 3-21 lists the structure of host/tape output and mnemonics.

Table 3-21. **Structure of host/tape output and mnemonics**

Mnemonic	Status field content	Maximum length (in characters)
STA *	Status	1
JNO	Job number	5
RNO	Report number	5
DAT	Date	9
TIM	Time	8
CJN	Customer job number	6
RNA	Report name	16
JDE	Job descriptor entry	6
JDL	Job descriptor library	6
MCC	Maximum copy	5
PWD	Pages written to disk	9
DPT	Department name	31
RNS	Report number suffix	1
NTM	Number of tapes mounted	5
GIR	Graphic images read	9
IPT	Input processing time	8
FLG	Flag	3
MES	Message/completion code	4
COP	Number of copies printed	9
PPC	Pages printed in this copy	9
PTB	Pages to bin	9
PTT	Pages to tray	9
GPP	Graphic pages printed	9
CC1	Primary completion code	9
CC2	Secondary completion code	9
	(Unused)	18
OPT	Output processing time	8
SPP	Simplex pages printed	9
DPP	Duplex pages printed	9
NLP	Number of lines printed	9
ENN	Entry number	5

Table 3-21. **Structure of host/tape output and mnemonics**

Mnemonic	Status field content	Maximum length (in characters)
PBC	Blank column	0

* Status (STA) is coded as:

- 1 unused
- 2 input
- 3 queued for output
- 4 output
- 5 done
- 6 aborted
- 7 aborted by reset
- 8 operator requested no recovery on rollover
- 9 report skipped for unknown reason



Note: The first block of a status file tape contains "STATUS FILE x," where x is 80 or 256 or 512. (This information is specific only to the SFS TAPE option.) The DISK ID and/or DATE and TIME parameters are found in the second block on the tape or the first block for the host-format disk file option. The blocks that follow contain the saved status file.

Every field in the saved status file (both tape and disk) has another leading field (in ASCII or EBCDIC, three characters long) containing the byte length of that field, including the additional 3-byte field length. For each job record within a block, a leading field (three characters long) contains the record length, including that leading field. Two records fit within a 512-byte block.

Structure of an SFS report entry

Each 512-byte block may contain up to three report entries. SFS accesses the entries in the status file report and formats them as described above. The following format represents a report entry from the status file on the DocuPrint 180. The contents for each report entry are defined in the figures below.

Table 3-22 shows status file report entry contents.

Table 3-22. **Status file report entry contents**

Bytes	Report entry contents
0-1	Status message (other bits in this word may be set) = ABORT 800 ₁₆ = DONE 500 ₁₆ = IN/PRNTG 400 ₁₆ = PRINTING 200 ₁₆ = QUEUED 100 ₁₆ = INPUT 20 ₁₆ = RESET 1 ₁₆ = RCVY N/S
2-3	Job number
4-5	Report number
6-11	Time: current year minus 1900; month; day; hour; minute; second

Table 3-23 shows status file input data contents.

Table 3-23. **Status file input data contents**

Bytes	Data from input processing
12-17	Customer job number
18-33	Customer report name
34-39	JDE
40-45	JDL
46-47	Maximum copy count
48-51	Pages written to disk
52-82	Department name
83	Report number suffix
84-85	Tape mounts

Table 3-24 shows status file output data contents.

Table 3-24. **Status file output data contents**

Bytes	Data from output processing									
86-89	Graphic images read									
90-92	Input processing time HH/MM/SS									
93	Flag byte: <table border="0"> <tr> <td><u>Bit 0</u></td> <td><u>Bit 1</u></td> <td><u>Bit 2</u></td> </tr> <tr> <td>0 = non-collate</td> <td>0 = simplex</td> <td>0 = not copy sensitive</td> </tr> <tr> <td>1 = collate</td> <td>1 = duplex</td> <td>1 = copy sensitive</td> </tr> </table>	<u>Bit 0</u>	<u>Bit 1</u>	<u>Bit 2</u>	0 = non-collate	0 = simplex	0 = not copy sensitive	1 = collate	1 = duplex	1 = copy sensitive
<u>Bit 0</u>	<u>Bit 1</u>	<u>Bit 2</u>								
0 = non-collate	0 = simplex	0 = not copy sensitive								
1 = collate	1 = duplex	1 = copy sensitive								
94-97	Completion code (in ASCII) / abort message (in ASCII)									
98-99	Customer job number (extension <MDSH> most significant)									
100-125	Unused									
126-127	Copies printed									
128-131	Pages printed									
132-135	Pages to bin									
136-137	Pages to tray									
138-141	Graphic pages printed									
142-143	Report completion code (binary)									
144-145	Secondary report completion code (binary)									
146-149	Unused									
150-152	Output processing time (HH/MM/SS)									
153-156	Simplex pages printed									
157-160	Duplex pages printed									
161-164	Lines printed									
165-167	End time stamp									

Forward and backward pointers

The forward pointer contains the sequence number of the next 512-byte block on the tape to be used. The first block is numbered 0, the second block 1, and so on. This is the next forward block on the tape except when a disk error occurs. The backward pointer provides the sequence number of the previous 512-byte block used on the tape.

Dumping the status file to tape

To obtain a dump of the status file copied to tape, use the ASCII JDE and the DUMP JDL supplied from the system software tape.

Audit Log

There are two utilities available through the System Control icon on the PC User Interface:

- Audit Log Utility and
- Audit Log Administration Utility.

Waste Management using the Audit Log Utility

Waste Management data is reported via the Audit Log Utility to provide information on how many sheets were delivered to the purge tray and removed by the operator (as in the case of a paper path jam). This helps you have more control over sensitive paper stocks.

The printer reports the number of sheets fed, delivered, and purged. The number of sheets removed is calculated by subtracting the delivered sheets from the total sheets fed.

Audit Log Utility

With the Audit Log Utility, Audit logs may be displayed, printed, or saved.

The new windows allow operator control of the Report Auditing Features of the DocuPrint 180 LPS. The new windows are:

- Audit Log
- Audit Log Print Options
- Audit Log Display Options
- Audit Log Display
- Audit Log Save
- Audit Log Search
- Reconciliation
- Sequence Error
- Stale Date Error.

Refer to the *DocuPrint 180 LPS PC User Interface Reference* for illustrations of each of these windows.

Audit Log Window

The Audit Log Window contains targets to allow you to close the window, list audit logs, display the Audit Log Print Options window, stop printing of an audit log report, stop an audit log save, and display the Audit Log Display Options window, and save and audit log.

You can select to display either a list of audit logs, showing the name of each report that is logged or a list of jobs that have audit logs, showing the name of each job that has audit reports logged. You can choose to display the list sorted by name or date in ascending or descending order. You can display lists that have selected error

values or the entire list of logs. Reports or jobs can be selected from the list or entered from the keyboard for further action. Selecting an audit job places all the options on a job instead of report basis. This makes access to a set of reports for a job more productive.

Audit Log Print Options Window

The Audit Log Print Options window contains a target to allow you to close the window, and apply and print the audit log with the selected print options.

You will select among the reporting options of Summary or Detail. You can select the content options of All, No Errors, Purge/Jam, Operations, Sequence Errors, Duplicate Errors, Reconciliation Errors, and Purge/Jam/Operational After data logged. Print options will be obtained from the Audit Log Options window and may be overridden by job stream data or by making new selections.

The report or job printout can have three major fields.

- **Heading** - This includes everything from the top of the report to the second dotted line. See figures 3-6 and 3-7.
- **Details** - This includes everything from the second dotted line from the top of the report to just before the line with "End of Report Reconciliation." See figure 3-6.
- **Trailer** - This includes everything from the line with "End of Report Reconciliation" to the end of the report. See figures 3-6 and 3-7.

Figure 3-6 illustrates a detail report.

Figure 3-6. **Detail Report**

Report Number: 1		Cust Job Number: 1		Audited Report Detail		Page 1																											
Job Number: 1		Job Date: 01/09/98		Report Name: ABC		Reconciliation, Sequence, Duplication Errors																											
Report Start Time: 09:31:30		Report Completion Status: Normal		Department: XYZ		Report Date: 01/09/98																											
				System: XXXXX		Process Date: 01/09/98																											
				Report End Time: 09:36:19		Report Duration: 00:04:49																											
						Log Expires: 01/09/98																											
MICR Mode: Not Installed		Bar Code Reader:		Lead Edge: Stress		Print Darkness: 5																											
Doc Nbr		Amount		Feed		Destination		Count		Time		Status/Identifying Text																					
				1 2 3 4 5 8 7 S A B C D E P																													
1		2.00										abc																					
8		2.00		2		A		*		1 09:31:38		def																					
										09:32:49		CONTINUE																					
										09:32:52		OS0020 Resuming OUTPUT																					
				2				p		1 09:33:29		12																					
				2				p		1 09:33:31		12																					
				2				p		1 09:33:33		12																					
				2				p		1 09:33:35		12																					
				2				p		1 09:33:41		12																					
				2				p		1 09:33:41		12																					
										6 09:33:41		Purged sheets																					
										4		Sheets removed																					
										09:35:33		CONTINUE																					
										09:35:37		OS0020 Resuming OUTPUT																					
2		2.00		2		A		*		1 09:36:15		ghi																					
3		2.00		2		A				1 09:36:15		jkl																					
4		2.00		2		A				1 09:36:15		mno																					
5		2.00		2		A				1 09:36:18		pqr																					
6		2.00		2		A				1 09:36:18		stu																					
7		2.00		2		A				1 09:36:18		vwx																					
8		2.00		2		A		+		1 09:36:17		yza																					
9		2.00		2		A				1 09:36:17		bcd																					
10		2.00		2		A				1 09:36:17		efg																					
				2		S				1 09:36:19		01																					
End of Report Reconciliation																																	
		Amount		Count		Feeders							Destinations																				
						1		2		3		4		5		6		7		S		A		B		C		D		E		P	
Report Totals		22.00		11		0		17		0		0		0		0		0		1		10		0		0		0		0		6	
Purged																				0													
Expected		20.00		10																													
Difference:		2.00		1																													
Scrap Report:														Waste Management Summary:																			
Sheets Fed:		21																				Purged Sheets:		6									
Sheets Delivered:		17																				Sheets Removed:		4									
Purge Deliveries:		6																															
Operator Removed:		4																															

Figure 3-7 illustrates a summary report.

Figure 3-7. Summary Report

```

Audited Report Summary
Report Name: ABC Reconciliation, Sequence, Duplication Errors
Department: XYZ
System: XXXXX
Report End Time: 09:36:19
Page 1
Report Number: 1
Job Number: 1 Cust Job Number: 1
Job Date: 01/09/98
Report Start Time: 09:31:30
Report Completion Status: Normal
Report Date: 01/09/98
Process Date: 01/09/98
Report Duration: 00:04:49
Log Expires: 01/09/98
-----
MICR Mode: Not Installed Bar Code Reader:
Lead Edge: Stress Print Darkness: 5
-----
Doc Nbr Amount Feed ----- Destination ----- Count Time Status/Identifying Text
1 2 3 4 5 6 7 8 A B C D E P
-----
End of Report Reconciliation
Report Totals Amount Count Feeders ----- Destinations -----
Purged 22.00 11 0 17 0 0 0 0 0 0 1 10 0 0 0 0 6
Expected 20.00 10
Difference: 2.00 1
-----
Scrap Report: Waste Management Summary:
Sheets Fed: 21 Purged Sheets: 6
Sheets Delivered: 17 Sheets Removed: 4
Purge Deliveries: 6
Operator Removed: 4
*****

```

Heading The heading consists of the fields shown in figure 3-8. Headings are printed for both Summary and Detail printouts.

Figure 3-8 illustrates a heading.

Figure 3-8. Heading

```

Audited Report Summary
Report Name: ABC Reconciliation, Sequence, Duplication Errors
Department: XYZ
System: XXXXX
Report End Time: 09:36:19
Page 1
Report Number: 1
Job Number: 1 Cust Job Number: 1
Job Date: 01/09/98
Report Start Time: 09:31:30
Report Completion Status: Normal
Report Date: 01/09/98
Process Date: 01/09/98
Report Duration: 00:04:49
Log Expires: 01/09/98
-----
MICR Mode: Not Installed Bar Code Reader:
Lead Edge: Stress Print Darkness: 5
-----

```

Table 3-25 shows the definitions of the heading fields.

Table 3-25. **Definitions of the heading fields**

Field	Definition
Report Number Job Number Report Date Job Date System Report Start Time Report End Time Report Duration Report Completion Status	All assigned by the system.
Process Date	Equal to BTEXT PRD value, or current date if omitted. Allows synchronization with date job was created on host. Will modify relative LDT values (e.g. W for weekly) to keep relative LDT specifications in sync.
Report Name	Equal to BTEXT RNA value
Cust Job Number	Equal to BTEXT CJN value.
Department	Equal to BTEXT DPT value.
Log Expires	Equal to (or calculated by) BTEXT LDT value.
System Status	All values are as of the instant the report began printing. Any changes in status would be reflected in the Operational details of the log.

Details The details consist of the fields shown in figure 3-9.

The Details section shows each purged sheet by feeder number and destination (P), as well as the cause of the purge, e.g. fuser cleanup (status code = 12). These appear in chronological order within the report details.

Figure 3-9 illustrates details.

Figure 3-9. **Details**

Doc Nbr	Amount	Feed ----- Destination ----- Count											Time	Status/Identifying Text				
		1	2	3	4	5	8	7	S	A	B	C			D	E	P	
1	2.00																	abc
8	2.00		2															def
																		CONTINUE
																		OS0020 Resuming OUTPUT
			2															12
			2															12
			2															12
			2															12
			2															12
			2															12
																		Purged sheets
																		Sheets removed
																		CONTINUE
																		OS0020 Resuming OUTPUT
2	2.00		2															ghi
3	2.00		2															jkl
4	2.00		2															mno
5	2.00		2															pqr
6	2.00		2															stu
7	2.00		2															vwx
8	2.00		2															yz
9	2.00		2															bcd
10	2.00		2															efg
			2															01
																		S

Table 3-26 shows the definitions of the details fields.

Table 3-26. **Definitions of the details fields**

Field	Definition
Doc Nbr	The document number. Equal to BTEXT SEQ value.
Amount	The dollar amount if a check is the document. Equal to BTEXT PRA value.
Feeder	Identifies the paper tray from which the paper was fed.
Destination	Identifies the bin or tray to which the document was sent.
Count	Identifies the number of sheets delivered for that document number. Duplicate errors are indicated by the plus sign (+) in the report details. Sequence errors are indicated by the asterisk (*) in the report details.
Time	Identifies the time of the event in the status/identifying text field.
Status/Identifying Text	<p>Identifies the event that occurred as shown on the PC UI message window. If a good delivery, then equal to BTEXT TXT value or 01 in the case of a sheet with no sequence attached; If a Purge sheet, then a status code; otherwise a Waste Management, system message, or command entered at the console by the operator. Up to 65 characters may be displayed in this field.</p> <p>The code (shown in figure 3-9 as "01") for showing the status of the sheets that did not have BTEXT="TXT=...".</p> <p>The code is represented by "XY",</p> <p>Where X is the reason if the "Y" value is 2 or that the sheet was delivered if the "Y" value is 0. See table 3-27.</p> <p>and</p> <p>where Y is the integrity value explaining how the sheet or set did. See table 3-28.</p>

Tables 3-27 and 3-28 list values for X and Y, respectively.

Table 3-27. **Value for X**

Value of X	Reason
0	Delivery
1	Fuser cleanup
2	Sequence error
3	Video error
4	Ordered stock
5	Misregistration
6	Barcode "no read" error
7	Fault
8	Stray sheet
9	Barcode "unexpected read" error

Value of Y	Reason
1	Good sheet
2	Scratch sheet

As an example, a status value of 72 would be "scratch sheet due to fault".

Trailers The trailer consists of the fields shown in figure 3-10.

The waste management data from the printer appears in the report as the "Purged sheets, Sheets removed" entry pair. The count value for the Purged sheets entry equals the sum of the purge sheet details immediately preceding it in the report. The totals for these entries are shown in the trailer under the "Waste Management Summary:" heading.

Figure 3-10 illustrates trailers.

Figure 3-10. Trailers

	Amount	Count	Feeders							Destinations							
			1	2	3	4	5	6	7	S	A	B	C	D	E	P	
Report Totals	22.00	11	0	17	0	0	0	0	0	0	1	10	0	0	0	0	6
Purged											0						
Expected	20.00	10															
Difference:	2.00	1															

Scrap Report:		Waste Management Summary:	
Sheets Fed:	21	Purged Sheets:	6
Sheets Delivered:	17	Sheets Removed:	4
Purge Deliveries:	6		
Operator Removed:	4		

Table 3-28 shows the definition of the trailer fields.

Table 3-28. Definitions of the trailer fields

Field	Definition
Report Totals	Total monetary value (sum of BTEXT PRA values for delivered sheets) and count of all good sheets delivered for the report, including system pages. Also shows source of all sheets fed (this includes purged, but excludes removed by operator) that were delivered, and destination of all good sheets.
Purged	Purge deliveries by destination to Sample and/or Purge tray.
Expected	Total monetary value (BTEXT RRA) and count of sheets expected (BTEXT NSE value) for the report.
Difference	Monetary and count differences between actual good sheets delivered and what was expected (per BTEXT RRA and NSE) for the report.
Sheets Fed	Total number of sheets fed for the report.
Sheets delivered	Total good sheets delivered to their intended destinations for the report.
Purge Deliveries	Number of sheets delivered to the Purge tray (e.g. for fuser cleanup after a jam) or Sample tray (i.e. if jam occurs above stackers).
Operator Removed	Number of sheets removed from the paper path before reaching their intended destinations (e.g. for paper jam clearance).

Table 3-28. **Definitions of the trailer fields** (continued)

Field	Definition
Purged Sheets	Identifies the number of sheets delivered for that document number. Duplicate errors are indicated by the plus sign (+) in the report details. Sequence errors are indicated by the asterisk (*) in the report details.
Sheets Removed	Number of sheets removed from the paper path before reaching their intended destinations (e.g. for paper jam clearance).

Audit Log Display Options Window

The Audit Log Display Options window contains a target to allow you to close the window and display the report with the selected options.

You can select among the reporting options of Summary or Detail. You can select the content options of All, No Errors, Sequence Errors, Reconciliation Errors, Duplicate Errors, and Purge/Jam/Operational After data logged. Display options can be obtained from the Audit Log Options window and may be overridden by job stream data or by making new selections.

Audit Log Display Window

The Audit Log Display window contains a target to allow you to close the window, to stop the Report display, and to display the audit log search window. This window displays a formatted audit log based on the parameters chosen in the Audit Log Window and the Display Options window. Components of the report structure such as column headings and totals are integral with the formatted display data, and may scroll out of view. Selected options will show on the window.

Audit Log Save Window

The Audit Log Save window contains targets to allow you to close the window and to apply the selected save options.

You select one of the following destination options of either the

- PC UI hard disk
- PC UI floppy disk.

The report name selected from the Audit Log Window will be displayed.

Audit Log Search Window

The Audit Log Search window enables searching for specific contents of the displayed Log. It searches for the contents of the Search For: field.

The Entire File button is the default selection when the window is opened. You would enter in the Search For: field the character string you wish to search for. When the string is found, it is highlighted in the Audit Log Display window.

The Audit Log Search window responds to the found character string by automatically changing the Search: button selection from Entire File to Forward and waits for the user to initiate another Search Next.

Reconciliation Error window

The Reconciliation window appears if the “Stop if reconcile error“ option was selected in the Audit Log Defaults window (see Audit Log Administration later in this section) and if a reconciliation error has occurred.

It contains a target to apply the selection of buttons in the window and closes the window. You can select a button to either ignore the error and continue printing or display the log.

One of three selections needs to be made when this window appears:

- Ignore the error and continue
- Ignore the error for the rest of the job
- Abort the job.

The system is locked up until a selection is made and the Apply/ Close header button is selected.

Sequence Error window

The Sequence Error window appears if the “Stop if sequence error“ option was selected in the Audit Log Defaults window (see Audit Log Administration later in this section) and if a sequence error has occurred.

It contains a target to apply the selection of buttons in the window and closes the window. You can select a button to either ignore the error and continue printing or display the log.

One of three selections needs to be made when this window appears:

- Ignore the error and continue
- Ignore the error for the rest of the job
- Abort the job.

The system is locked up until a selection is made and the Apply/ Close header button is selected.

Stale Date window

The Stale Date window is displayed when there is a stale date fault on an Audit Logged job. The stale date conditions are set up through the BTEXT commands in the job stream.

One of three selections needs to be made when this window appears:

- Ignore the error and continue
- Ignore the error for the rest of the job
- Abort the job.

The system is locked up until a selection is made and the Apply/Close header button is selected.

Audit Log Constraints/Exceptions

The detail reporting option will only be available if the system has the bar code option installed.

Reports cannot be printed until the printer is idle.

No user entered file extensions are allowed. All file extensions will be supplied by the system.

All Audit Logs are on the PC UI hard disk.

If 'Job' is selected, it may take a while to format the details of several, large reports.

Audit Log Administration Utility

With the Audit Administration Log Utility, Audit logs and their reporting options may be administered.

The new windows allow operator control of the Audit Log Administration Features of the DocuPrint 180 LPS. The new windows are:

- Audit Log Administration
- Audit Log Default Options
- Audit Log Disk Usage
- Disk Approaching Full.

Refer to the *DocuPrint 180 LPS PC User Interface Reference* for illustrations of each of these windows.

Audit Log Administration window

The Audit Log Administration window will contain targets to allow you to close the window, to list audit logs, to delete a log, provoke a window that displays disk usage status, and to provoke a window to set default options. A large menu displays the Log names of all audit logs on the PC UI. Confirmation is required for all deletions.

Audit Log Default Options window

The Audit Log Default Options window will contain targets to allow you to close the window, and to apply the current options. Check boxes will be used to select whether a STOP command should be issued automatically if the report does not reconcile; whether a STOP command should be issued automatically if the report has sequence errors; You can select the audit reporting control parameters of the BTEXT command if not explicitly specified in the command (i.e default setting, or job stream data.) The selected settings will be maintained across sessions as the default settings.

Audit Log Disk Usage window

The Audit Log Disk Usage window will contain targets to allow you to close or redisplay the window, Output fields show the total amount of disk space all audit logs consume in kilobytes, as a percentage of available space, and the number of log files.

Disk Approaching Full

The disk is considered approaching full when it exceeds 85% full. This window will first appear after the disk reaches 86% full and will reappear every additional WHOLE percentage thereafter. This condition is checked at the completion of every report or after every 1000 records logged against any report. The starting percentage will be reset to 86% if the session is restarted or if Audit Logs are deleted and the disk is less than 86% full. If Audit Logs are deleted and the disk is still more than 86% full, the current disk full percentage is saved as the starting percentage.

4. Editing source files

This section describes the LPS editor, the editor commands, running editing sessions, and modifying records.

What the editor is

The LPS editor is a utility which provides file editing facilities as part of normal operator communication and control. These facilities are available for the creation, modification, and maintenance of files when the system is idle. The editor has commands for creating and modifying source files (.JSL, .FSL) as well as commands for the maintenance of any type of system disk file.

How the editor works

The editor allows direct access to each line (or record) of a source file. The user enters single-line commands which provide the following facilities:

- Creating a sequenced source file
- Inserting, reordering, and replacing lines or groups of lines of text
- Reordering and renumbering records within a file
- Merging part of one file into another
- Matching, moving, and substituting character strings within a specified range of text lines
- File maintenance (allowing the user to copy, rename, and delete files).



Note: The PC Editor window (available from the Utilities menu) can also be used to create a source file.

Source files

When a file is being modified, the keyed-in commands are applied to a temporary file in working storage, not to the permanent file on a system disk. When a request is made to edit an existing file, the specified file is copied into working storage. All editing operations are performed upon the working file, and it is saved permanently on a system disk only when specifically requested by the user. The working file is also saved from one editing session to another unless explicitly cleared, but any intervening use of the editor is likely to destroy its contents.

Viewing source files

Use the TYPE command to view a source file without using the editor record commands GET and DISPLAY.

Storing source files

The LPS has one system disk drive, DP0:, with the option of three additional disk drives, DP1:, DP2:, and DP3:. To optimize file access, files of a given type are stored on a preferred drive; this is known as file biasing. The actual biasing scheme varies depending on the number of drives available.

All source files are stored in DP0: in a one disk system. Source files tend to be stored on DP0: in a two-disk system, whereas they tend to be stored on the highest disk number in a three- or four-disk system. The user may store all source files on any system disk by specifying a *disk-id*. To access a specific system disk directly, the *disk-id* is specified in the command as DP0:, DP1:, DP2:, or DP3:. If a *disk-id* is not entered, the system transfers the files to or from the appropriate disk.

Each source file has line numbers associated with its data records. These line numbers are supplied by the editor during file creation. Files entered with open-reel tape have line numbers added to them at the time they are entered into the system. Source files entered from the online subsystem through HOSTCOPY must have line numbers added with the editor (refer to the MERGE command).

Files you can edit

You can edit source files which are cataloged in the file type directories shown in table 4-1.

Table 4-1. **Source file type directories**

File type	Contents
CMD	Command file source files
DAT*	Data files
FSL	Form source language files
JSL	Job source language files
LIB*	Utility file
MSC*	Temporary user files
PCH	Patch files
TMP*	Temporary user files

*Some files of this type can be edited but some cannot.

You cannot edit non-source files. They are cataloged in the file type directories shown in table 4-2.

Table 4-2. **Non-source file type directories**

File type	Contents
CME	Copy modification entry object files
FIS	Font interchange standard files
FNT	Form data files
FRM	Form object files
ICT	Ink catalog
IDR	Ink directory
IMG	Digitized graphic image files
JDL	Job descriptor library object files
LIB*	Utility file
LGO	Logo data files
LOG	System integrity log files
OSD	OSDS diagnostic files
PDE	Page descriptor entry object files
SAF	SAFES diagnostic files
STK	Stockset files
SYS	System data files
\$Y\$	System files
TPF	Temporary patch files
TSK	System task image files
TST	RTEXT object files

*Some files of this type can be edited but some cannot.

You can access files within these directories only by a subset of the editor commands. These commands list file type directories (LIST and FILE), delete files (DELETE), copy files (COPY), or rename files (RNAME).

Editor commands

This subsection describes the editor commands.

Three types of editor commands

Editor commands fall into three categories:

File/tape/floppy commands	Commands that apply to an entire file, tape, or floppy. These commands may be given at any time.
Record commands	Commands that act upon a record or a group of records within a file. These commands may be given only after a file has been selected for editing.
Intrarecord commands	Commands that make changes within an individual record. These commands generally manipulate character strings and may be given only after a specific set of records has been selected.

The syntax of each command is explained in the following section along with examples of use. A special category of commands, editor command file commands, is listed and described in the "Using command files" chapter.



Note: The following editor commands may also be initiated at the system level—COPY, DELETE, FID, FILE, FLOPPY, LIST, REVIEW, RNAME, TAPE, TYPE.

Composite editor commands

Four editor commands are composite commands and perform the combined functions of two or more editor command functions. These commands and their descriptions are discussed below.

CE	Removes all data lines from the working file and terminates the editing session. Refer to the CLEAR and END commands for further information.
SCE [disk-id][file-id[,n[,m]]]	Saves the contents of the working file (or a subset of it) into permanent file storage, removes all data lines from the working file, and terminates the editing session. Refer to the SAVE, CLEAR, and END commands for further information.
GPCE [disk-id]file-id[n-m]	Brings an existing file as specified into working storage, formats for printing all or selected lines of the specified file. It then removes all data lines from the working file, terminates the editing session, and delivers a hard copy listing to the sample tray (or a stacker). <i>This command cannot be abbreviated to the first three characters.</i> Refer to the GET, PRINT, CLEAR, and END commands for further information.
PSCE [disk-id][file-id[,n[,m]]]	Performs the same as the SCE command but also prints a hardcopy of the working file.

Editor syntax conventions

The conventions used to define editor command syntax are listed below. Also refer to this manual's introduction for basic syntax and masking conventions.

- You can enter all commands (except composite commands such as SCE) in full or enter only the first three characters of the command. For example, you can enter RENUMBER or REN.
- When keying in a composite command (CE, SCE, or GPCE), all characters must be entered.
- All commands are executed by pressing the <ENTER> key.
- Examples of system/user dialog key ins are underlined.
- *File-id* refers to the name of a file and the file directory in which it is cataloged. Thus, when specifying file id, the user must key in *file-name.file-type*. File name is a 1- to 6-character file name. File type is one of the file directories under which the file name is cataloged. For example, SIGMA.JSL would be the file id of a file named SIGMA which is cataloged in the JSL file directory.
- When specifying the file id in file maintenance commands, the masking feature can be used unless otherwise noted. An asterisk (*) or question mark (?) is used as a wild-card or string lookup for the file name. The asterisk makes single and multiple alphanumeric characters, even a blank, a match. The question mark makes only single alphanumeric characters, including a blank, a match.
- *Disk-id* refers to one of two to four system disks, which are defined as DP0:, DP1:, DP2:, or DP3:. File manipulation editor commands which do not specify a disk number transfer the files to or from appropriate disk according to file type. You can direct access to or from a specific disk.
- The maximum line number for a file is 32,765. The maximum number of records that may be created in a file is 10,000.
- The *n-m* parameters in the command syntax represent values as described in table 4-3.

Table 4-3 shows the *n-m* parameters and their values.

Table 4-3. **Editor *n-m* parameters**

Parameter	Value
<i>n-m</i>	Starting and ending line number for command
<i>n</i>	Only line number for this command
<i>n-</i>	Starting line number to end of file
<i>-m</i>	Beginning of file to ending line number.

Using the <TAB> key

You can use the <TAB> key in character mode to insert 8 spaces into text.

You can use the <TAB> key in the PC UI window mode to move the cursor to the next data entry field. When the editor runs from the PC UI window mode, the <TAB> key has no function; it does not insert 8 spaces into text.

Stack printing

The stack printing feature permits you to spool editor print files such as the output of LIST, PRINT, and TAPE LIST,PRINT commands. When the END command terminates the editor, all files which are stacked in the editor spool file are printed. The printed output goes to the sample tray if 20 or fewer physical pages have been formatted. Because a stack is equivalent to a report, the following conventions apply:

- Page numbers start at 1 for each stack
- If the LPS is running in duplex, the first page of a new stack prints on the front side of a new page, not on the back side of the previous stack's last page.

Running an editing session

This subsection describes how to run an editing session.

EDIT

Use this command to create or modify a source file, invoke the editor by keying in EDIT at the LPS keyboard. If a source file already exists on disk, use this command to access or execute it.

Syntax EDIT [*disk-id*][*filename.file-type*|@*filename* [*.file-type*]]

Parameters Table 4-4 shows the parameters of the EDIT command.

Table 4-4. **Parameters of the EDIT command**

Parameter	Description
<i>disk-id</i>	Specifies the system disk (DP0:, DP1:, DP2:, or DP3:) on which the file resides.
<i>file-name</i>	A 1- to 6-character file name that exists on disk.
<i>file-type</i>	One of the file directories (which contains a source file) under which the file name is cataloged.
@ <i>file-name</i>	Specifies that the editor is to retrieve commands from an editor command file. If no file type is entered, the system defaults to CMD. Further information on editor command files is presented in the "Using command files" chapter.

If the *file-name.file-type* is specified, the requested file is brought into working storage. When creating a new file, a file id must be assigned when the newly created file is saved.

After you invoke the editor, the editor prompts for a command by displaying EDIT>.

Example

```
OS1000 READY FOR COMMANDS („TIME) HH:MM:SS
EDIT SYSPDL.JSL
EDIT>
```

CONVERT

Use this command to edit files containing lowercase characters using the uppercase keyboard. To insert or reference lowercase characters in a file, the lowercase characters to be converted are preceded by the pound sign (#). The pound sign is used by the editor to indicate the start of conversion to lowercase characters. The CONVERT command should be reset with the NOCONVERT command.

Syntax CONVERT

Example The CONVERT command can be used if the text line This is the end (which was entered into a disk file from either tape or the online interface) is to be altered. This text line could also be entered in lowercase at the keyboard.

```
EDIT>CONVERT
EDIT>MOD 10
000010 This is the end
EDIT>S/T#HIS/T#HAT/
000010 That is the end
EDIT>NOCONVERT
```

NOCONVERT

Use this command to reset the CONVERT command so that no uppercase to lowercase conversion takes place.

Syntax NOCONVERT

GET

Use this command to copy an existing file into working storage.

Syntax GET [*disk-id*]*file-id*

Parameters Table 4-5 shows the parameters of the GET command.

Table 4-5. **Parameters of the GET command**

Parameter	Description
<i>disk-id</i>	Specifies the system disk (DP0:, DP1:, DP2:, or DP3:) that contains the file.
<i>file-id</i>	Specifies the file.

Example OS1000 READY FOR COMMANDS (TIME) HH:MM:SS
 EDIT
 EDIT> GET SYS2.JSL

The *file-id* becomes associated with the working storage so that a SAVE command without a *file-id* restores working storage back into the file from which it came.

When the GET command is keyed in with a file existing in working storage that has not been saved (using the SAVE command), the THE FILE HAS NOT BEEN SAVED DO YOU WANT TO SAVE IT (YES/NO)? message displays.

A YES response causes the editor to ignore the GET command. The operator can then issue the SAVE command. If the response is NO, the GET command is invoked, and the previous contents of working storage are lost. A message is displayed if there is insufficient disk space to get a file.

KEYS

Use this command to view the beginning and ending line numbers of the working file.

Syntax KEYS

Example EDIT> GET SYS2.JSL
 EDIT> KEYS
 Beginning line number 000010
 Ending line number 000050

MERGE

Use this command to insert the contents of a disk file into working storage where specified and to add line numbers to source files created on the host computer and downloaded to the printer. The message CANNOT GET, FILE IMPROPERLY SEQUENCED appears if a file must be merged. The current records in working storage are not destroyed when the records of the new file are brought in.

Syntax MERGE [*disk-id*]*file-id*[,*n*][,*s*]

Parameters Table 4-6 shows the parameters of the MERGE command.

Table 4-6. **Parameters of the MERGE command**

Parameter	Description
<i>disk-id</i>	Specifies the system disk (DP0:, DP1:, DP2:, or DP3:) that contains the file to be moved into working storage.
<i>n</i>	Specifies the sequence number to be assigned to the first record of the file being merged with the contents of working storage. The default is the last record number in the working file +10.
<i>s</i>	Specifies the sequence increment. The default is 10.

SAVE

Use this command to save the contents of the working storage file (or a subset of it) as a disk file.

Syntax SAVE [*file-id*[,*n*[,*m*]]]

If no *file-id* is specified, the contents of the working storage file are saved under the name currently associated with it (as set by a previous GET command or as set when the editor is called). If no *file-id* is currently associated with the working storage file, the operator is prompted for a *file-id* under which to save its contents. A set of lines may be saved by specifying the starting line number *n* and the ending line number *m*. The default for *n* is the beginning of the file. The default for *m* is the end of the file.

If a *file-id* is given, and a file already exists with that *file-id*, the The file already exists, Do you want to overwrite it (Y/N)? message displays.

If YES is entered, the file is overwritten and its original contents are lost. If NO is entered, the SAVE command is ignored. A message is displayed if there is insufficient disk space to save a file.

CLEAR

Use this command to remove all data lines from the working file. It has no parameter options.

Syntax CLEAR

Example The following is an example of CLEAR command usage.

EDIT> CLEAR Clear working storage

EDIT> INSERT 10,10 Issue next command

Procedure If a file exists in working storage (and has not yet been saved) when the CLEAR command is entered, the editor responds with the following message:

The file has not been saved
Do you want to save if (YES/NO)?

If you enter N or NO, the CLEAR command takes effect, and the Work file cleared message displays.

If you enter Y or YES, the CLEAR command is ignored, and the Command ignored message displays.

You may then save the file by keying in the SAVE command.

END

Use this command to terminate an editing session. If you enter the END command and a file has not been saved using the SAVE command, the The file has not been saved Do you want to save it (Y/N)? message displays.

If you enter YES, the editor ignores the previous END command, and the message Command ignored displays. You may then enter the SAVE command. If you enter NO, the editor ends the editing session and retains the contents of working storage for a future session.

Control is returned to the system, and no more editor commands are accepted until the editor is requested again. The message OS1000 READY FOR COMMANDS is displayed after the END command takes effect.

Syntax END

Restrictions There is only one working storage file; any intervening use of the editor is likely to destroy its contents. (Refer to the CE, SCE, and GPCE commands described in this chapter.)

Modifying records

This group of commands manipulates the records of the file currently in working storage. Masking is not applicable.

DISPLAY (display lines)

Use this command to display a listing of all lines or selected lines of the current working storage file.

Syntax DISPLAY [*n*[-*m*]]

Parameters *n-m* are as previously defined in the "Editor syntax conventions" section in this chapter.

Examples The following are some examples of DISPLAY command usage.

DISPLAY	Display all lines.
DISPLAY 10-20	Display lines 10 through 20.
DIS 5-	Display lines 5 through the end of the file.
DIS -100	Display lines from the beginning of the file through line 100.
DIS - 100 - 125	Display lines from the beginning of the file through 100. Ignores the '-125'.

Restrictions The following restrictions apply to using the DISPLAY command:

- If the number of lines to be displayed does not exceed display capacity (nominally 23 lines), all requested lines are displayed.
- If the number of lines to be displayed exceeds display capacity, the operator is prompted with the message:

```
Enter return to continue
```



Note: RETURN refers to the <ENTER> key.

This informs the operator that all requested lines have not been displayed. To display another set of lines, the user presses the <ENTER> key. If any other response is given, the display is halted.

DUPLICATE (copy lines)

Use this command to copy a line or group of lines to another point in the file.

Syntax DUPLICATE *n[-m][,p[,s]]*

Parameters Table 4-7 shows the parameters of the DUPLICATE command.

Table 4-7. **DUPLICATE command parameters**

Parameter	Description
<i>n[-m]</i>	Starting and ending line numbers. For additional information, refer to the "Editor syntax conventions" section in this chapter.
<i>p</i>	Specified lines are moved to a new location starting with line number <i>p</i> .
<i>s</i>	New line numbers are incremented by <i>s</i> . The default value for <i>s</i> is 1.

FIND (find lines with match)

Use this command to display all lines in the specified range containing the given character string.

Syntax FIND [*n[-m]*]/*string*/

Examples The following are some examples of FIND command usage.

FIND /JDE/	Display each line that contains the characters JDE.
FIN 5-100/JOB/	Display lines 5 to 100 that have the characters JOB.
FIN 5/SYSTEM/	Display line 5 if it contains the word SYSTEM.
FIN 5-/SYSTEM/	Display lines 5 to end-of-file that contain the word SYSTEM.
FIN - 100/JOB/	Display each line from 1 to 100 that has the characters JOB.

INSERT (insert lines)

Use this command to add a line or group of lines into the file. You are prompted by the next sequence number and input terminates upon receiving a null line (such as pressing the <ENTER> key) or encountering an existing record.

Syntax {INSERT *n*[,*s*]} *n text*

Parameters Table 4-8 shows the parameters of the INSERT command.

Table 4-8. **Parameters of the INSERT command**

Parameter	Description
<i>n</i> [, <i>s</i>]	Lines are inserted starting at <i>n</i> and incremented by <i>s</i> . Line <i>n</i> must not already exist. The default value for <i>s</i> is 1.
<i>n text</i>	Any character string. Enter just the desired line number, one blank, and then the text.

Examples The following are some examples of INSERT command usage.

- Enter text on line 55, key in:

```
EDIT>55 ACCT USER=(BIN,TRAY);
```

With this type of insertion, the editor replaces an existing line without notifying you any further.

- Insert lines starting at 10 and increment by 10 for next line.

```
EDIT>INSERT 10,10
```

- Insert lines starting at 15 and increment by 1.

```
EDIT>INS 15
```

MODIFY (modify lines)

Use this command to specify a range of lines that can be modified by subsequent string commands. These lines are displayed after the MODIFY command has been issued. The default for the modify range is the entire file. If the operator specifies a modify range, all records in that range are displayed. String modification commands are defined in the "Modifying parts of records" section in this chapter.

Syntax MODIFY [*n*[-*m*]]

Examples The following are some examples of MODIFY command usage.

- Prepare to modify lines 5 through 100.

```
EDIT>MODIFY 5-100
```

- Substitute occurrences of JDE with JOB in lines 5 through 100.

```
EDIT>S/JDE/JOB/
```

- Prepare to modify line 5.

```
EDIT>MOD 5
```

If no match is found in the modify range, the NO MATCH FOUND message displays.

- Delete text HOST=OSWTR from line 5.

EDIT>D/HOST=OSWTR/

MOVE (move lines)

Use this command to relocate a line or group of lines to new line positions.

Syntax MOVE *n[-m],p[,s]*

Parameters Table 4-9 shows the parameters of the MOVE command.

Table 4-9. **MOVE command parameters**

Parameter	Description
<i>n[-m]</i>	Specified lines (<i>n-m</i>) are moved to a new location.
<i>p[,s]</i>	New location starts with line number <i>p</i> . The new line numbers are incremented by <i>s</i> . The default value for <i>s</i> is 1.

PRINT (print lines)

Use this command to send a printout to the sample tray (or a stacker) of all or selected lines of the current working storage file.

Syntax PRINT [*n[-m]*]

Procedure To print the file, the operator must exit from the editor (using the END command). The operator is permitted to spool editor print files. The current working storage file is intact if the user enters the editor after the file is printed. The listing is in portrait mode.

Three title lines precede the data requested for printing and appear on each logical page:

```

FILE NAME:      fileid   PRINTED ON:  date time
DISK ID:  disk id

1.....10.....20.....30.....40.....50.....60.....70.
    
```

The *file-id* is the name and type associated with the current working storage file. If *file-id* is ****NONE****, a hardcopy of the data was requested before the data in the working storage file was given a file name. This is possible if the data is a result of a MERGE command used to obtain the data of a file rather than using the GET command, the data is in a new file that has not been saved on the disk using the SAVE command, or the working storage file is restored upon entering the editor from READY FOR COMMANDS. (Refer to the "Composite editor commands" section in this chapter for more information.)

REMOVE (delete lines)

Use this command to delete a line or group of lines from the working storage file.

Syntax [REMOVE] *n*[-*m*]

Parameters Table 4-10 shows the parameters of the REMOVE command.

Table 4-10. **Parameters of the REMOVE command**

Parameter	Description
<i>n</i>	Specifies line to be deleted (required).
- <i>m</i>	If there is a range, <i>m</i> is also required (a hyphen cannot be used to imply the first or last line).
REMOVE	Another way to remove a line from the working storage file is by entering the number of the line to be removed. To delete a group of lines, enter the beginning and ending line numbers separated by a hyphen.

Examples The following are some examples of REMOVE command usage.

- Remove lines 5 through 25.

```
EDIT>REMOVE 5-25
```
- Remove line 15.

```
EDIT>REM 15
```
- Delete line 55.

```
EDIT>55
EDIT>1 RECORD DELETED
```
- Delete lines 10 through 40 if the line numbers are incremented by 10.)

```
EDIT> 10-40
EDIT>4 RECORDS DELETED
```

RENUMBER (renumber lines)

Use this command to renumber the lines of the working storage file. The default value for *n* and *s* is 10.

Syntax RENUMBER [*n*[,*s*]]

Parameters Table 4-11 shows the parameters of the RENUMBER command.

Table 4-11. **Parameters of the RENUMBER command**

Parameter	Description
<i>n</i>	Specifies beginning line number.
<i>s</i>	Successive data lines are assigned numbers in increments of <i>s</i> .

REPLACE (substitute lines)

Use this command to replace a line or group of lines, and to insert lines in their place.

Syntax REPLACE *n*[-*m*][,*s*]

The above command is identical to the following commands:

```
REMOVE n-m
INSERT n[ ,s]
```

A line may also be replaced by keying in the line number, a blank, then text.

SORT (sort lines)

Use this command to put the records of the working file in alphabetical sequence. Character positions 2 through 7 are considered when reordering the records. Although the SORT command is designed to be used on the output from the FID command, any working file may be sorted.

The non-sorted character positions are available to be occupied by "tags" to enable easy substitution at a later date for the creation of command files.

Syntax SORT

Procedure As filenames typically consist of uppercase letters and numbers, the sort priority used by the SORT command is:

```
?, A,B,C...Z,0,1,2,...9
```

where ? is any character which is not an uppercase letter or a number. A lowercase letter or special character is ignored (as are any following characters), and the record is written to the beginning of the file.

Restrictions If the editor work file is empty when you issue the SORT command, the system displays the message Nothing to sort.

Example In the following example FID creates a file named FSLFIL.CMD which lists all the .FSL files existing on the LPS disks. Next GET is used to put the FSLFIL.CMD file in the editor's working area. Then FSLFIL.CMD is sorted, creating the FSLFIL.TMP file which is printed (or displayed).

```
EDIT> FID FSLFIL.CMD FSL
EDIT> GET FSLFIL.CMD
EDIT> SORT
EDIT> PRINT (or DISPLAY)
```

STEP (display next line)

Use this command to display each record of a file, one at a time, starting with the next record as specified in the MODIFY command.

Syntax STEP
 or
 <ENTER>

Example The following are some examples of STEP command usage.

EDIT> MODIFY 10 Set modify range
 ...line 10 is displayed...

EDIT> STEP
 ...text of next line after line
 10 is displayed...

EDIT> S/MOST/HOST Change text in next line after line 10
 ...1 string changed

EDIT> STEP
 ...next line is displayed

EDIT>... User may enter an intrarecord command
 .. (such as S,F,P, or D) or another STEP
 ..

Pressing the <ENTER> key is an implied STEP command. If there is an active working file, the modify range is stepped as if a STEP command had been issued. When an attempt is made to step past the last record of the file, the message Invalid sequence number is displayed.

Modifying parts of records

Use the following commands (C, D, F, O, P, S) to modify character strings within a record or records. Use these commands with the MODIFY command which specifies the record or records to be altered. After altering a line, the system displays it.

C (change string)

Use this command to change one character string to another character string.

Syntax C [k]/string₁/string₂/

Parameters Table 4-12 shows the parameters of the C command.

Table 4-12. **Parameters of the C command**

Parameter	Description
<i>k</i>	Indicates that the <i>k</i> th occurrence of the character string in each line is to be affected (<i>k=0</i> indicates that all occurrences of the specified character string are to be affected). When <i>k</i> is specified, it must be preceded by one blank character. The default for <i>k</i> is the first occurrence of the specified character string.
<i>string</i> ₁	Replaced by <i>string</i> ₂ for lines specified in the MODIFY command.
<i>string</i> ₂	Changed to <i>string</i> ₁ for lines specified in the MODIFY command.

D (delete string)

Use this command to delete a character string from a record or records in the working storage file.

Syntax D [*k*]/*string*/

Parameters Table 4-13 shows the parameters of the D command.

Table 4-13. **D command parameters**

Parameter	Description
<i>k</i>	Has the same meaning as in C (change string).
<i>string</i>	Default value is 1 character string deleted from lines specified in the MODIFY command. In addition to the slash (/), a dollar sign (\$), or single quote (') may be used as an intrarecord command delimiter.

Examples The following are some examples of D command usage.

- Modify line 5. Line 5 is displayed.
EDIT>MODIFY 5
- Delete the first occurrence of SYSTEM from line 5.
EDIT>D/SYSTEM/
- Range of lines which may be altered is 5 through 70.
EDIT>MOD 5-70
- Delete all occurrences (specified by *k=0*) of JDE from each line, starting with line 5 and proceeding through line 70. If *k* were set to 2 (instead of 0), only the second occurrence of JDE in each line would be deleted.
EDIT>D 0/JDE/

P (insert prior)

Use this command to insert text **prior** to a character string.

Syntax P [*k*] /*string*₁/*string*₂/

Parameters Table 4-14 shows the parameters of the P command.

Table 4-14. **Parameters of the P command**

Parameter	Description
<i>k</i>	Has the same meaning as in C (change string).
<i>string</i>	Insert <i>string</i> ₂ just prior to <i>string</i> ₁ for lines specified in the MODIFY command.

F (insert following)

Use this command to insert text **following** a character string.

Syntax F [*k*] /*string*₁/*string*₂/

Parameters Table 4-15 shows the parameters of the F command.

Table 4-15. **Parameters of the F command**

Parameter	Description
<i>k</i>	Has the same meaning as in C (change string).
<i>string</i>	<i>string</i> ₂ is inserted following <i>string</i> ₁ for lines specified in the MODIFY command.

O (overlay string)

Use this command to overlay text by a character string. It is useful for replacing characters which are unknown or cannot be created from the keyboard or for writing over a string without having to key in the entire *string*₁. (Compare this action with the substitute string command.)

Syntax O [*k*] /*string*₁/*string*₂/

Parameters Table 4-16 shows the parameters of the O command.

Table 4-16. **Parameters of the O command**

Parameter	Description
<i>k</i>	Has the same meaning as in C (change string).
<i>string</i> ₁	The <i>k</i> th occurrence of <i>string</i> ₁ is overlaid by <i>string</i> ₂ for lines specified in the MODIFY command.
<i>string</i> ₂	The <i>k</i> th occurrence of <i>string</i> ₁ is overlaid by <i>string</i> ₂ for lines specified in the MODIFY command.

Example The following are some examples of O command usage.

```
EDIT> MODIFY 5           Modifies line 5
000005 Fred Fred Fred Fred Fred

EDIT>O /Fred/John/      Modifies first occurrence of the string
                        "Fred" with "John".

000005 John Fred Fred Fred Fred

EDIT>O 2/Fred/John/     Modifies second occurrence of the string
                        "Fred" with "John".

000005 John Fred John Fred Fred

EDIT>O 0/Fred/John/     Modifies all occurrences of the string
                        "Fred" with "John".

000005 John John John John John
```

S (substitute string)

Use this command to **substitute** one character string with another character string.

Syntax S [*k*]/*string*₁/*string*₂/

Parameters Table 4-17 shows the parameters of the S command.

Table 4-17. **Parameters of the S command**

Parameter	Description
<i>k</i>	Has the same meaning as in C (change string).
<i>string</i> ₁	Replaced by <i>string</i> ₂ for lines specified in the MODIFY command.
<i>string</i> ₂	Substituted for <i>string</i> ₁ for lines specified in the MODIFY command.

Example The following are some examples of S command usage.

```
EDIT>ABCDEFGHIJKL M    This is the initial string.
EDIT>S/GHX/GHI/       Enter this command.
EDIT>ABCDEFGHIJKL M    The modified string is displayed by the
                        LPS.
```

(Refer to and compare with the overlay string command.)

Examples of editor sessions

The following examples illustrate the use of some editor commands. All commands in these examples are invoked by pressing the <ENTER> key.

Create and save a .JSL file

This example shows you how to create and save a file in the JSL directory.

Example 1 Create and save a file in the JSL directory

```
OS1000 READY FOR COMMANDS (TIME) hh:mm:ss
EDIT
EDIT>INS 10,10
000010  SYSPDL:SYSTEM;
000020  VOLUME HOST = POWERVS, PLABEL = YES;
000030  BLOCK LENGTH = 2048;
000040  ACCT USER = (BIN,TRAY);
000050  21:JOB;
000060  END;
000070  <RETURN> or <ENTER>
EDIT>SAVE SYSPDL.JSL
EDIT>CLEAR
Work file cleared
EDIT>END
```

Modify and resave the .JSL file

This example shows you how to modify and resave the .JSL file.

Example 2 Modify and save file from example 1

```
EDIT
EDIT>GET SYSPDL.JSL
EDIT>KEYS
Beginning line number 000010
Ending line number 000060
EDIT>INS 11
000011 /* SYS JDL */
000012<ENTER>
EDIT>FIND 10-60/JOB/
21:JOB;...000050
EDIT>MODIFY 50
21:JOB;
EDIT>S/JOB/JDE/
21:JDE;
EDIT>DIS 40
ACCT USER=(BIN,TRAY);...000040
EDIT>REMOVE 40
1 RECORDS DELETED
EDIT>REN
EDIT>DISPLAY
'
.
.
.
} Lines appearing on display
'

EDIT>SAVE
EDIT>CLEAR
WORK FILE CLEARED
EDIT>END
```

List other .JSL files and insert one into the .JSL file

This example shows you how to use file manipulation commands.

Example 3 File manipulation commands

```
OS1000 READY FOR COMMANDS (TIME) hh:mm:ss
EDIT
EDIT>COPY SYSPDL.JSL TSTSYS.JSL
CREATING FILE TSTSYS.JSL
EDIT>FILE JSL
```

```
OSWTR.JSL
IBMPDL.JSL
TSDPDL.JSL
.
.
.
TSTSYS.JSL
```

} List of all files in JSL director

```
EDIT>GET TSTSYS.JSL
EDIT>PRINT
PRINT JOB QUEUED, MUST EXIT TO PRINT
EDIT>END
OS1010 Starting job 00003
OS1000 READY FOR COMMANDS (TIME) hh:mm:ss
OS1020 Job 00003 has completed input phase
OS0020 Resuming OUTPUT
OS1030 Job 00003 has completed printing
OS1000 READY FOR COMMANDS (TIME) hh:mm:ss
EDIT
EDIT>DELETE SYSPDL.JSL
File SYSPDL.JSL deleted
EDIT>END
```

5. Using command files

This chapter contains the following sections:

- Command files
- Commands in command files
- Sample command files.

Command files

This section describes the command files and how to use them.

System, editor, and font editor command files

A command file is a set of commands which has been created using the editor and stored in an executable file, with a .CMD file extension. It can be a system command file, an editor command file, or a font editor command file, depending on which processor is required for execution. A command file, for example, can be used to compile all .JSL files from disk.

Special command file commands (described in this chapter) can be included in a command file to provide interaction with the operator during the file's execution, string substitution, and conditional execution of commands.

Command files are usually .CMD files. If you save a command file with a file extension which is not .CMD, you must specify both the file name and the file extension in order to execute the file. For example, a command file called TEST.CMD can be executed with @TEST or @TEST.CMD, whereas a test file called TEST.TMP can be executed only with @TEST.TMP.

Using system command files

System command files are useful for preserving and executing frequently used sets of system commands. This command file feature can be useful for these functions: system startup and disk initialization, testing, and shift or day initialization. A system command file may contain any of the standard system and command file commands, and may execute an editor command file.

Syntax @*file-name*[.CMD]

Restrictions Some editor command files do not run properly as system command files. Others may run very slowly due to repeated loading of the editor.

Refer to the "Executing command files" section in the "LPS print processing" chapter for additional information.

Operator entries during execution

While the system command file is being processed, any operator key-in is acted upon *between* the commands contained in the command file. (The operator key-in takes effect when the <ENTER> key is pressed.)

Error processing

When a command can be initiated either at system level or from the editor from within an editor command file, an error message is displayed if a processing error occurs. The operator may abort or continue the command file execution. If a processing error occurs when one of these same commands is executed from a system command file, the system does not pause and command file processing continues.

.PAUSE processing

When a .PAUSE command is encountered in a system command file, the following message is displayed:

```
OS1275 Command file suspended
```

Processing stops until the operator keys in the RESUME command or presses the <ENTER> key. Any other key-ins received from the operator during this period function normally. If an @ABORT command is entered at this point, command file processing is discontinued and the command file is aborted.

.WAIT processing

A system command file *must* include a .WAIT command after each print job (whether a START, SAMPLE, or disk print job, such as an FDL or PDL compilation with printing specified). Failure to include the .WAIT command results in the commands following the print job request being refused while the system is printing. When the .WAIT command is encountered, the following message displays:

```
OS1275 Command file suspended
```

Command file processing waits for the completion of the print job. When the LPS becomes idle, the next command file entry is processed. Any other key-ins received from the operator during this period function normally. If an @ABORT command is entered at this point, command file processing is discontinued, and the command file is aborted.

```
*SAMPLE SYSTEM LEVEL COMMAND FILE
*TEST NEW FSL'S AND JSL FOR PRODUCTION
;COMMAND FILE TO PERFORM FDL, PDL AND START TEST JOB
.HOME
;MOUNT TAPE CONTAINING TEST FSL FILES
;AFTER TAPE IS MOUNTED TYPE ENTER
.PAUSE
EDIT @CPYFSL.CMD „Example 1)
TAPE UNLOAD
;FDL IN PROGRESS
FDL NEWFSL,NOP
.WAIT
;MOUNT TAPE CONTAINING TEST JSL FILES
;AFTER TAPE IS MOUNTED TYPE ENTER
.PAUSE
EDIT @CPYJSL.CMD „Example 2)
TAPE UNLOAD
;PDL IN PROGRESS
PDL NEWJDL,NOP
.WAIT
;IF COMPILES WERE SUCCESSFUL
;TYPE ENTER |- IF NOT TYPE @ABORT
.PAUSE
;TEST JOB IS BEING STARTED
;USE TEST TAPE TEST001 WHEN TAPE MOUNT REQUESTED
```

You would input START JOB1,NEWJDL to start offline processing at this point.

```
.WAIT
;COMMAND FILE COMPLETED
.EXIT
```



Note: Do not include online and offline operator commands in a command file.

Using editor command files

Editor command files are useful for preserving and executing frequently used sets of editor commands. An editor command file may contain any of the standard editor and command file commands and may also invoke ACCOUNT commands (refer to the "Tracking system activity" chapter). This allows the use of a command file to add and delete department names in the accounting file.

An editor command file may be initiated at system level by invoking the editor with a command file specified in the form:

```
EDIT @file-name [.CMD]
```

It may also be initiated at the editor command level:

```
EDIT> @file-name [.CMD]
```

Points to note

When a .PAUSE command is encountered in an editor command file, the following message is displayed:

```
Enter return to continue
```

Processing stops until the <ENTER> key is pressed.

An @ABORT command is not allowed in editor processing. If an @ABORT command is attempted, the message FILE DOES NOT EXIST displays and command processing continues.

Example 1

```
*EDITOR COMMAND FILE | CPYFSL.CMD
*COMMAND FILE TO COPY FSL FILES
COPY TAPE NEWFSL.FSL
END
```

Example 2

```
*EDITOR COMMAND FILE | CPYJSL.CMD
*COMMAND FILE TO COPY JSL FILES
COPY TAPE NEWJDL.JSL
END
```

Using font editor command files

Font editor command files are used to preserve and execute sets of font editor commands. For detailed information regarding font editor commands, refer to the "Editing Font Files" chapter in this document. A font editor command file may contain any of the standard font editor and command file commands.

Execution is initiated from the font editor in a manner consistent with editor command files as described in the preceding section.

Verifying command file type

This subsection describes how to verify command file type.

.BCP

Many command files must be executed by a specific processor. If a command file is written for the font editor, it will not function under the text editor. The .BCP command is used to ensure proper operation. This command results in a branch to the specified label only if the command file is invoked by the correct processor. It does not branch if the .BCP command is encountered by incorrect processors.

Syntax .BCP {EDITOR | SYSTEM | FEDIT} *.num*

Parameters Table 5-1 shows the parameter of the .BCP command.

Table 5-1. **Parameter of the .BCP command**

Parameter	Description
<i>.num</i>	A label having a value of 1 to 32767.

Example

```
* THIS COMMAND FILE MUST BE RUN AT THE SYSTEM LEVEL
*
.BCP SYSTEM 20
; THIS COMMAND FILE MUST BE RUN AT THE SYSTEM LEVEL
; PLEASE RESTART IT
```

```
.EXIT }
.20  }
.    } body of system level comma
.    }
.    }
.EXIT }
```

In the preceding example, the .BCP command serves as a sentinel precluding any processor except OCS from executing the system command file.

Commands in command files

This section describes the command file commands and how to use them.

Displaying messages

To display messages or comments for the operator when executing a command file, place a semicolon (;) in the first position of the record. Some utilities such as REALLOCATE require operator key-ins in response to questions asked by the processors. Displayed messages can be used to instruct the operator's response. The responses still have to be entered by the operator and cannot be provided by the command file.

Documenting the command file

A record beginning with an asterisk (*) is ignored. This record type can be used as documentation in the command file.

Summary table of command file commands

Table 5-2 summarizes the command file commands.

Table 5-2. **Command file commands**

Command	Function
.ASK	Alters symbol settings.
.ASKS	Substitutes character string for variable name.
.ASKV	Assigns symbol integer values.
.BCP	Branches if correct processor.
.BEZ	Branches if symbol is zero.
.BF/.BR/.BT/.BX	Specifies branching instructions.
.BNZ	Branches if symbol is not zero.
.CHAIN	Chains editor command files.
.DEC	Decrements a symbol.
.DELAY	Delays for a specified time.
.EXIT	Terminates a command file.
.HOME	Clears LPS display if running in character mode.
.INC	Increments a symbol.
.NOSHOW	Discontinues displaying logical commands.
.PAUSE	Suspends execution of command file.
.SETC	Sets symbol to current logon class.
.SETF	Sets symbol false.
.SETT	Sets symbol true.
.SETV	Sets symbol to a value.
.SHOW	Displays logical commands.
.WAIT	Suspends command file processing until print job completes.

Using the basic command file commands

This subsection describes how to use the basic command file commands.

.CHAIN

The .CHAIN command closes the current command file and begins processing the file specified.

All values and strings are cleared.

Syntax .CHAIN {*file-name* | *file-id*}

Parameters Table 5-3 shows the parameter of the .CHAIN command.

Table 5-3. **Parameter of the .CHAIN command**

Parameter	Description
<i>file-id</i>	Refers to the name of a file, and the file directory in which it resides, in the form <i>file-name.file-type</i> . If no <i>file-type</i> is specified, it defaults to the CMD directory.

PC UI window mode If the LPS is running in PC UI window mode, .CHAIN suspends processing until the operator presses the <ENTER> key. Add a prompt for the operator to press <ENTER> after each .CHAIN command.

.DELAY

The .DELAY command suspends command file processing for the specified period of time.

Syntax .DELAY *n*

Parameters Table 5-4 shows the parameter of the DELAY command.

Table 5-4. **Parameter of the DELAY command**

Parameter	Description
<i>n</i>	The number of seconds to delay further processing. Delay time is limited to five minutes.

.EXIT

The .EXIT command ends command file processing. Ending the command file also discontinues command file processing.

Syntax .EXIT

.HOME

The .HOME command clears the LPS display. The cursor is set to the upper left corner of the display.

Syntax .HOME

.PAUSE

The .PAUSE command temporarily suspends the execution of a command file. A message is displayed and processing stops until the <ENTER> key is pressed, or the @RESUME command is entered. If @ABORT is entered at this point, command file processing is discontinued.

Syntax .PAUSE

.WAIT

When this command is encountered, a message displays and command file processing waits for print job completion. When the LPS becomes idle, the next command file entry is processed. Any other key-ins received from the operator during this period will function normally. If @ABORT is entered at this point, command file processing discontinues.

Syntax .WAIT

Using the logical processing command file commands

Command file logical commands cause conditional execution of the command. Commands are provided to set symbols to a true or false state, substitute a set of characters for a variable name, and assign symbols an integer value. Other commands test the symbols/variables and continue execution at the designated place in the file. The place at which execution is resumed is designated with a label.

Using symbol setting commands

There are eight valid symbols that may be set true, false, or assigned a value. These symbols are the letters A through H, represented by the character *x* in this section. When the character pair %*x* (where *x* is one of the symbols A through H) is found in a record of the command file, the value that was assigned to that symbol is substituted for it.

.SETT <i>x</i>	The symbol <i>x</i> is set to true.
.SETF <i>x</i>	The symbol <i>x</i> is set to false.
.SETC <i>x</i>	The value of the current logon class is assigned to symbol <i>x</i> .
.SETV <i>x=v</i>	The symbol <i>x</i> is set to the value of <i>v</i> , where <i>v</i> is a decimal number in the range 0-255.
.DEC <i>x</i>	One is subtracted from the value of <i>x</i> . If a value of zero is decremented, the next value is 255.
.INC <i>x</i>	One is added to the value of <i>x</i> . If a value of 255 is incremented, the next value is zero.
.ASK <i>x</i> 'question text'	The question text is displayed with ? (Y/N) appended to it. The setting of symbol depends on the response provided by the operator. An entry of Y sets the symbol to true. An entry of N or pressing the <ENTER> key sets the symbol to false. Single quotes (') around the question text are required.
.ASKV <i>x</i> 'question text'	The question text is displayed. The response is expected to be an integer between 0 and 255, and the symbol <i>x</i> is assigned that value. Single quotes (') around the question text are required.

Using string substitution commands

String substitutions may be made with two commands: .ASKV (described previously) and .ASKS.

.ASKS {I|J|K} 'text of message'

Substitutes a string of characters for a symbol during the execution of a command file.

A message displays, and an operator response is required. The response is a string of up to 32 characters which are assigned to the specified symbol. When the character pair %I, %J, or %K is found in a record of the command file, the character string entered by the operator is substituted for it. Single quotes (') around text of message are required.

Example 1

```
ASKS I 'ENTER FILE NAME TO DELETE'
DELETE %I
```

displays when the command file is executed:

```
ENTER FILE NAME TO DELETE
```

In response to this, the operator enters a character string which is substituted for the %I in the DELETE command. The specified file is then deleted.

Example 2

The commands .BT and .BF function on the variables A through H. An unassigned variable is false, and any string input makes it true. For example, the commands:

```
.ASKS I 'ENTER NAME OF FILE TO DELETE'
.BF I 80
DELETE %I
.80
```

delete the file if a name is entered, but the DELETE command is branched around if only the <ENTER> key is pressed.

Using sequence control commands

The commands that control the flow of processing in a command file are as follows where *.num* defines a label. The label is a period followed by a number in the range 1-32767.

<code>.BR num</code>	Unconditional branch to <i>num</i> .
<code>.BT x num</code>	Branch to <i>num</i> if symbol <i>x</i> is TRUE.
<code>.BF x num</code>	Branch to <i>num</i> if symbol <i>x</i> is FALSE.
<code>.BX x .num0,num1,...numn</code>	Test symbol <i>x</i> against the labels <i>num₀</i> , <i>num₁</i> ,... <i>num_n</i> where <i>n</i> is an integer (0-255). If the value of the symbol does not correspond to any label, no branch occurs. The number of labels (<i>num₀</i> to <i>num_n</i>) should correspond to the number of choices available when symbol <i>x</i> was given a value.

Example

Enter your choice:

0) QUIT

1) Option 1

2) Option 2

`.ASKV H ' ; Enter number corresponding to your selection:'`

`.BX H 9999, 20, 100,`

`* QUIT, Option1, Option2,`

The symbol H will have one of three values (0 to 2) which must correspond to the number of labels that the command BX tests H against.

<code>.BEZ x num</code>	Branch to <i>num</i> if symbol <i>x</i> equals zero.
<code>.BNZ x num</code>	Branch to <i>num</i> if symbol <i>x</i> does not equal zero.

Using debugging commands

Logical commands are not displayed by default. For debugging purposes, they may be displayed by using the `.SHOW` command.

<code>.SHOW</code>	All logical commands following this command are displayed. The display may be turned off with the <code>.NOSHOW</code> command.
<code>.NOSHOW</code>	Logical commands following this command are not displayed.

Sample command files

This section contains several sample command files.

Editor command file to delete files

This example illustrates how to delete files.

Example 1 Command file commands are illustrated below. Underlined text represents user-created messages that may be selectively displayed when the command file is executing.

```
;SAMPLE EDITOR COMMAND FILE
.HOME
.ASK A 'ARE YOU LOGGED ON TO LEVEL 2 OR 3'
.BF A 80
.ASK A 'DO YOU WANT TO COPY TEST FILES TO DISK'
.BF A 60
;MOUNT TAPE CONTAINING TEST FILES
;AFTER TAPE IS MOUNTED
.PAUSE
.ASKV B 'ENTER NUMBER OF TEST FILES TO BE COPIED (0
- 3)'
.BX B 60,56,54,52
.52
COPY TAPE LABEL GBAR.FRM
.54
COPY TAPE LABEL ABC.JSL
.56
COPY TAPE LABEL PCHA.TMP
.60
.ASK A 'DO YOU WANT TO DELETE THE TEST FILES FROM
DISK'
.BF A 80
DELETE GBAR.FRM, ABC.JSL, PCHA.TMP
;** FILES HAVE BEEN DELETED **
.70
.ASKS I 'ENTER NAME OF ANOTHER FILE TO DELETE'
.BF I 80
DELETE %I
.BR 70
.80
END
```

Messages of records 7 and 8 above are augmented by the system with the message ENTER RETURN TO CONTINUE, which is displayed when the .PAUSE command (record 9) is executed:

```
MOUNT TAPE CONTAINING TEST FILES
AFTER TAPE IS MOUNTED
ENTER RETURN TO CONTINUE
```

Similarly, the message of record 5 and the message of record 19 are augmented with ? (Y/N). For example, the message of record 5, DO YOU WANT TO COPY TEST FILES TO DISK becomes DO YOU WANT TO COPY TEST FILES TO DISK? (Y/N).

Sample verification of command file type

This example illustrates how to verify command file type.

Example 2 Example of test for correct processor.

```
(Filename = RNEDIT.COMD)
* THIS COMMAND FILE MAY BE RUN BY THE EDITOR OR THE
* SYSTEM
*
.BCP EDITOR 100
*EDIT IS NOT THE PROCESSOR
.BCP SYSTEM 20
;THIS COMMAND FILE MUST BE RUN USING THE EDITOR.
;PLEASE RESTART IT WITH THE EDITOR.
.EXIT
```

```
.20 }
EDIT } "body of command file)
. }
. }
```

```
.EXIT
```

6.

Editing font files

This chapter contains information on using the font editor to run a font editing session.

About the font editor

This section describes the font editor and provides you with font information references.

Reference documentation for fonts

Functional and descriptive font information is available in various customer documents. For example, the *4850 Highlight Color* and *4135 LPS Font User Guide* are collections of font data sheets for the fonts, which comprise the standard font library. Figure 6-2 is an example of a typical two-page font data sheet (specifically applicable to a Xerox-created font file, designated as L05TAB).

Similar information is also available for any font which currently resides on the system disk by keying in the system level SAMPLE command followed by an appropriate file id. Figure 6-3 is an example of a sample printout produced by keying in SAMPLE L05TAB.FNT, where L05TAB.FNT is the complete file id composed of the file name, L05TAB, a delimiter (the period), and the file type, FNT.

Any standard font library font file which is not residing on the system disk may be loaded from the Standard Font Library open-reel tape, floppy disk, or cartridge tape. Files can also be downloaded from a channel-attached host. Additional font-related information and assistance are available from the Xerox Font Center.

Font editor

The font editor utility enables a font-knowledgeable user, with logon class 2 or higher access privileges, to create font files from existing non-licensed and licensed font files source files. You are required to determine what existing source font files need accessing, when they need accessing, and which characters need copying.

Within the context of this chapter, font file information is categorized as being either general in nature and applicable to the entire font file (for example matrix length, file name in ASCII, program version, header-character width tables, pitch, and so on) or specifically and uniquely related to an individual character (for example bit map). As described in this chapter, any existing font file may function as a primary or secondary source font file by virtue of when, how, and why it is selected.

How to use the font editor

Evaluate the general and specific characteristics of all existing font files and decide which will best fulfill the requirements of the new font file. User-created font files differ from source font files primarily in the composition of the character sets. The new font file is either an abridged, enhanced, or hybrid version of one or more source font files.

An existing font file can be used in a font editing session as either:

- The primary source font file which provides general overall font characteristics, plus specified characters
- A secondary source font file, which provides only specified characters added to the primary source font file.

During a font file editing session, the required source font files are specified and accessed. The primary source font file is specified first and then the secondary font files are specified.

The general characteristics of the font file are obtained from either:

- The first selected licensed source font file (when the complement of selected source font files is composed of both licensed and non-licensed or only licensed font files)
- The primary source font file (when all the selected source font files are nonlicensed).

The general characteristics of the new font file are augmented with security information from the first selected licensed source font file.

Characters contained in the new font file are copied from the primary or secondary source font files and may be enumerated as lists of character literals, or octal or hexadecimal values. Each character in the new font file (whether copied from a fixed-pitch or proportional source font file) has an associated character width field of information.

Font files may be created either manually in an interactive mode or automatically as specified within a designated command file.

Both the source and output font files can be up to 1 megabyte in size.

Restriction The font editor cannot edit a double-byte font (for example, kanji font).

Summary table of font editor commands

Table 6-1 lists and describes font editor keyword/commands.

Table 6-1. **Font editor keyword/command summary**

Keyword/command	Function
CHARACTER	One of three keywords (CHARACTER, HEXADECIMAL, or OCTAL) sets the selection mode. CHARACTER signifies that subsequent characters included in the new font file are expressed as character literals.
CLOSE	Explicitly ends access to the current secondary source font file.
END	Explicitly ends font editing session.
FEDIT	Invokes font editor.
HEXADECIMAL	One of three keywords (CHARACTER, HEXADECIMAL, or OCTAL) sets the selection mode. HEXADECIMAL signifies that subsequent characters included in the new font file are expressed as hexadecimal values.
INCLUDE	Precedes the list of characters copied from the currently accessible source font file to the new font file without character code reassignments (refer to RECODE).
INPUT	Specifies the primary source font file.
OCTAL	One of three keywords (CHARACTER, HEXADECIMAL, or OCTAL) sets the selection mode. OCTAL signifies that subsequent characters included in the new font file are expressed as octal values.
OPEN	Specifies secondary source font file.
OUTPUT	Stores the current contents of the working storage as a new and permanent font file.
RECODE	Specifies the list of characters copied from the source font file to the new font file with new character code assignments (refer to INCLUDE).
REORDER	Changes accessing sequence of characters within a specified font file as required to achieve compatibility with Interpress standards.
SUBSTITUTE	Allows the operator to replace any blank with another character.
#	In addition to being the symbol for the pound sign, this character also signifies shifting from uppercase to lowercase, or vice versa, alternately. This use is operative only in character mode.



Note: 1. In addition to the above listed keywords, the space character, hyphen, and comma are recognized as delimiters by the font editor.

Note: 2. Whenever the font editor has been invoked and is ready to accept a keyword command, the font editor prompt FEDIT> is displayed.

Running a font editing session

This section illustrates how to run a font editing session.

Messages during the session

Messages are displayed during a font editing session when appropriate. These messages may:

- Advise the user that the results of a previous editing session may have to be saved before the current font editing session is started
- Remind the user of possible need to save the results of the current font editing session before terminating the session
- Indicate a condition of possible incompatibility (which can occur when selecting characters from several existing font files and adding to a new font file)
- Indicate the existence of another file with the same name as the one which is about to be stored
- Advise the user that a character has been specified more than once during the same font editing session
- Inform the user that the new font has exceeded the maximum font file size of one megabit.

Starting a font editor session

To accommodate the various types of font editing sessions, the font editor is invoked by the FEDIT system level command in one of three ways.

Syntax FEDIT
FEDIT *file-name*[.FNT]
FEDIT *@file-name*[.CMD]

The first command invokes the font editor so that it can process keyword commands. After the font editor is invoked the font editor displays the prompt FEDIT>.

The second command form initiates a manual, interactive mode font editing session. In this form, the font editor is invoked, and the primary source font file is specified simultaneously by the same command. After the font editor has selected the specified font file and is ready to accept the next keyword command, FEDIT> displays.

The third command form specifies a command file for a preprogrammed font editing session. If the font editor has performed all commands within the .CMD file and is not terminated (such as no END command has been performed), the font editor displays the prompt FEDIT>.

Reordering a font file

This subsection describes how to reorder a font file.

REORDER and FEDIT

Occasionally, the bit maps of a user-created font file must be reordered in ascending sequence to be compatible with Interpress standards. Either of two commands can be used to accomplish this:

Syntax FEDIT
REORDER *file-name*[.FNT]

Parameters Table 6-2 shows the parameters of the FEDIT and REORDER commands.

Table 6-2. **Parameters of the FEDIT and REORDER commands**

Parameter	Description
FEDIT	The keyword command for invoking the font editor. It may be spelled out in full or abbreviated to the first three letters.
REORDER	A keyword command which reorders the designated font file into ascending sequence. This function is performed automatically without further input. Upon completion, the designated font file is Interpress compatible. The keyword may be spelled out in full or abbreviated to the first three characters.
<i>file-name</i> [.FNT]	The <i>file-id</i> of an existing font file and its bit maps. The <i>file-type</i> designator .FNT is the default.

Upon completion of the reordering process for the specified font file, the font editor remains invoked and is available to perform another REORDER command, or a manual font editing session in which the primary source font file is specified with an INPUT command. The REORDER command may not be issued during a font editing session once a primary source font file has been specified.

To exit the font editor (or to end a font editing session), enter the END command.

Specifying the primary source font file

This subsection describes how to specify the primary source font file.

FEDIT

An existing font file is generally selected as the primary source font file because it contains one or more characters required by the new font file. It also has the required general and overall font characteristics desired for the new font. The primary source font file may be specified with either (1) the FEDIT command or (2) an INPUT command (for command initiation, refer to the "INPUT" section of this chapter).

The following command format invokes the font editor and selects the primary source font file simultaneously:

Syntax FEDIT *file-name* [.FNT]

Parameters Table 6-3 shows the parameters of the FEDIT command.

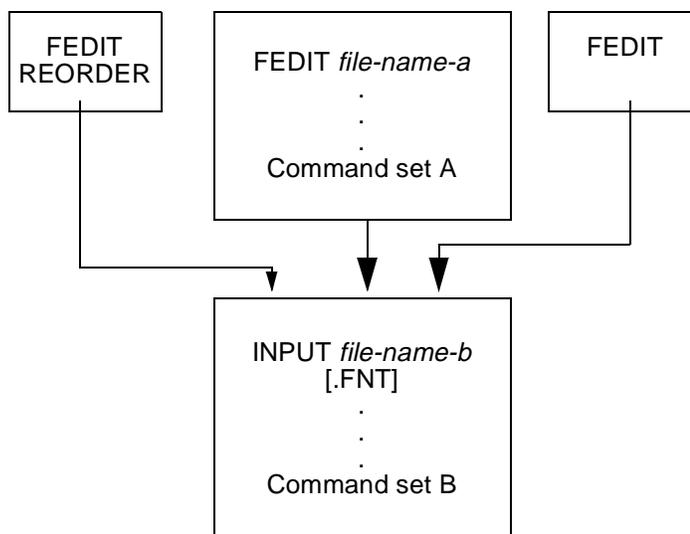
Table 6-3. **Parameters of the FEDIT command**

Parameter	Description
FEDIT	The keyword command for invoking the font editor. It may be spelled out in full or abbreviated to the first three letters.
<i>file-name</i> [.FNT]	The <i>file-id</i> of an existing font file which is the primary source font file. The <i>file-type</i> designator .FNT is optional and assumed if omitted. If the specified font file does not exist, an error message is displayed.

INPUT

If the font editor has been previously invoked, the primary source font file is specified with an INPUT command. As illustrated in figure 6-1, the font editor may have been previously invoked to perform a reordering process or to create a font file.

Figure 6-1. **INPUT command with font editor invoked**



Parameters Table 6-4 shows the parameters of the INPUT command.

Table 6-4. **Parameters of the INPUT command**

Parameter	Description
INPUT	The keyword command which selects the designated font file as the primary source font file. The keyword command may be abbreviated to the first three characters.
<i>file-name-b</i> [.FNT]	The <i>file-id</i> of a source font file which is selected as the new primary source font file. The <i>file-type</i> designator, .FNT, is assumed if omitted.
<i>command set B</i>	Represents a sequence of commands which fully defines the creation and disposition of the new font file using <i>file-name-b</i> as the primary source font file.
FEDIT REORDER	Commands which defined a previous font editing session.
FEDIT <i>file-name-a</i> and <i>command set A</i>	Represent a sequence of commands which define a previous font editing session.
FEDIT	Represents the case where only the font editor was invoked.

The INPUT command normally opens a new primary source font file. Appropriate general overall font characteristics information is copied from the current or new primary source font file into the new font file.

Restrictions The INPUT command should not be issued until any previously modified font file has been saved with an OUTPUT command. If this has not been done, an error message is displayed. The user has the option of entering an appropriate OUTPUT command or proceeding with the new INPUT command and discarding the previously modified font file.

Selecting characters using the correct mode

The character selection mode keyword commands are preparatory in nature and work in conjunction with subsequent INCLUDE and RECODE commands. The character selection mode may be specified as HEXADECIMAL, OCTAL, or CHARACTER. When the Font editor is invoked, the HEXADECIMAL mode is assumed by default. The assumed or specified character selection mode prevails throughout the current font editing session until respecified.

In the HEXADECIMAL mode, the characters copied into the new font file are specified by the character code assignments expressed as hexadecimal values.

In the OCTAL mode, the characters copied into the new font file are specified by the character code assignments expressed as octal values.

In the CHARACTER mode, the characters copied into the new font file are specified as character literals.

When the terminal is used as the input device, both uppercase and lowercase characters are available with the shift key.

The character mode is appropriate when all the specified characters can be entered from the keyboard. The HEXADECIMAL or OCTAL modes may be used to specify any characters which cannot be entered directly on the keyboard, including delimiters and non-graphic or nonprinting characters or symbols (for example the blank, comma, and hyphen characters are always recognized as delimiters within the font editor). To be included within the new font file, they may be specified by using the hexadecimal or octal character code values.

Specifying new font file characters

All characters included within a new font file are specified as parameters of either the INCLUDE or RECODE commands. The bit maps of the specified characters are copied from the selected source font file into the new font file being created. If the same character is specified more than once during the same font editing session, a warning is issued and the specified character will have the attributes copied from the font file when first specified.

INCLUDE

The INCLUDE command specifies which characters will have the same character code assignments in the new font file as in the source font file. An INCLUDE command is effective on the most recently selected source font file, providing a primary source font file has been specified for INPUT. More than one INCLUDE command may be issued for the same selected source font file. The parameters of an INCLUDE command are expressed as hexadecimal, octal, or character values as defined by the Character Selection Mode.

Example FEDIT
INPUT filename[.FNT]
CHARACTER
INCLUDE A-Z, #A-Z

In the above example, the FEDIT command invokes the font editor; and the INPUT file name[.FNT] command selects a designated font file as the primary source font file for the font editing session.

Parameters Table 6-5 shows the parameters of the INCLUDE command.

Table 6-5. **Parameters of the INCLUDE command**

Parameter	Description
CHARACTER	Specifies a selection mode of character literals.
INCLUDE	A keyword command that precedes the lists of characters, specified as literals and delimited by commas and hyphens, copied from the source font file to the new font file without changing the character code assignments; where, A-Z is an abbreviated listing of uppercase characters comprising the alphabet having contiguous character code assignments, starting with A and ending with Z. #A-Z or a-z is an abbreviated listing of lowercase characters comprising the alphabet having contiguous character code assignments, starting with a and ending with z. The preceding pound sign in this example functions as an uppercase to lowercase toggle.

SUBSTITUTE

When a font is created for characters that are not included in the new font, there will be a blank. The SUBSTITUTE command allows the operator to replace any blank with another character.

Syntax SUBSTITUTE *m n*

Parameters Table 6-6 shows the parameters of the SUBSTITUTE command.

Table 6-6. **SUBSTITUTE command parameters**

Parameter	Description
<i>m</i>	Copies the character represented by <i>m</i> (a hexadecimal or octal code, or a character literal) from the INPUT font file.
<i>n</i>	Becomes the character represented by <i>m</i> (a hexadecimal or octal code, or a character literal) in the new or modified font file.

RECODE

Use this command to copy characters and assign hexadecimal or octal codes in the new font file which are different from their codes in the source font file.

Syntax RECODE [$n = m$ | $n-m = x-y$ | $n = x-y$]

Parameters Table 6-7 shows the parameters of the RECODE command.

Table 6-7. **Parameters of the RECODE command**

Parameter	Description
$n = m$	Copies the character represented by m (a hexadecimal or octal code, or the character literal) to the character in the new font file, which is represented by n .
$n-m = x-y$	Copies the characters represented by the hexadecimal or octal codes (or character literals) in the range $x-y$ to characters in the new font file which are represented by the range $n-m$.
$n = x-y$	Copies the characters represented by the hexadecimal or octal codes (or literal characters) in the range $x-y$ from the source file to characters in the new font file which are represented by the hexadecimal or octal codes (or character literals) beginning with n .

Example

```
OCTAL
RECODE 44= 232
HEXADECIMAL
RECODE 3A=4146
```

In the above example, the OCTAL keyword command specifies that the character selection mode is octal. This means characters specified by subsequent RECODE command must be expressed as octal values representing character code assignments. As shown, a character within the source font file having a character code assignment equal to octal 232 is copied into the new font file and has a character code assignment of octal 44.

In a similar manner, the HEXADECIMAL keyword command sets the character selection mode to hexadecimal. Then a string of contiguous characters having character code assignments of hexadecimal 41 through 46 in the source file are copied as contiguous characters into the new font having character code assignments of hexadecimal 3A through 3F.

Using a secondary source font file

You may select fonts as secondary source font files in order to add characters to the new font file. If no secondary source font file is required, (such as the user-created font file is complete after the general information and specified characters are copied from the primary source font file), the new font file should be saved by issuing an OUTPUT command described later in this chapter.

OPEN

If one or more secondary font files are necessary to furnish additional characters to the new font file, each secondary font file must be explicitly specified with an OPEN command.

Syntax OPEN *file-name* [.FNT]

Parameters Table 6-8 shows the parameters of the OPEN command.

Table 6-8. **Parameters of the OPEN command**

Parameter	Description
OPEN	A keyword command which opens the specified font file. The keyword may be abbreviated. An OPEN command is valid only after a primary source font has been specified by the INPUT command.
<i>file-name</i> [.FNT]	The <i>file-id</i> or name of a font file which is opened and accessed as a secondary source font file. The <i>file type</i> , .FNT, is assumed if omitted.

While the specified secondary source font file is open:

- The primary source font file is unavailable until the secondary source font is closed (refer to the "CLOSE" section in this chapter).
- The character selection mode may be HEXADECIMAL, OCTAL, or CHARACTER.
- The INCLUDE and RECODE commands may be specified, as required, to enumerate the characters to be copied from the secondary source font file with or without being recoded.

The system monitors the selection of secondary source font files and issues an appropriate warning message whenever a condition of possible font incompatibility is detected. It is assumed that a font-knowledgeable user understands the full consequence of any error/warning message before continuing.

If a secondary source font file is the first licensed font selected for this new font file, the general overall characteristics initially copied from the primary source font file are overwritten with general overall characteristics and augmented with security information from the licensed font file.

A selected secondary source font file may be closed explicitly or implicitly.

CLOSE

Use this command to close a secondary source font file.

Syntax CLOSE

Parameters Table 6-9 shows the parameter of the CLOSE command.

Table 6-9. **Parameters of the CLOSE command**

Parameter	Description
CLOSE	Closes the currently open source font file. The primary source font file reassumes a foreground role after the secondary source font file is closed.

Closing font file without using CLOSE

An open secondary source font file may be implicitly closed (such as closed without using the CLOSE command) in several ways.

- By issuing an OPEN command to select another existing font file as a secondary source font file to obtain additional characters for the new font file
- By issuing an INPUT command to initiate a font editing session for another new font file
- By issuing an END command to terminate all font editing operations
- By issuing an OUTPUT command.

Saving font files you create

This subsection describes how to save font files you create.

OUTPUT

Use this command to store as a permanent font file any user-created font file.

Syntax OUTPUT *file-name* [.FNT]

Parameters Table 6-10 shows the parameters of the OUTPUT command.

Table 6-10. **Parameters of the OUTPUT command**

Parameter	Description
OUTPUT	A keyword command which stores the temporary font file being created within the working storage by the font editor as a permanent file having the designated name. The command may be abbreviated to three letters.
<i>file-name</i> [.FNT]	The designated name of the newly created file. The file type .FNT, is assumed if omitted.

Restrictions The OUTPUT command is valid only if characters have been copied from a source font file (with the INCLUDE and RECODE commands).

If an existing font file has the same name as that specified in an OUTPUT command, the system issues a message requesting confirmation that the existing font file should be overwritten.

If Y is entered, the OUTPUT command is executed. This occurs when the user is working on a long font editing session and wishes to save the intermediate results. The latest intermediate results supersede the preceding intermediate results.

If N is entered, the OUTPUT command is ignored. This response helps to prevent the inadvertent loss of an existing font file. The current newly created font file may be saved by issuing another OUTPUT command and using a unique file name.

The contents of working storage are not affected by the OUTPUT command. Thus, a series of new font files where the first file is a subset of a latter file may be created. This is done by issuing an OUTPUT command at intermediate points within a font editing session using the same primary source font file and any number of secondary source font files.

Ending the font editing session

This subsection describes how to end a font editing session.

END

Use this command to exit the font editor and end a font editing session. Execution of an END command is contingent upon the status of the latest font file. If all required information for the latest font file has been copied and saved with an OUTPUT command, the END command is executed immediately. If changes to the latest font file have not been saved, the system issues an error message when an END command is encountered. The user must indicate whether the new font file is to be saved or lost.

Syntax END

7. LPS print processing

This chapter contains the following sections:

- Overview
- Laser printing systems (LPS)
- Types of LPS processing
- Online printing systems
- Downloading files from the host to the LPS.

Overview

Xerox laser printing systems (LPS) enable host mainframe computers, workstations, and networked personal computers to produce publications and other documents, incorporating graphics, forms, logos, signatures, and a variety of fonts.

LPS benefits

A variety of print speeds, print resolutions, and features are provided by the LPS product line. There are laser printing systems designed for use in medium and large data processing environments needing high-quality computer reports, financial documents, and other forms-intensive output. Management information systems and data center processing environments benefit from high-volume laser printing systems in which built-in intelligence eliminates the need for the host computer to store and manage fonts, forms, and their document resources.

The high-resolution printing of Xerox laser printing systems makes them especially suitable for use in data centers producing external documents in which print quality is particularly important.

Laser printing systems (LPS)

This section describes the Laser printing systems (LPS) and their components.

LPS component types

Laser printing systems (LPS) have two components.

- Hardware** Refers to all the physical components of the LPS. Examples are the 1/4-inch cartridge tape drive, the PC UI, the feeder/stacker modules, and the physical subsystems of the system controller and the printer. Refer to the *Xerox DocuPrint 180 LPS Product Reference* and to the *Xerox DocuPrint 180 LPS Operator Guide* for detailed descriptions of the hardware components of your LPS.
- Software** Refers to all coded instructions (programs) which are executed by the LPS. Some programs interface with LPS hardware, some with LPS firmware, and some with other software. Examples are the LPS operating system, the PDL compiler, and the output task.

LPS hardware components

This section describes the hardware components.

Advanced imaging subsystem (AIS)

The Advanced Image Subsystem or AIS is made up of three PWBs. These are the EAIPX PWB, EPBIX PWB and EGIFM PWB. The function of each of these PWBs is described briefly here.

Advanced image processor The Enhanced Advanced Image Processor (EAIPX) PWB is the controller of the set. It contains a processor, an advanced graphic display controller chip, an integrated decompressor chip, and interface circuitry for the other two PWBs of the set. Text and metric memory also reside on the EAIPX PWB.

Graphics, ink and font memory The Enhanced Graphics, Ink, Font Memory (EGIFM) PWB is the memory PWB used to store graphic data and font bit maps. The PWB supports memory configurations of 64MB and 256MB of graphic and font memory.

Graphic and font memory stores compressed font bit map and graphic image data which is then sent to, and processed by, the EAIPX PWB. The EGIFM PWB has a unibus interface so that the LPS software can load it directly.

Page buffer and IOT interface The Enhanced Page Buffer & IOT Interface (EPBIX) PWB contains the video buffers and IOT interface logic. The video page buffer capacity is 64MB allowing it to hold a bit map for a 14.33- by 17-inch page for assured printing.

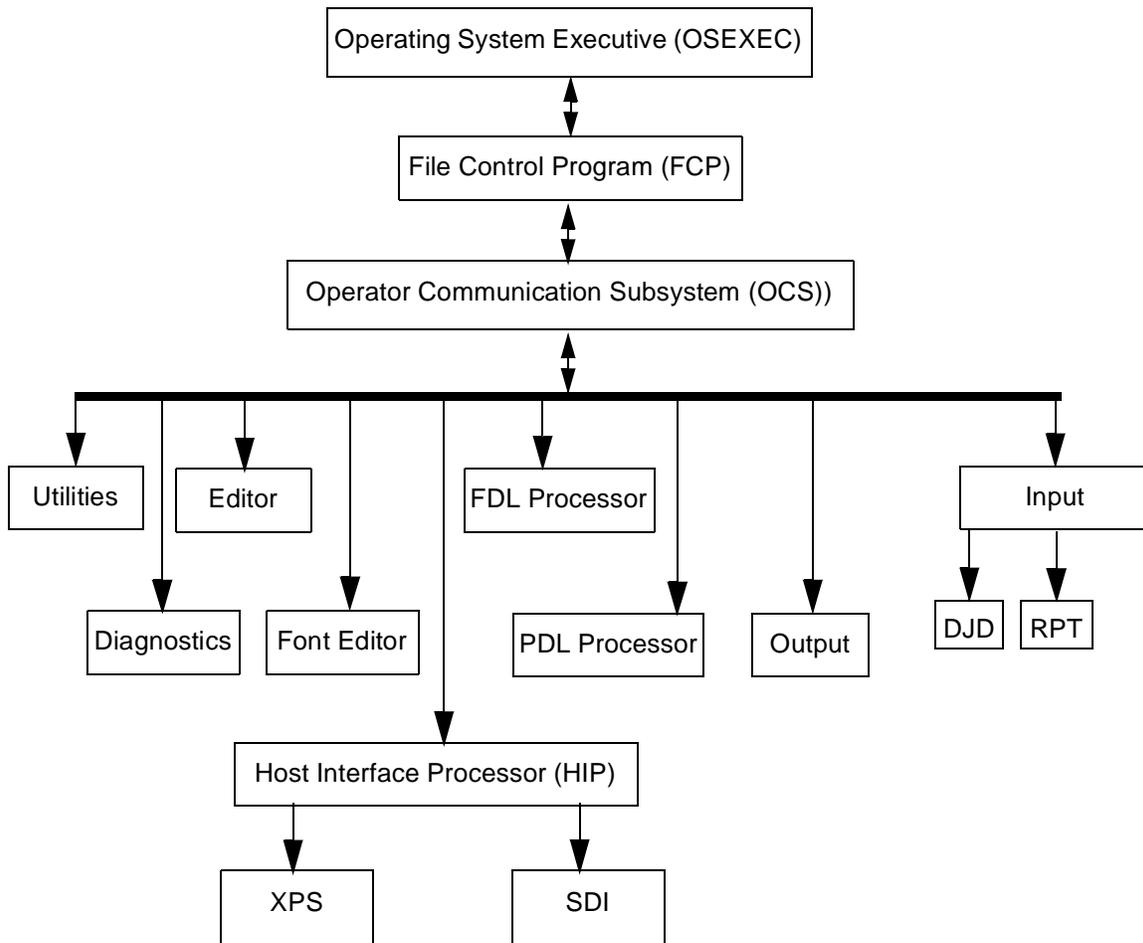
LPS software components

The LPS operating system software (OSS), or system controller, runs both background and foreground processing. All Xerox LPS systems are run by basically the same software. The OSS is delivered for installation and system generation on open-reel tapes, 1/2-inch or 1/4-inch cartridges. The OSS tasks are:

- Operating system executive (OSEXEC) task
- File control program (FCP)
- Operator communication subsystem (OCS)
- Diagnostic tasks
- System utilities
- Editor task
- Font editor task
- Input task
- Dynamic job descriptor (DJD) task
- Report processing task (RPT)
- Output task
- Print description language (PDL) compiler
- Forms description language (FDL) compiler
- Host interface processor (HIP) task.

Figure 7-1 shows, in their hierarchical relationships, the major utilities or tasks run by the system controller.

Figure 7-1. **Major LPS processing tasks**



Operating system executive task

An operating system task that always runs. It interfaces with the LPS hardware and logs hardware errors. In addition, it queues devices, manages resources, establishes priority for software tasks, and schedules processing.

File control program (FCP)

A file control or file management subsystem task that manages disk resources. It manages and allocates all disk space, creates all disk files, and accesses disk files.

Operator communication subsystem (OCS)

Acts as an interface between the operator and software tasks. OCS receives input from the operator and displays messages to the operator. OCS also interfaces between system tasks.

Diagnostic tasks

The operating system diagnostic software (OSDS) runs either under the OSEXEC task or offline. It controls the processing for the PROBLEM command. Diagnostic support is also provided to your service technician by the stand-alone field engineering software (SAFES).

System utilities	The LPS has many utility programs which run under the OSEXEC task. They provide maintenance services. Utilities, for example: back up the system (DSR); track the unusable sectors on disks (FCHECK); assign a maximum size to the working print file (REALLOCATE); eliminate wasted space on the disks (COMPRESS); make used space available again (PURGE).
Editor task	Creates and modifies disk files. It uses a temporary work file and stores the contents of the work file permanently on disk when you save the file. It sends files to print; creates .FSL and .JSL files; directs .CMD files to execute; and manipulates temporary (.TMP), miscellaneous (.MSC), patch (.PCH), and data (.DAT) files. Refer to the "Editing Source Files" chapter of this manual for more information.
Font editor task	Creates and modifies font files. It uses a temporary work file and stores the contents of the work file permanently on disk when you save the file. Refer to the "Editing Font Files" chapter of this manual for more information.
Input task	Creates a job control block based on the JDL, handles job messages to the operator, reads in print job data, and unpacks and converts the data. Then it selects and deletes blocks or records; records any special processing instructions (for page offsets, DJDE records, CMEs); and records the disk addresses of the font, form, and image files to be used for the job. The result is a page buffer and a page log for each page to be printed. The page buffer consists of the variable data and print instructions for the page. The page log consists of tracking information used in processing the page.
Dynamic job descriptor (DJD) task	Compiles the dynamic job descriptor entries (DJDE) in the job stream. (The DJDEs give the printer instructions for printing based on the data in the job stream.)
Report processing task (RPT)	Works in connection with the input task.
Output task	Uses the page logs written by the input task to load fonts and variable data into the AIS. It takes paper from the indicated tray and manages report integrity with multiple checks. It sends the page buffer information to the AIS and coordinates the activity of the AIS with the printer. In addition, output manages delivery of the printed pages to the correct bins, and performs page recovery if necessary.
Print description language (PDL) compiler	Loads the print description language (PDL) compiler and looks in the JSL library for the file to be compiled. Then it analyzes the .JSL statements and creates the object module (the .JDL file).
Forms description language (FDL) compiler	Loads the forms description language (FDL) compiler and looks in the FSL library for the file to be compiled. Then it analyzes the .FSL statements and creates the object module (the .FRM file).

Three LPS user roles

Although this section lists specific tasks for the system administrator, the programmer, and the operator, in practice the assignment of these duties varies widely from site to site. For example: your site system administrator may also be a programmer; your lead operator may be responsible for installing software versions; your programmer may take care of system backups and run print jobs.

System administrator role

As system administrator, you ensure that the LPS hardware and software are working properly. You:

- Perform the system generation
- Install new software versions and apply software patches
- Decide and enforce logon and command privileges
- Decide and enforce usage standards and procedures
- Set up and maintain system backup schedules and procedures
- Set up and maintain disk management procedures.

Programmer role

As programmer, you decide how each page of your printed document looks. You decide:

- The stock to be used for each page
- The trays to be used to feed the stocks to the printer
- The page layout
- The use of forms, logos, signatures, and images
- The placement, font, and size for fixed and variable text.

As the programmer, you also decide how to update existing documents.

If your document requires forms, you code programs in forms description language (FDL). Each .FSL file is compiled into an .FRM file which tells the printer how to build the print image of the form.

You then code programs in print description language (PDL). (Your .JSL files may call .FRM files.)

Then you invoke the PDL compiler to compile the .JSL file into a .JDL file which tells the printer:

- What variable and fixed data to use
- Which fonts, forms, images, and logos to use
- How to feed stock for the print job
- How to build the print image for each page
- How many copies to print.

You may also use host-resident software to create PostScript or HP PCL files which your LPS can print.

Operator role

As operator, you monitor and maintain the printer and manage the print jobs run on the LPS. You:

- Power on the LPS and restart the LPS after power loss or rollover
- Add dry ink and lubricant, and replace waste containers
- Clean the tape, floppy, and cartridge drives
- Load and unload tapes, and control print job input from tape
- Control print job input from online sources with HIP
- Sequence, start, stop, continue, sample, and abort print jobs
- Load and unload stock as needed by the print jobs
- Read the meter and report the readings
- Report LPS problems to your LPS service representative.

Key terms in job processing

You must understand three key terms in job processing.

- Input processing** Refers to the software tasks and firmware functions which read the variable data and job instructions and prepare page-by-page print information for output processing. The OSEEXEC task, OCS, and the input task (defined in the “LPS software components” section) play major roles in input processing.
- Output processing** Refers to the software tasks and firmware functions which convert print information for a page into a video image of the page to be printed, send the video data to the printer, print the video data, and deliver the printed page to a bin. The AIS, the output task, and OCS (defined in the “LPS software components” section) play major roles in output processing.
- Page recovery** Monitoring and page recreation service performed by the output task during output processing. The output task tracks each page of a print job to verify that the page is successfully printed and delivered to a bin. If the page data is corrupted, incompletely printed, or misaligned, or if the page is caught in a paper jam as it travels to the output bin, the output task recreates the video image and resends the page to be printed and delivered.

Job flow overview

This section describes the job processing flow.

Input sources

The input source for an LPS print job can be open-reel tape, a cartridge tape drive (CTS), an online channel from a host, a Dynamic Disk Interface (DDI) channel, or a Xerox Print Service (XPS) Manager channel.

Job flow

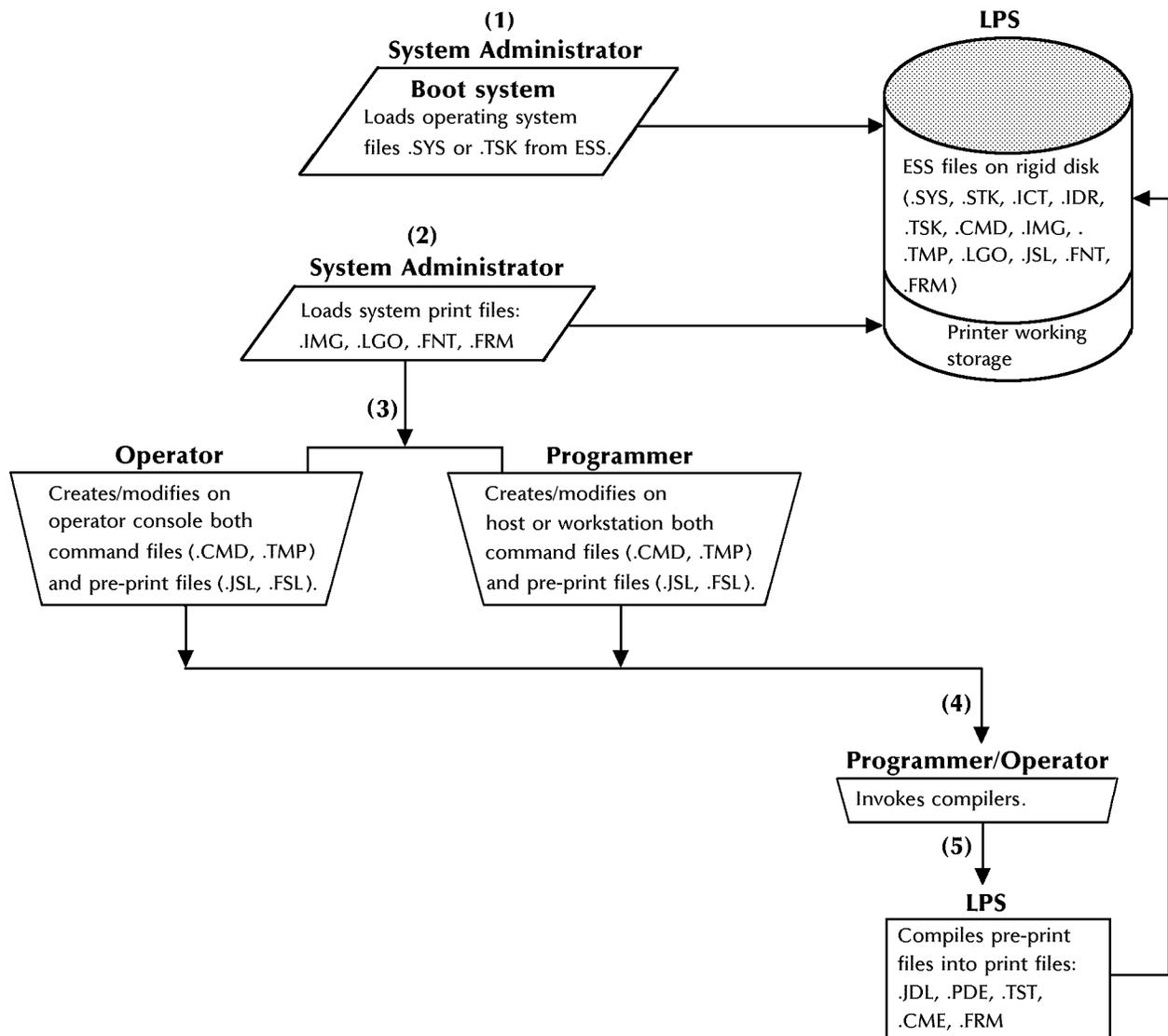
Job flow is the processing directed by the system controller which converts raw data into video format for the printer. Job flow consists of input and output processing.

Preparing for a print job

Before a print job runs, the system administrator, programmer, operator, and LPS work together to prepare for this process.

Figure 7-2 shows the print job process.

Figure 7-2. **Preparing for print job processing**



- 1 **System Administrator**
- 2 **System Administrator**
- 3 **Operator (or Programmer)**
- 4 **Programmer (or Operator)**
- 5 **LPS**

System Administrator The system administrator loads fonts, logos, graphics, and signatures to the printer from tape, floppy, or the host.

System Administrator The system administrator determines the fonts, forms, and graphics system parameters needed by the job.

Operator The operator resets the system parameters to match the fonts, forms, and graphic requirements of the job.

Programmer (or Operator)	<p>The programmer (or operator) creates the .JSL and .FSL source files.</p> <p>The programmer (or operator) invokes the FDL compiler to compile the .FSL files for the print job.</p> <p>The programmer (or operator) invokes the PDL compiler to compile the .JSL files, and the global .CME, .TST, and .PDE files for the print job.</p>
LPS	<p>The LPS completes any foreground processing and waits in IDLE status for the operator to enter the START command specifying the .JDE and .JDL files for the print job.</p>

Job flow steps

The steps required to process a print job are summarized in figure 7-3. Each step is outlined below.

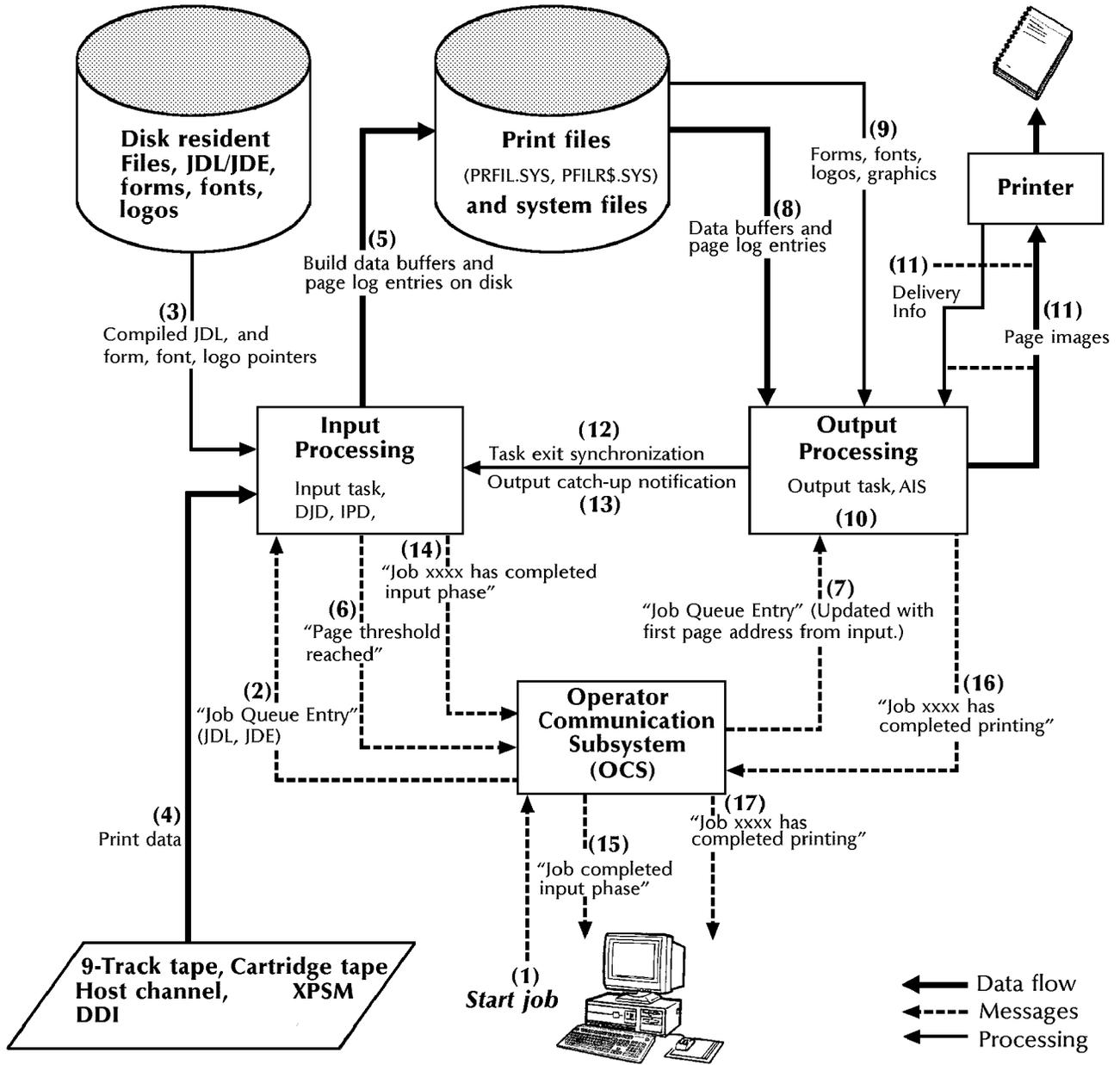
OCS processing	<p>The OCS processing steps are:</p> <p>Step 1. The START command identifies the JDE/JDL created by the programmer for the job. The OCS task reads the START command and notifies the OSEXEC task which loads the output, input, DJD, RPT, and IPD tasks.</p> <p>Step 2. Next, the OCS task writes a job queue entry (JQE) for the job in the job queue buffer. The JQE includes information from the start command, for example the JDE and .JDL names, and the number of copies.</p>
Input processing	<p>The input processing steps are:</p> <p>Step 1. The input task first initializes the job queue buffer from the start JDE/JDL information by recording job characteristics such as tape device and online status.</p> <p>Step 2. Then the input task finds the disk addresses of the compiled .JDL, form, font, and logo and graphic files to be used in printing the variable data.</p> <p>Step 3. Next, the input task reads the variable data.</p> <p>Step 4. Then the input task formats the variable data and writes it to the print file PRFIL1.SYS. For each page in the print file, input task creates a page log entry containing all the necessary disk addresses for all necessary resources. Recovery information is stored in the recovery file, PFILR\$.SYS.</p> <p>Step 5. When the input task processes all the data or reaches page threshold, it messages the OCS task that the output task can begin.</p>
OCS processing	<p>The OCS task sends a message to output indicating that output processing can start. This message includes the address in the PRFIL1.SYS file of the information for the first page to be printed.</p>

Output processing The output processing steps are:

- Step 1.** The output task reads the starting page log entry and builds an entry in its own page tracking buffer. The page tracking buffer contains the information output uses for page creation (text, data, form, and font addresses, counts, and flags for options such as simplex or duplex, which feeder or destination to use).
- Step 2.** Now the output task copies portions of the font, form, graphic, and logo files from disk to the AIS.
- Step 3.** Next, the output task instructs the AIS to build a video representation of the page. The AIS stores the resultant video image into the page buffer.
- Step 4.** Then the output task instructs the AIS to send the page image to the printer. The AIS sends the image, and the output task uses the page tracking buffer to monitor and record the printed page's delivery to the output destination.
- Step 5.** The output task identifies each page successfully printed and delivered. The input task can then reuse that page's space in the print file to process an incoming page. The output task also identifies each page unsuccessfully printed or delivered. The output task uses the page information stored in the PRFIL1.SYS to repeat the steps which recreate the page image, send it to the printer, and monitor its delivery to the bin. Then the output task releases the space for that page in the PRFIL1.SYS file.
- Step 6.** When the output task has verified the successful printing and delivery of all pages prepared for it by the input task, the output task signals the input task that output processing is waiting for more formatted page and page log buffers.
- Step 7.** When the input task reaches the end of file marker or end of job criteria on the input data, the input task messages the OCS task that input processing of the print job is complete.
- Step 8.** The OCS task then displays the following message to the operator.
`Job xxxx has completed input phase.`
- Step 9.** When the output task finishes printing, it messages the OCS task that the job has completed printing.
- Step 10.** The OCS task then displays the following message to the operator.
`Job xxxx has completed printing.`

Figure 7-3 summarizes the job processing flow steps.

Figure 7-3. Overview of job processing flow



Input processing

Input data is processed and written to the system disks for subsequent printing. During processing, the following functions are performed:

- Block selection/deletion
- Extraction of records within blocks
- Record selection/deletion
- Detection of user-defined offset points
- Detection of report delimiter records
- Detection of dynamic job descriptor entry (DJDE) records
- Detection of user-specified font usage requests
- Character translation
- Printer carriage control processing
- Replacement of input text with prespecified static text
- Image placement
- Initiation of automatic recovery procedures when required
- Page log write.

Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for additional information.

Block selection/deletion	You may specify that certain data blocks of input data are to be deleted for processing purposes. These options are specified through the use of the BSELECT or BDELETE statement. If no option is specified, all data blocks are selected for processing.
Record extraction	Record extraction is performed according to the structure of the input data. In the case of data input from tape, blocks may be variable or fixed in length, and records within blocks may be of fixed, variable, or undefined length. Records may be separated by constant delimiters in which case a search process is required to locate records within blocks.
Record selection/deletion	You may specify that only certain data records are to be processed for printing. Alternatively, you may specify that certain data records are to be deleted from the printed report. These options are specified through the use of the RSELECT or RDELETE statement for offline only. If no option is specified, all data records are selected for printing. All selected data records are moved from the input buffers to disk blocking buffers in preparation for writing to disk.
Detection of report delimiter records	You may specify that the end of a report is to occur upon detection of certain data within an input record. This is done through the use of the RSTACK (online, offline) and BANNER (online only) statements. Logically separated reports are physically offset from one another in the output trays.

Detection of dynamic job descriptor entry (DJDE) records

DJDE information is processed from the input data stream and modifies current job characteristics. This allows you to tailor the printing environment according to the particular needs of a report. DJDE information is applied to control the printing of data occurring after the DJDE information. DJDE records may occur within report delimiter records. In this case, the DJDE information applies to the next report.

Detection of user-specified font usage requests

You may specify a byte within the data record to indicate the use of a specific font in printing the data contained in each record. The specified byte is an index into an ordered list of fonts. You may intermix fonts within a print line by specifying different fonts within different input data records, each of which contains a printer carriage control to inhibit normal upspacing.

Character translation

As selected data records are moved from the input buffers to the disk blocking buffers, character translation is performed as necessary. Specify the input code with the VOLUME statement's CODE command.

Printer carriage control processing

You can specify a printer carriage control (PCC) type according to the type of host system. Each PCC value specified corresponds to a particular action to be taken (for example space or skip before printing, print or not, and space or skip after printing). Channels are associated with page print line positions. Spacing and skipping actions are processed by associating the appropriate laser scan line or dot position with the beginning of each data line to be printed. The starting print line position that is generated is part of the control information associated with each print line and directs the character dispatcher in the printing of the line.

Copy modification entry

You may specify a copy modification entry (CME) command to replace or highlight selected portions of page data with prespecified static data, to specify character-to-character font changes, and to replace copy-sensitive data offline. Different CMEs may be associated with different copies of a report.

Image placement

You may include compressed graphic images or image callouts within the text stream. Up to 16 images of data can be positioned on the page. Refer to the *Xerox DocuPrint 180 LPS Forms Creation Guide* and *Xerox DocuPrint 180 LPS PDL Reference* for additional information.

Automatic recovery

If a printing or page delivery problem occurs, the page image is sent again to output processing.

Page log write

As the input data is processed and written for subsequent printing, the input processor also makes entries into a page log which describes how the page is to be printed. The page log entries are subsequently read by the output processor.

Output processing

Output data is read from the system disks and dispatched to the printer a page at a time with the image generator. During processing, the following functions are performed:

- Page log(s) read
- Font(s) loaded into AIS bitmap memory
- Font specification(s) loaded into AIS text and metric memory
- Form(s) loaded into text and metric memory
- Image(s) loaded into AIS bitmap memory
- Paper feeding control
- Report integrity control.

Page logs read

The page log is read into memory to provide the output task with the font, form, and graphic image information required for imaging the page, and the disk address of the imaged page. It also provides a look-ahead capability for determining what new fonts or forms are required for the next page.

Fonts loaded into AIS memory

Fonts are loaded into AIS bitmap memory as required by the page log for the next page to be imaged. This occurs only if required fonts are different from the fonts currently being used and not currently loaded for previous pages.

Font specifications loaded into AIS memory

Font specifications are loaded into AIS text and metric memory to provide the necessary line and character spacing information for imaging the page using that font.

Forms loaded

A form is loaded into AIS text and metric memory for the next page to be printed if it is different from the form currently being printed and not currently loaded for previous pages.

Images loaded into AIS memory

Images are loaded into AIS bitmap memory as required by the page log for the next page to be imaged. This occurs only if these images are different from those currently being used and not currently loaded for previous pages.

Paper feeding control

Feeding from the feeder trays is controlled by the output task. When the tray in use becomes empty, an automatic switch can occur to the full tray. (This optional automatic feeding feature is selectable at the terminal or PC UI.) When differing stock is loaded, it may be selected as a function of the criteria embedded within the input data stream.

Report integrity control

The integrity of the report is maintained by the system through character parity checking, automatic page tracking along the paper path, aborting any page which may be affected by hardware error, and automatic page recovery of all affected pages after a paper jam is cleared.

Operator interaction with input and output processing

How and when an operator can affect the print job flow are identified in figure 7-4. Each step is outlined below.

OCS processing The OCS processing steps are:

- Step 1.** The operator enters the START command.
- Step 2.** The input task locates needed files and creates page logs and page buffers.
- Step 3.** The output task may prompt the operator to load stock or confirm cluster definitions.
- Step 4.** The operator may load stock, confirm or modify cluster definitions, or replace toner.
- Step 5.** The operator enters the CONTINUE command or presses the CONT button.
- Step 6.** Input and output processing resume.
- Step 7.** The operator may check job status, cancel or abort jobs, monitor messages, load stock, confirm or modify cluster definitions.

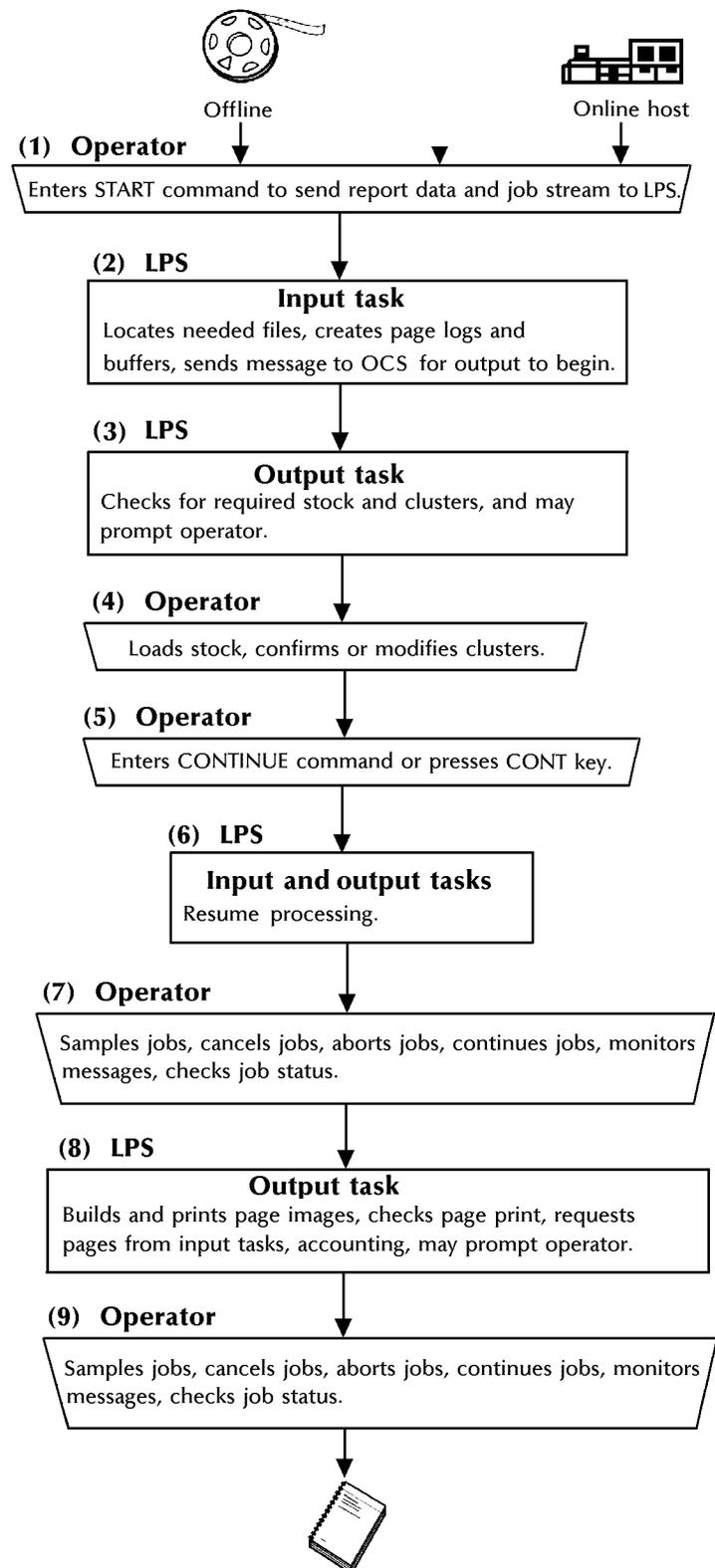
If an operator action interrupts input processing, steps 5 and 6 are repeated.

- Step 8.** The output task directs the building of the page image and prints the page images, verifies page delivery, requests pages from the input task, and records job accounting data. The output task may also prompt the operator to perform the tasks listed in step 3.
- Step 9.** The operator may sample jobs, check job status, cancel or abort jobs, monitor messages, load stock, confirm or modify cluster definitions.

If an operator action interrupts input processing, steps 5 and 6 are repeated.

Figure 7-4 shows how and when an operator can affect the print job flow.

Figure 7-4. **Operator interaction with input and output task steps**



LPS processing status

The LPS printer displays status messages for input processing, output processing, and remote host processing. These message entries appear on the upper right-hand corner of the system controller status area on the terminal when it is in character mode.

These messages appear with the time of day entry. The messages consist of:

- Input status and input device
- Output status and output device
- Host status and host device.

Tables 7-1 through 7-5 list the possible components for each portion of the three messages. The components appear in pairs in order to identify the status of each device. For example, the input message "ActvTape" indicates that the input task is actively processing a tape job.

Table 7-1 lists the possible components of the input device status.

Table 7-1. **Input device status**

Input device	Significance
Tape	Input from tape
Disk	Input from disk
Host	Input from IBM online interface

Table 7-2 lists the possible components of the input processing status.

Table 7-2. **Input processing status**

Input status	Significance
Wait	The input task is waiting for the operator to enter the START command.
Actv	Processing. (The input task is preparing pages for the output task.)
Stop	Input processing has stopped because the operator entered the STOP I, a label processing error was received, or another data error occurred.
Sat.	Input processing has stopped because it has used all the disk space currently available. The output task must release disk space before input can resume processing.

Table 7-3 lists the possible components of the output processing status.

Table 7-3. **Output processing status**

Output status	Significance
Wait	The output task is waiting for input to process pages so that output can prepare the pages for printing.
Actv	Printing. (The output task is sending page images to the printer.)
Stop	Output processing has stopped because the operator entered the STOP O command.
C-up	Catch-up: output has sent to the printer all the pages processed by input. Output is waiting for more pages processed by input.
Jammed	The printer is jammed, a fault or some other fault condition exists or the fuser is warming up and a job is queued for output.

Table 7-4 lists the possible components of the remote host device status.

Table 7-4. **Remote host device status**

Remote host device	Significance
Host	The input data source is the IBM online interface.
XPF	The input data source is the HIP XPAF interface.

Table 7-5 lists the possible components of the remote host processing status.

Table 7-5. **Remote host processing status**

Remote host status	Significance
Attn	LPS attention is being reported to the host.
Offl	Offline. The host is offline.
Nrdy	The host is not ready to send data.
Wait	The LPS is waiting for the host to send data.
Recv	The LPS is receiving data from the host.
Send	The host is sending data to the LPS.
Spnd	The host connection with the LPS is suspended.
OnLP	Online is pending. The host is about to go online to the LPS.

Types of LPS processing

The system executes four types of mutually exclusive print jobs. They are categorized by the method used to initiate them:

- Operator-entered START command jobs
- Operator-entered SAMPLE or HARDCOPY command jobs for a file or group of files
- System utility (PDL, FDL, EDIT) jobs
- Host interface processor (HIP) jobs.

This means, for example, that you can enter a SAMPLE command if a previous SAMPLE command is being processed or if the system is idle, but not if a PDL compile listing is being printed.

Utility, command, compiler, and editor processing

This section describes the commands for running utilities, compilers, and editors.

Running utilities

The utilities supplied with the system are invoked by name. You can run a utility only when the system is idle (such as when the print job queue is empty). Currently included in this category are BBU COMPRESS, DSR, IMA, OSDS, PSC, PURGE, REP, REA, SCL, SFS, SFT, and SST.

Syntax *task-name parm1,parm2,...*

Parameters Table 7-6 shows the optional parameters of the utility commands.

Table 7-6. **Parameters of the Utility command**

Parameter	Description
<i>task-name</i>	1- to 6-character identifier for the task.
<i>parm₁</i> <i>parm₂</i>	Task-specific parameters supplied with the command.

The ABORT command aborts utilities. You do not have to specify the utility since no other jobs or tasks can be active.

Executing command files

Command files are sets of commonly executed commands created with the editor and saved in the CMD directory. To initiate execution, an at sign (@) must precede the name of the command file.

Syntax *file-name.file-type*

Parameters Table 7-7 shows the optional parameters of the command files.

Table 7-7. **Parameters of the Command file**

Parameter	Description
<i>file-name</i>	1- to 6-character filename that exists on disk.
<i>file-type</i>	One of the file directories (which contains a source file) under which the <i>file-name</i> is cataloged. The default filetype is CMD.

Operator commands available during system level command file processing are as follows:

@RESUME

Causes the command file to resume processing after a .PAUSE command has caused suspension.



Note: Certain commands are disabled while in Print Mode.

@ABORT

Aborts command file processing.

Running compilers

The compilers supplied with the system are invoked by name. You can run a compiler only when the system is idle (such as when the print job queue is empty). Currently included in this category are FDL and PDL.

Syntax *task-name parm1,parm2,...*

Parameters Table 7-8 shows the optional parameters of the compiler commands.

Table 7-8. **Parameters of the Compiler command**

Parameter	Description
<i>task-name</i>	1- to 6-character identifier for the task.
<i>parm₁</i> <i>parm₂</i>	Task-specific parameters supplied with the command.

The ABORT command aborts compilers. You do not have to specify the compiler since no other jobs or tasks can be active.

Running editors

The editors supplied with the system are invoked by name. You can run an editor only when the system is idle (such as when the print job queue is empty). Currently included in this category are EDIT and FEDIT.

Syntax *task-name parm1,parm2,...*

Parameters Table 7-9 shows the optional parameters of the editor commands.

Table 7-9. **Parameters of the Editor command**

Parameter	Description
<i>task-name</i>	1- to 6-character identifier for the task.
<i>parm₁</i> <i>parm₂</i>	Task-specific parameters supplied with the command.

The ABORT command aborts editors. You do not have to specify the editor since no other jobs or tasks can be active.

Online printing systems

This section describes various features of a Xerox laser printing system that is channel attached to a host computer and explains how you can access these features.

The LPS may have other interfaces as well: DocuPrintServer (formerly XPSM) interfaces, and DDI interfaces.

Channel-attached LPS

A laser printing system equipped with the *online interface unit* (OLI) may be attached to a byte multiplexer, block multiplexer, or selector channel of an IBM system 360 or 370 30xx, 43xx, and 9370 models, or of an IBM system 390 ES/9000. The channel-attached LPS operates in one-byte, six-byte, eight-byte, or burst mode. No programming changes to the IBM operating system software are required, provided that a 3211 Printer/3811 Printer Control Unit or a 4245 printer is already supported on the host system.

A channel-attached LPS can receive input from the host computer in online 3211/4245 mode. In online 3211/4245 mode, the data stream emulates IBM 3211/4245 line printer format, with or without embedded DJDEs.

System generation

On laser printing systems without a tape drive, the operating system software (OSS) is not loaded from a Xerox system software tape (SST). Instead, the OSS is installed with the HOSTCOPY option of the sysgen processor. HOSTCOPY reads the SST from a host tape drive and sends the OSS to the LPS over the channel. Alternatively, system software is also available on 1/4-inch cartridges.

Refer to the *Xerox DocuPrint 180 LPS System Generation Guide* for further system generation (sysgen) information.

Online 3211/4245 mode

Basic LPS input processing functions are augmented by the following functions for online 3211/4245 mode operations:

- Handling of all interactions with the online interface (OLI) hardware.
- Building of translate tables based on the present contents of the universal character set buffer (UCSB) and the current FOLD/UNFOLD command in effect.
- Processing of forms control buffers (FCBs) received from the host to redefine channel-to-line number assignments and to set margins based on the print position indexing byte.

Print description language (PDL) statements

PDL statements are defined in the *Xerox DocuPrint 180 LPS PDL Reference*. PDL statements which are unique to online (or different from their offline counterparts) are defined more completely in the following paragraphs.

Creating a JDE/JDL

An online job descriptor library (JDL) and its job descriptor entries (JDEs) are created by the user according to the same rules and syntax as described in the *Xerox DocuPrint 180 LPS PDL Reference*. The one PDL statement that must be specified for normal online processing is VOLUME HOST=IBMONL. The HOST command in the VOLUME statement performs two functions:

- Allows the LPS to accept data from the online channel.
- Allows PDL to use a different set of defaults for the RECORD and LINE statements.

The defaults which are automatically selected when VOLUME HOST=IBMONL is specified are as follows:

```
RECORD      LENGTH=150
LINE        PCCTYPE=IBM3211 | IBM4245
           DATA=(0,150)
```

Figure 7-5 shows a sample of an online job descriptor library.

If ANSI carriage control commands are used in the host application program, the host operating system converts each command to a valid 3211/4245 command (listed in the "Online Notes" appendix) before sending it to the printer. Any commands other than those listed are rejected by the LPS online interface (and by the 3211/4245). The PCCTYPE parameter may be coded to IBM3211 or to IBM4245. IBM3211 is the default.

Because the carriage control command is sent prior to, rather than as part of, the data record, offset to fields (such as DJDE prefix and font index) are typically one byte less for online than offline, where the carriage control byte is part of the data record. For the same reason, the online DATA command of the LINE statement has a default offset of zero rather than one.

DJDE processing

Printing is controlled through parameters from the user-defined .JDE file which may be dynamically modified by dynamic job descriptor entries (DJDEs). DJDEs are user created and are processed by the LPS as part of the print data from the host machine. The *Xerox DocuPrint 180 LPS PDL Reference* defines their format and the PDL statements necessary to let the system know they are part of the input data.

Print-and-skip carriage control commands which are associated with online DJDE records are treated as skip-immediate carriage control commands. If the LPS is not at the channel specified, a skip occurs; but if it is already there, no skip occurs unless the last command received with a data record was a print-without-spacing command. Print-and-space carriage control commands associated with DJDE records are ignored.

Nonprint files (for example, font files) can be downloaded from the host using the FILE DJDE. (Refer to the "Downloading files from the host to the LPS" section in this chapter.)

Online optimization

The OPTIMIZE command of the PDL VOLUME statement allows the user to select options for enhancing online throughput. Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for more information.

Copy-sensitive copy modification entries (CMEs)

Copy modification entries (CMEs) which are copy sensitive (that is, a copy range has been specified on the MODIFY command of the PDL OUTPUT statement) are ignored online since the data is not automatically available multiple times. To produce copy-sensitive output, the user transmits data the number of times the data needs to be modified. Thus, if six copies of a report are desired and one CME applies to copies 1 through 4, and another CME applies to copies 5 and 6, the report should be transmitted from the host system twice. The first transmission contains a copy count of 4 with CME1 specified, and the second transmission contains a copy count of 2 with CME2 specified. As an alternative, the application may be successfully printed with only one transmission by using copy-sensitive forms. (Refer to the FORMS command section of the PDL OUTPUT statement in the *Xerox DocuPrint 180 LPS PDL Reference*.)

Report separation

Report separation is defined in terms of banner page detection or other user-defined processing criteria. Logically separated reports are physically offset from one another in the output bins. The two PDL statements which can be used to control online report separation are BANNER and RSTACK. Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for more information.

Universal character set buffers (UCSBs)

The 3211/4245 universal character set buffer (UCSB) feature is supported by the LPS. The UCSB is used by the LPS software as a basis for generation of folded and unfolded translate tables. In a folded translate table, uppercase characters are substituted for only those lowercase characters which are defined as unprintable in the UCSB. Note that any lowercase characters defined as printable in the UCSB are still printed as lowercase if available in the font. In an unfolded translate table, blanks are substituted for characters defined as unprintable in the UCSB. The translate table is rebuilt at the start of each report, when a selected JDE/JDL is processed, when a new UCSB is transmitted from the host, and when a new FOLD or UNFOLD command is received.

The selection of a folded or unfolded translate table is based on the FOLD and UNFOLD 3211/4245 commands. If FOLD has been most recently received from the host, a folded translate table is built; if UNFOLD is the current command, an unfolded table is built.

Online users may restrict UCSB information by accepting or suppressing the normal processing of host-transmitted UCSB data. This can be accomplished by the UCSB = IGNORE command of the PDL LINE statement. If UCSB = IGNORE, lowercase characters print normally if they are present in the font and as blanks if they are not. The translate table is rebuilt only at the start of a report.

The UCSB command can be changed from IGNORE to PROCESS in a selected JDE, but the translate table is not rebuilt until a UCSB load, FOLD or UNFOLD is transmitted from the host.

For most online reports, specification of UCSB = IGNORE is appropriate. The LPS font capability normally negates the need for these translations which were originally defined to handle print train characteristics. If a report using lowercase characters is printed using a font without lowercase, the user may (1) switch to UCSB = PROCESS in a selected JDE, and (2) load a UCSB with lowercase characters defined as not printable and with FOLD specified. The first step allows the UCSB to process. The second step causes a folded translate table to be built so that uppercase characters are substituted for lowercase. At the start of the next report, UCSB = IGNORE comes back into effect and the translate table is rebuilt based on the CODE statement in the .JDE file.

UCSB processing UCSB processing involves the following:

- When a UCSB is transmitted from the host, it is saved on an LPS disk. When sysgen is executed, the UCSB saved on the disk is reinitialized to define all characters as printable.
- If UCSB = PROCESS and the UCSB suppresses special characters used in DJDE syntax (for example = , ; ()), DJDE syntax errors result.
- UCSB = IGNORE should be specified if a CODE other than EBCDIC is specified. If UCSB = PROCESS, a UCSB load causes the CODE default to revert to EBCDIC until end of report or until a selected JDE/JDL is processed.
- More information on the UCSB is provided in the "UCSB" section of the "Online Notes" appendix.

Forms control buffer (FCB)

The 3211/4245 forms control buffer (FCB) feature is supported by the LPS. The FCB defines channel positions and forms length. Online users may restrict the forms control buffer information by accepting or suppressing the normal processing of host-transmitted FCB input. This can be accomplished by the FCB = IGNORE command of the PDL LINE statement.

Vertical format control processing

The interaction of the vertical format controls in the JSL and the FCB sent from the host is as follows:

- When a START command is entered, the VFU table and its associated BOF in the JDE come into effect.
- If no VFU is specified in the JDE, the last FCB sent from the host is retrieved from the disk and becomes active.
- If no BOF is specified in the JDE, the length of the last FCB sent from the host is retrieved from the disk and becomes BOF. Note that TOF is not affected by the FCB.
- If FCB = PROCESS and an FCB is sent from the host, it overrides the previous FCB or VFU. Its length replaces the previous BOF. Furthermore, its contents and length are saved on the disk.
- When a DJDE is used to change a channel assignment, any previous assignments to the channel being changed are suspended until the end of report. At the end of report, DJDE changes are discarded and the last active FCB or VFU comes back into effect.
- When sysgen is executed, the FCB saved on disk is reinitialized to the length of 66 lines with the following channel assignments:

Channel	1	2	3	4	5	6	7	8	10	11	12	9
Line	4	10	16	22	28	34	40	46	52	58	64	66

The default corresponds to the STD2 default supplied by IBM.

- If a print-and-skip to channel command is sent for an undefined channel, the LPS executes a print-and-space-one command.
- If FCB=PROCESS, an FCB load causes positioning to TOF. If the system is already at TOF, page eject is suppressed. If the line number prior to the FCB load is greater than TOF, a page eject occurs.
- The print position indexing (PPI) byte of the FCB is not suppressed by FCB = IGNORE. If an FCB with a PPI byte is received, indexing is invoked; if an FCB without a PPI byte is received, indexing is turned off (even if FCB = IGNORE). Use of the PPI byte to shift the starting print position may cause problems in recognition of both DJDEs and banner pages.

Online record length

The default record length supported by the Xerox laser printing systems is 150 bytes, the same default as the 3211/4245 printer. The user can choose to use larger record lengths with the following restrictions:

- For print record lengths from 151 bytes to 214 bytes, do not use the print position indexing feature that is normally available on the 3211/4245. (One should also note that this feature is invoked only with FCBs; if FCB = IGNORE is invoked, this restriction is not significant.)
- For print record lengths from 215 bytes to 2140 bytes, the user must not use the OPTIMIZE command which provides for online buffering for improved performance. Refer to the PDL VOLUME statement description in the *Xerox DocuPrint 180 LPS PDL Reference* for more information.

To select a record length longer than 150 bytes, the user must specify the new value as the LENGTH command on the PDL RECORD statement and must also modify the DATA command of the LINE statement in order to print the extended character/record length.

Points to note

Keep in mind the following:

- If the data record transmitted from the host exceeds the RECORD LENGTH specified, the record is truncated to the specified record length and no warning is provided.



Note: This action is consistent with the 3211/4245 printer.

- The online dump JDL specifies a RECORD LENGTH of 150 bytes. If you are planning to use a longer record length, modify the JDL to dump the data accurately.
- The LENGTH command in the RECORD statement cannot be changed with DJDE since this would require dynamic rebuffering. Instead, you should consider modifications to the default online .JDL file to simplify operations in the online environment.
- The OPTIMIZE command of the VOLUME statement can be changed with the DJDE JDE= subject to the restrictions listed in OPTIMIZE.

Figure 7-5 shows a sample online job descriptor library.

Figure 7-5. **Sample online .JDL file**

```

ONLINE: JDL:
VFU1:  VFU      ASSIGN=(1,4),ASSIGN=(2,10),ASSIGN=(3,16),
              ASSIGN=(4,22),ASSIGN=(5,28),ASSIGN=(6,34),
              ASSIGN=(7,40),ASSIGN=(8,46),ASSIGN=(9,66),
              ASSIGN=(10,52),ASSIGN=(11,58),ASSIGN=(12,64),
              TOF=1,BOF=66;
              VOLUME      HOST=IBMONL,
                          OPTIMIZE=(NCC,NDC,NPR);
              LINE        PCCTYPE=IBM3211,VFU=VFU1,
                          UCSB=IGNORE,FCB=PROCESS;
              ACCT        USER=(BIN,TRAY);
IDEN  PREFIX='DJDE',SKIP=7,OFFSET=2,OPRINFO=YES;
/*                                          */
/*          TABLES AND CRITERIA          */
/*                                          */
T1:   TABLE MASK='?' ,CONSTANT='HE?DE? PAGE';
C1:   CRITERIA CONSTANT=(1,11,EQ,T1),LINENUM=(1,10);
T2:   TABLE CONSTANT='TRAILER PAGE';
C2:   CRITERIA CONSTANT=(0,12,EQ,T2),LINENUM=(1,10);
T3:   TABLE CONSTANT='EOJ';
C3:   CRITERIA CONSTANT=(0,3,EQ,T3),LINENUM=(1,20);
T4:   TABLE CONSTANT='// JOB';
C4:   CRITERIA CONSTANT=(0,6,EQ,T4),LINENUM=(1,20);
T5:   TABLE CONSTANT=(10)'*';
C5:   CRITERIA CONSTANT=(0,10,EQ,T5),LINENUM=(50,10);
/*                                          */
/*          JOBS WITH NO BANNER PAGE      */
/*                                          */
DFLT: JOB;
/*                                          */
/*          JOBS WITH HEADER PAGES ONLY   */
/*                                          */
HDRP: JOB;
      BANNER TEST=C1,HCOUNT=2,TCOUNT=0;
/*                                          */
/*          JOBS WITH TRAILER PAGES ONLY  */
/*                                          */
TRLP: JOB;
      BANNER TEST=C2,HCOUNT=0,TCOUNT=3;
/*                                          */
/*          JOBS WITH BOTH HEADER AND TRAILER PAGES */
/*                                          */
BOTH: JOB;
      BANNER TEST=(C1 OR C2),HCOUNT=2,TCOUNT=3;
/*                                          */
/*          OTHER JOBS                    */
/*                                          */
EOJ:  JOB;
      BANNER TEST=C3,TCOUNT=1;
JOB:  JOB;
      BANNER TEST=C4,TCOUNT=1;
POWER: JOB;
      BANNER TEST=C5,TCOUNT=2;
/*                                          */
      END;

```

Operating the system

Operation of the LPS in online 3211/4245 mode is essentially the same as the basic LPS offline system. The operator commands specific to online operations are described in table 7-10. They allow the LPS to go online, to go offline, and to control various online operations.

Placing the system online or offline

Use the following procedure to place the LPS online.

1. At the LPS terminal, key in the online command. The system responds by displaying `OS0030 System is "ON-LINE"`
2. At the host terminal, issue the VARY ONLINE (or equivalent) command.

Until you enter a START command, the LPS is in an online but NOT READY state, and the host does not transmit any data.

Use the following procedure to move the LPS offline.

1. At the host terminal, make sure the job, queue, or class is purged/drained, then issue the VARY OFFLINE (or equivalent) command.
2. At the LPS terminal, key in the command ENDJOB to purge the LPS print file, then key in the command OFFLINE. The printing system responds by displaying the message `OS2570 Vary system offline at the host and enter "CONTINUE I"`. After you enter CONTINUE I, the message `OS0040 SYSTEM IS OFFLINE` displays.



Note: Some host systems may not allow a VARY OFFLINE (switching tape drives) while an Intervention Required is posted. In this case, the LPS may be taken offline before varying offline as long as the host is not reinitialized at that time.

Starting an online job

Use the START command to initiate an online job (printed output) and to set the online interface to READY. After you enter the START command, you can initiate data transmission from the host to the LPS. It is best to initiate data transmission after the `ONLINE WAITING FOR DATA` message displays. The START command is described in the "START (start a print job)" section of the "Running print jobs" chapter.

Monitoring online operations

The LPS operator may monitor the current input status, output status, the current remote source, and remote source status. (For details, refer to the "Verifying system status" section of the "Setting LPS parameters" chapter and the "Checking print job status" section of the "Running print jobs" chapter.)

Printing all completed pages

Use the DRAIN command to print all completed pages without ending the online job in progress. When the DRAIN command executes, output processing catches up with input processing, leaving two pages yet to be printed. These pages are printed after:

- You enter an ENDJOB command
- The host transmits additional pages and you enter a DRAIN command.

This command is ignored if the system is offline. Pages are printed if data pages are in the print file or in memory; pages are not printed if there are no data pages in the print file or in memory, if only header banner pages are in the print file or memory, or if there is only a partial page in the print file or memory. A page is considered completed if a transition to the next page has occurred (skip-to-channel-1 or print commands that cause BOF to be exceeded).

Completed pages in memory (except for header banner pages) are written to the print file before output is started. If the command is ignored or no pages are printed, the printing system responds by displaying OS1170 Invalid command while offline or OS1160 No pages waiting to be drained.

Ending an online job

Situations arise when the LPS becomes inactive for lack of data from the host, but there is still some data remaining to be printed. This condition generally occurs when reports do not contain trailer banner pages indicating an end-of-report or the last trailer has not been sent completely. At this point, the LPS must be informed externally with an ENDJOB command to print out the remaining pages. An ENDJOB command will cause a page transition and flush the buffer. At the completion of this operation, the system waits for the next START command.

Multicopy disk saturation

Making multiple copies of a report may lead to a situation in which operator intervention is required. If the number of pages of a multicopy document is smaller than the capacity of the system print file (as determined by the REALLOCATE system command), no intervention is required. But if the document is greater than the size of the print file, operator intervention is required.

If disk saturation occurs on a multicopy report, the following message is displayed:

```
OS6300 Current report saturates disk. enter option:
      "CONTINUE I" - Continue multicopy mode
      "PRINT1"     - Print one copy of current
                   report
      "ENDJOB"     - Print all copies of report
                   so far, end
      "ABORT O"    - Abort current report
      "ABORT n"    - Abort current job
```

The possible operator actions are as follows:

- Key in CONTINUE I to continue multicopy processing and print the requested copies of the report n pages at a time, where n is the number of pages that the print file can hold at one time. The copies then need to be collated manually.
- Key in PRINT1 to discontinue multicopy processing for this report only, print one copy of the current report, and continue accepting data from the host. The report must then be retransmitted from the host for each copy required.
- Key in ENDJOB to mark the current job as ended (system to wait for the next START command).
- Key in ABORT O to abort the current report being printed; resume printing with the next report.
- Key in ABORT n (where n is the current job number) to go NOT READY, mark the current LPS job as aborted, and purge the current LPS print file.

Handling MOUNT FORM message

The host terminal message MOUNT FORM X ON YY, where YY is the particular LPS device, can imply a change of stock, an LPS-catalogued form, or a multicopy form. If this message is displayed, the LPS operator should key in ENDJOB to print unimaged pages from the last report. The operator should then set up for the next job based on the message content on the host display and enter the appropriate START command on the LPS to begin the next job.

Table 7-10 describes the operator commands specific to online operations.

Table 7-10. **Online operator commands**

Command	Function
ABORT <i>n</i>	Instructs the LPS to go NOT READY, mark the current LPS job <i>n</i> where <i>n</i> is the LPS job number as aborted, and purge the current LPS print file (such as any pages remaining unimaged are not printed).
ABORT O	Aborts the current report being printed. Printing resumes with the next logical report in the current LPS print file.
ABORT I	Aborts the current report in input. Input proceeds to the next report and resumes normal processing.
CONTINUE	Instructs the LPS to go READY for normal operations.
DRAIN	Instructs completed pages in the print file or in memory to be printed without ending the job in progress.
ENDJOB	Instructs the LPS to go NOT READY, mark the current LPS job as terminated, and print all remaining unimaged pages.
OFFLINE	Instructs the LPS to go offline.
ONLINE	Instructs the LPS to go online and NOT READY.
RESET	Instructs the LPS to go offline and purge the current print file (stops all system printing activities).
START	Instructs the LPS to go READY and process host commands based upon: <ul style="list-style-type: none"> • Host-supplied control information (FCB, UCSB). • User-supplied parameters (JDE, JDL, DJDE).
STOP	Instructs the LPS to go NOT READY and to stop accepting all commands from the host except SENSE.

Host-induced idle time

Under various circumstances, the LPS may experience host-induced time delays when receiving print data. The magnitude and frequency of these delays vary considerably depending on the type of host spooler being used, the type of I/O channel connection, and the number of devices sharing the channel. Experience has shown that these host-induced delays can be several milliseconds every few lines up to many seconds in between *host-defined* jobs.

The online IDLE (SEC) entry on a user's accounting page may provide an indication of the LPS experiencing host-induced delays. The value associated with this entry represents only *whole-second*, host-induced delays detected by the LPS. It does *not* take into account multiple milliseconds delays which may be occurring and which could, over the course of a report, add up to additional seconds. Thus the online IDLE (SEC) field represents the *minimum* amount of host-induced delay that the LPS may be experiencing.

Online recovery

In the event of a system failure, the data which has been formatted for print and stored on the disk can be imaged upon system restart if job recovery is requested. Data which was being received and processed at the time of the rollover is lost and requires retransmission.

The amount of data which requires retransmission depends on two factors: the options specified on the PDL VOLUME OPTIMIZE statement and DJDEs which indicate the use of interleaved graphics.

- If NPRECOVER is specified, up to nine data pages are buffered in memory before being written to the disk. When end of report is detected, all pages in memory are written to the disk. This means that if NPRECOVER is specified, up to nine data pages would require retransmission, but not typically from more than one report.
- If NPRECOVER is not specified, up to two pages would require retransmission. If interleaved graphics are being transmitted at the time of system failure, the amount of data requiring retransmission depends on the number and size of the graphics.

The following messages are displayed during the recovery process:

OS0030	System is 'ONLINE'
OS2960	Back up host to start of report or n pages whichever is less.
OS2035	INPUT stopped. Enter CONTINUE I when host is ready.



Note: The number of pages (*n*) is determined by the number of skip-to-channel-1 commands encountered since the last set of pages were written to the disk.

Online dump

The online dump feature provides the user with the capability to generate a hard copy of the command and data transmission taking place between the host computer and the LPS.

Starting and ending dump sessions

The dump session is started with a START command when the LPS is online and in the idle state. The online dump JDE/JDL used must specify VOLUME HOST = OLDUMP and CODE = EBCDIC.

The dump session is normally terminated by either the ENDJOB or the offline operator command (refer to the "Online notes" appendix). Other online operator commands during a dump session perform the standard functions.

Dump format

Each host command directs the printer to perform an operation, or it provides control information used during printing of a report. The operations performed are either print with carriage control, carriage control only, or control (for example load FCB). All print commands and some control commands transmit data to or from the LPS. The dump format is designed such that for each of these commands there exists a HEADER and DATA segment. For skip or space immediate commands and the control commands FOLD, UNFOLD, BLOCK DATA CHECK, ALLOW DATA CHECK, RAISE COVER, and DIAGNOSTIC GATE, the dump format consists of a HEADER segment only. An example of the online dump format is provided in the "Online notes" appendix.

Table 7-11 describes the six fields of the HEADER segment.

Table 7-11. **HEADER segment fields**

Field	Description
HOST COMMAND	The English translation of the host command in hexadecimal.
END STATUS	One byte of status in hexadecimal transmitted to the host.
SEQ#	Incremental count in decimal of the host commands received during a dump session. The maximum number generated is 999,999.
LENGTH	Length in decimal of the data transmitted.
DATE	Date when the host command was received by the dump processor.
TIME	Time when the host command was received by the dump processor. The time includes tics of seconds where one tic is approximately one millisecond.

The DATA segment consists of the input data in hexadecimal transmitted to or from the host with EBCDIC translation based upon the CODE statement. Input data that cannot be translated is left as blanks. Each line of data is preceded by a byte count in hexadecimal.

Points to note

Note the following:

- DJDEs, banner, and other logical processing functions are not honored when running in the dump mode, so an ENDJOB must be entered to end the dump session.
- The NO-OP command from the host is not dumped since it is intercepted and processed by the OLI hardware and is transparent to the LPS software.
- Hardware errors are reported in the system error log but are not dumped due to Host/OLI/Driver interface constraints.
- Commands processed when input is stopped (for example when paper trays become empty) are not dumped.
- Print records longer than the specified RECORD LENGTH value are truncated without warning. If records longer than 150 bytes are to be sent, the RECORD LENGTH parameter in the online dump JDE/JDL should be modified to the longer length.
- Entering the SAMPLE command (with parameters) is not allowed while the system is online whether or not a job is running. Exceptions to this are: pressing the <SAMPLE> key or entering the SAMPLE command with no parameters.
- The "HIP START ..." command is accepted, but HIP will be unable to submit a print job while the system is online.

Downloading files from the host to the LPS

Files may need to be downloaded from the host for various reasons. In a production printing environment, file storage space may be limited by the requirement of a large print file. At some installations, files such as form files which undergo periodic revision may be compiled and kept on the host computer and transmitted to the printer when required by a job. System patch tapes may be required. Other LPS font and logo tapes may need to be downloaded.

Two methods of downloading files are available:

- HOSTCOPY utility
- DJDE FILE command.

The HOSTCOPY utility requires the system to be placed offline and idle, then rebooted from disk to enter the HOSTCOPY command from the system menu. It does not allow files to be transferred while normal printing activity is taking place.

The DJDE FILE command transfers files while normal printing activity is taking place; keep in mind that the file to be transferred may need to be preconditioned to prevent trailing-blank truncation by the host spooler.

(A third method of file transfer, using the LPS/host file transfer interface, may be used to transfer user files and also allows an online user to obtain the status of the reports submitted for printing on the LPS. A remote connection between the host and LPS is used for file transfer.)

Valid download file types

Files may be downloaded in two formats: card-image files and LPS-labeled files. Certain destination file types are not accepted for either LPS-labeled or card-image files. These are file types OSD, SAF, SYS, \$Y\$, and TSK. In addition, card-image format files are restricted to destination file types CMD, FSL, JSL, MSC, PCH, TMP, and TPF.

HOSTCOPY

HOSTCOPY runs on online systems. It provides a means for accepting files transmitted from the host computer and storing those files on the LPS system disk. Files in two different formats may be transmitted: (1) user-created card-image files and (2) LPS-labeled files (such as font, logo, and patch tapes).

Card-image format files

Files may be transmitted to the LPS in 80-character EBCDIC records. A group of records stored as a data file is called a packet. Each packet must be preceded by a control record which identifies the file name and type where the following records are stored. This control record has the following format starting in column 1:

\$\$\$START *file-name.file-type*

where *file-name* is a 2- to 6-character name and *file-type* is one of the valid file type designations listed previously. Either a period or a

comma is acceptable as the separator between the file name and the file type.

One or more EBCDIC packets may be transmitted at one time. Each packet should be preceded by the \$\$\$START record. After the last packet, the transmission must be ended by another control record. The format for this record, starting in column 1 is:

\$\$\$END

Each packet transmitted may not exceed 3072 records. Exceeding this limit results in truncation of the file to 3072 records.

If any problem is found with the \$\$\$START record, one of the following messages is displayed with the \$\$\$START record displayed on the next line.

Invalid file type
INVALID CHARACTER IN FILE NAME OR FILE TYPE
MISSING "." OR "," IN \$\$\$START RECORD

If any of the above errors are encountered, the file is not created on the disk, and the user is given the opportunity to continue or abort HOSTCOPY. Any file names greater than six characters are truncated to six characters. If a \$\$\$START record is followed immediately by a \$\$\$END record, the message FILE SIZE EQUALS ZERO, is displayed. The file is not created on the disk, and the user is given the opportunity to continue or abort HOSTCOPY.

Font and logo tape file format

Font and logo tapes are LPS-labeled tapes issued by Xerox. They have been structured for use with online systems as well as offline systems. The last physical file on these tapes is a concatenation of all previous files on the tape. It is the last file which is in a format that can be transmitted across the host I/O channel to the printing system. A host utility must be used to advance the tape to the beginning of the last file. Font and logo files have 128-byte blocks, followed by blocks of multiples of 512 bytes, from 512 bytes to 8192 bytes. The files must be deblocked to 128-byte records before transmission to the LPS.

Example The following is a sample JCL using the IEBGENER host utility to send a concatenated file to the LPS. In the example, "01E" is the device address of the printing system. For illustration purposes, the tape to be transmitted contains 246 individual system files. Appended to these 246 files are a dummy ENDFIL file and a concatenated file containing the preceding 247 files. Thus, the first parameter after the LABEL = is the file number of the file to be transmitted (in this case, file number 248).

```
//LOADXERX EXEC PGM=IEBGENER,REGION=80K
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD UNIT=01E,DCB=(RECFM=FB,LRECL=128,BLKSIZE=128)
//SYSUT1 DD UNIT=TAPE,DCB=(RECFM=FB,LRECL=128,BLKSIZE=8192),DSN=XEROX,
// DISP=OLD,LABEL=(248,NL,EXPDT=98000),VOL=SER=XEROXO
```



Note: Some operating systems may require that an FCB and UCSB be transmitted with the job. These are accepted, but they are ignored by HOSTCOPY.

The program (for example IEBGENER) which transmits files to the LPS must not add page numbers or do any reformatting of the data. In particular, no additional records may be added between the files.

File transmission

The startup procedure for readying the LPS to accept files across the channel is as follows:

Vary the LPS offline at the host system, and ensure that the LPS status is offline IDLE. Press the Boot switch.

The following message appears:

```
READY *
$BD <ENTER> (user's response)

**** SYSGEN PROCESSOR ****

COMMAND

COMMANDS DISPLAY SYSGEN COMMANDS
BOOT BOOT THE OPERATING SYSTEM
SYSGEN BUILD OR UPDATE SYSTEM FILES ON DISK FROM TAPE OR HOST
FORMAT CLEAR AND FORMAT DISK PACK
HOSTCOPY COPY USER FILES FROM HOST TO DISK
AUTO AUTO SYSGEN
MINI CONFIGURATION CHANGE ONLY

ENTER COMMAND
>HOSTCOPY <ENTER> (user's response)
*SYSGEN RUNNING FOR HOSTCOPY ONLY*
THIS IS A FILE COPYING PROCEDURE, NOT A SYSGEN
CONTINUE OR ABORT? ENTER 'C' OR 'A'
```

This message warns the operator that sysgen not HOSTCOPY, must be used when transmitting a system software tape (SST). If the operator enters C, the system responds with:

WAITING TO READ FILES FROM HOST

Notify the host computer operator to vary the printing system online.

The printing system is now ready to receive data in either format discussed earlier. Any records sent prior to a \$\$\$START or LPS tape label are ignored. As soon as a valid \$\$\$START record or LPS tape label is received from the host, the following message is displayed:

TAPE NOW BEING PROCESSED

The data which follows is stored on the LPS system disk as one or more files. The message CREATING FILE xxxxxx.yyy is displayed for each file.

HOSTCOPY posts the following message on the LPS terminal when it detects a \$\$\$END record or a Xerox-formatted ENDFIL.END logical file:

END OF TAPE FOUND
IS ANOTHER TAPE TO BE SENT? Y or N

If another tape is to be transmitted, the operator enters Y. HOSTCOPY continues to accept data from the channel. When N is entered, the following message is displayed:

VARY OFFLINE AT HOST, THEN ENTER C

After C is entered, HOSTCOPY terminates. Once the operating system is reloaded after HOSTCOPY completes, the FDL and PDL compilers should be invoked in the usual way to process any new .JSL and .FSL files. Files loaded onto the system with the HOSTCOPY command must be retrieved by the editor MERGE command prior to editing so that valid sequence numbers can be inserted. Refer to the "Editing source files" chapter for more information.

If EBCDIC files are transmitted, the final CREATING FILE xxxxxx.yyy message is displayed after response to the message END OF TAPE FOUND. IS ANOTHER TAPE TO BE SENT? Y or N.

DJDE FILE

This DJDE provides a capability of loading card image or LPS-labeled files to the LPS disk while a printing job is in progress. The print job can be invoked solely for the purpose of downloading files; in other situations, the files can be interleaved with variable text data so that they are available when referenced by input or output processing. In either case, a DJDE packet must immediately precede the data for a file (or batch of files) which is copied to the LPS disk. The DJDE packet must include a FILE command which is used to signal the start of the file data.

The FILE command is a record-oriented DJDE and is applied immediately. It can be included with other record-oriented or page-oriented DJDE commands in a DJDE packet. The FILE command is described in the *Xerox DocuPrint 180 LPS PDL Reference*.



Note: The DJDE FILE command does not allow a file replacement or deletion transaction to occur if the file to be replaced or deleted is secured and the logon level is not 5.

Card-image file processing

The FILE command is checked for valid parameters, and an appropriate amount of disk space is made available for the file.

- If the destination file type is not acceptable for a card-image file, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `File processing error - Illegal destination file type`. Input then proceeds to read and discard the extent of the file.
- If the file name is improperly specified, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `File processing error - Invalid destination file name`. Input then proceeds to process the data as variable text data.
- If more card-image records are read than is predicted by the parameter specifying the maximum number of card images, the extra records are read and discarded. The preceding DJDE packet is printed on an OPRINFO page together with the message `File xxx.yyy overflow - Excess card images lost`.
- If the file cannot be created because of insufficient space on the disk, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `No space on disk for file xxx.yyy`. Input then proceeds to read and discard the extent of the file.

When a card-image file record is read, the PCC byte is ignored. Block and record delimiters (if any) are used to determine record length. Records with a length greater than 80 bytes are truncated to 80 bytes; records with a length less than 80 bytes are padded with ASCII blanks to 80 bytes. Character translation may be performed, depending on the JDE then in effect.

Sequence numbers are not inserted into the records prior to storing, so the file must be retrieved by the editor MERGE command prior to editing so that valid sequence numbers can be inserted. Refer to the "Editing source files" chapter for more information.

Any DJDE record serves to terminate card-image record processing for the file being downloaded.

- If a DJDE record is encountered and one or more card-image records have been written in the file, the file is truncated to an appropriate size and closed. If OPRINFO = YES, the message `FILE xxx.yyy CREATED` is listed on an OPRINFO page.
- If a DJDE record is encountered before any card-image records have been filed, the preceding DJDE packet is printed on an OPRINFO page together with the message `FILE xxx.yyy ERROR - NO CARD IMAGE DATA`. The null file is deleted.

File processing mode is concluded and print mode is resumed with the DJDE record unless the DJDE record contains another FILE command.

LPS-labeled file processing

The FILE command is checked for valid parameters, and the LPS label record at the beginning of the file is read. If the destination file type is acceptable, the file size field is obtained from the label record, and the extent of the file is calculated.

- If the very first label record does not meet the requirements of a label record, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `File processing error - No label record`. Input then proceeds to process the data as variable text data and prints, intermixed with the report.
- If the destination file type is OSD, SAF, SYS, or TSK, or if the file type specification in the DJDE command is not one of the file types included in the LPS file directory, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `FILE PROCESSING ERROR - ILLEGAL DESTINATION FILE TYPE`. Input then proceeds to read and discard the extent of the file.



Note: If the destination file type is \$Y\$, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the following messages:

```
OS6700      Syntax error in DJDE.
OS2006      Enter CONTINUE I or ABORT I.
OS6736      File processing error - No label
             record.
OS2006      Enter CONTINUE I or ABORT I.
```

- If the file cannot be created because of insufficient space on the disk, one of two courses of action is taken. If file space is freed up after a previous report is completely printed by output, input displays the message `Disk full - INPUT waiting for OUTPUT to complete` and waits for such a report to be completed before trying again to create the file. If there is no possibility of disk file space becoming available, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message `No space on disk for file xxx.yyy`. Input then proceeds to read and discard the extent of the file.

The file's data records are read and stored into the appropriately named file until the total extent of the file is spanned. PCC bytes and block and record delimiters are ignored while processing the file data.

- If the DJDE command `FILE = ()` signaled the start of LPS-labeled data, input presumes that more than one LPS-labeled file may be batched together and checks for another label record when it completes processing a file.
- If `OPRINFO = YES` and the file transfers are successfully concluded, the file name and file type of each file transferred are listed on the OPRINFO page, if selected, together with a brief indication of whether the file was newly created or replaces an existing file.

Delimited records for LPS-labeled files

To prevent loss of data through trailing blank suppression procedures on the host spooler, data records for LPS-labeled files must be delimited by appending an extra nonblank character at the end of each record. The same nonblank character must be used with all records processed in the file processing mode.

If the nonblank character that is appended at the end of the records is not consistently present throughout the extent of a file, the preceding DJDE packet is printed on an OPRINFO page, if selected, together with the message FILE xxx.yyy PROCESSING ABORTED DUE TO DATA ERROR. Input then deletes the partial file that was created and proceeds to process the data as variable text data.

Small print files and online report control

This section describes how to print small files.

LEASH

Small print files, values less than the minimum values can be reallocated by restricting input processing to control report spooling. (Refer to "Reallocating space to the print file" section of the "Setting LPS parameters" chapter.) Online users (IBM 3211/4245 only) can specify the number of reports the input task can process ahead of output. In effect, input is "leashed" to output. The length of the "leash" is measured in reports. To enable this control mechanism, a special command file has been developed (LEASH.CMD) that prompts you for the number of reports and then applies a patch that allows the report control method to take effect. To run the LEASH.CMD command file, log on to system level 3, then enter:

Syntax @LEASH

LEASH sends reports to input processing and then queues them for output processing. The LPS prompts you to send the next queued report to output by displaying the message:

Intervention required.

The number of reports entered in the command file session represents the maximum number of reports that can be queued to output.

8. HIP processing on the LPS

The following conventions are used in this chapter to describe commands entered from the laser printing system keyboard:

- Optional parameters to a command are enclosed by brackets as follows:

[parameters]

- Several options for a parameter are shown separated by a slash as follows:

[Selection 1/Selection 2/Selection 3]

- An optional parameter list is indicated by one comma and an ellipsis within brackets which follow a parameter as follows:

[parameter[,...]]

Spaces may be inserted following commas (,) and semicolons (;) to improve legibility.

Initiating communications

The HIP task resides on the laser printing system disk. It is loaded into memory and executed when the operator enters the following command:

`HIP [option[: option [;...]]]`

The options are the same options used to control communications. Operator commands follow.

The HIP task responds with the confirmation message:

```
HP0010 HOST Interface Processor loaded
```

If the host is not online and responding, the following message is displayed periodically until the host comes online or HIP is terminated:

```
HP1240 HOST not responding
```

Operator commands

Enter all commands at the LPS keyboard in the following format:

HIP [option[: option [...]]]

Once initiated, HIP software operates as a background task in parallel with normal printing system activities. The HIP task accepts files for printing or storing on the system disk depending on the communication mode. Host job print files are stored temporarily in a HIP buffer file, and START commands are submitted to OCS for printing. Normally, the files are printed in the order received, but the operator can alter this order.

A list of all the command options and their actions follows. Note that listed options are separated by semicolons while parameters are separated from the option keyword by a space. If the system detects an error in the command line, the incorrect part of the line is displayed. The system requests you to correct and reenter the command as follows:

```
HP2300 Error in command at: <part of command in
error>
HP2310 REENTER COMMAND
```

Some commands are not valid for all hosts, and some commands are not valid if a host is not selected. When this happens, one of the following messages is displayed:

```
HP1320 Function is not supported by current host
HP2320 Invalid command with current HOST type
HP2330 Invalid command with no HOST specified
```

HIP

Use this command to invoke both the HIP ONLINE and the HIP START commands. It also loads the HIP processor for the last host loaded.

It is the routinely used method to start the HIP task with the default specified host as defined in HIP.LIB. (Refer to the "HIP library commands" section in this chapter.)

Syntax HIP

HIP ABORT

This command aborts all jobs or the listed jobs from the HIP internal queue. The SEND option specifies that a HIP to host file transmission job will be aborted; otherwise, it is assumed to be a host to HIP transmission job. The job file currently being received will be aborted if either ALL or no parameter is specified. Queued and active jobs cannot be aborted. Only jobs being received or pending may be aborted. If a listed *jid* is not eligible, it will be listed in the appropriate error message shown below. The status of the resultant job queue can be determined by displaying the queue with a HIP SHOW JOBS [parameter] command.

If ALL is specified, confirmation is requested by the message:

```
HP2110 All jobs will be lost. Confirm with 'YES'.
```

One or more of the following messages will also be displayed:

```
HP0210 Transmission aborted: JOB ID 0000.
```

```
HP0120 Job aborted; JOB ID 0000.
```

```
HP1140 Cannot abort job(s): 0000 [,...].
```

```
HP1100 No jobs to abort.
```

Syntax HIP ABORT [SEND][{*jid*[,...]}|ALL]

Parameters Table 8-1 shows the parameters of the HIP ABORT command.

Table 8-1. **Parameters of the HIP ABORT command**

Parameter	Description
SEND	Aborts a HIP-to-host file transmission.
(no SEND) parameter	Aborts a host-to-HIP file transmission.
<i>jid</i>	Specifies the name of the file to abort.
ALL	Aborts all host-to-HIP file transmissions.

You must initiate file transfers to an XPF or XPS host from the host, not using the HIP SEND command.

HIP CONCATENATE

This command controls the concatenation mode of processing, or displays the current mode. The commands HIP CONCATENATE and HIP CONCATENATE DISPLAY report the current processing mode and the operator display mode. The commands HIP CONCATENATE {ON | OFF} and HIP CONCATENATE DISPLAY {ON | OFF} result in concatenation mode processing or operator mode to turn on or off. For all commands, the selected mode of processing is reported by one of the following messages:

```
HP0090 Job Concatenation mode is 'ON'.
HP0095 Job Concatenation mode is 'OFF'.
HP0080 Concatenated job operator display mode is
'ON'.
HP0085 Concatenated job operator display mode is
'OFF'.
```

Syntax HIP CONCATENATE [DISPLAY]{{ON | OFF}}

Restriction Jobs using RSTACK cannot be run as a concatenated job. They **must** be run as separate jobs. Neither HIP nor INPUT enforce this restriction, which means, you must ensure that such jobs are not submitted as concatenated jobs.

HIP CHANGE

This command changes a HIP spooled data file into a standard printer file resident on the disk. Only one HIP CHANGE command can be processed at a time. The default file name is HPxxxx.XCS where xxxx is the four-digit jid and the default type is XCS. Note that the file-type XCS must be coded in the ACCEPT command in the HIP.LIB file in order to be an accepted file-type by the CHANGE command. The only accepted file types are those file types listed in the ACCEPT command defined in HIP.LIB. If a file name is used, the entire syntax, with type being optional, must be specified. The input HIP data file must be completely received and in the PENDING state with HIP in the STOP state. The output file name must be unique from any file name already on the disk of that particular printer. A successful CHANGE will show a status of STORED in the HIP JOBS display, and the LPS will not print the job with HIP.

Syntax HIP CHANGE *jid* [TO *file-name* [*.file-type*]]

Parameters Table 8-2 shows the parameters of the HIP CHANGE command.

Table 8-2. **Parameters of the HIP CHANGE command**

Parameter	Description
<i>jid</i>	Specifies the HIP assigned number of the job to store on the disk as a file.
<i>filename</i>	Renames the file you are storing on the disk.
<i>file-type</i>	Specifies the file type.

HIP DRAIN

This command instructs HIP to end the current concatenated job and prints any remaining pages of a previously processed job. HIP remains in the START mode and will restart a new concatenated OCS job with the next job available in its queue. If HIP is not processing concatenated jobs, this command has no effect. No HIP messages are displayed in response to this command.

This command can be used at any time. Excessive use during concatenated job processing may reduce printing throughput by causing extra engine starts and stops between successive jobs.

Syntax HIP DRAIN

HIP END

This command causes HIP to shut down in an orderly manner. A STOP and OFFLINE are effected. When the current files being received/transmitted and printed (if any) are completed, the internal job queue will be saved on the system disk for later restart. HIP terminates and the task is unloaded from memory. This command is irreversible once issued. HIP may not be brought online until the system completes the HIP END process and displays the following message:

```
HP1020 HOST Interface Processor unloaded.
```

Syntax HIP END

HIP EXIT

This command is identical to the HIP END command. You can use HIP EXIT and HIP END interchangeably to terminate HIP processing in an orderly manner.

HIP HOST

This command selects the specified host when the system has more than one host. It is not necessary on systems with only one host because HIP performs the function automatically each time it is loaded if the HOST default in HIP.LIB is the configured host. HIP must be offline and idle in order to switch from one host to another. If the specified host does not exist on the system, this command is ignored. If the system is not offline, or is not idle, the command is rejected and one or more of the following messages are displayed:

```
HP2020 Selected HOST not present on system
HP2400 HIP interface must be OFFLINE. enter 'HIP
OFFLINE'.
HP7100 Can not change HOSTS while jobs are Pending
or Active.
```

Syntax HIP HOST {XPF or XPS}

Parameters Table 8-3 shows the parameters of the HIP HOST command.

Table 8-3. **Parameters of the HIP HOST command**

Parameter	Description
XPF	Enables communication between a host and the LPS in an IBM MVS/XA environment. XPF is also called XPAF.
XPS	Identifies Xerox Print Services (XPS) Manager as the printer front end to other hosts.

Once you access one of these communication modes, you can use any of the following HIP commands to control communications.

HIP JOBS

This command is identical to the HIP SHOW JOBS command. You can use HIP JOBS and HIP SHOW JOBS interchangeably to check the status of jobs in the queue.

HIP ONLINE

Use this command to start the HIP task with the default specified host as defined in HIP.LIB. During HIP processing, the ONLINE command is not available. Enter HIP alone to invoke both the HIP ONLINE and the HIP START commands.

Syntax HIP ONLINE [ALL]

Parameter Table 8-4 shows the parameter of the HIP ONLINE command.

Table 8-4. **Parameters of the HIP ONLINE command**

Parameter	Description
ALL	Specifies that both print and nonprint files are accepted.

ONLINE ALL is accepted in logon level class 1 and above for all hosts. The default is that only print files are accepted. As print jobs are received, they are entered into HIP's internal job queue and become eligible for submission to OCS for printing. One of the following messages is displayed:

```
HP1210 HOST ONLINE. Accepting ALL jobs.
HP1220 HOST ONLINE. Accepting print jobs only.
HP2010 No HOST Selected.
HP2030 "ALL" not allowed at current LOGON level.
HP0200 Failed to go ONLINE.
```

If the selected host is not online and responding, the following message is displayed periodically until the host comes online or the online request is terminated with HIP OFFLINE command:

```
HP1240 HOST not responding.
```

HIP RESET

This command clears the job queue. HIP must be offline and have no jobs in a PENDING, QUEUED, OQUEUED, PRINTING, or ACTIVE state. If any of these conditions are not met, the following message is displayed:

```
HP1280 'HIP RESET' not allowed. Invalid task state.
```

Syntax HIP RESET

Procedure To prevent accidental loss of the jobs in the queue, the following message is displayed:

```
HP2110 All jobs will be lost: Confirm with 'Y'
```

After the queue has been cleared, the following message is displayed:

```
HP1120 HIP job queue cleared
```

One or more of the following messages are also displayed:

```
HP0210 Transmission aborted: JOB ID 0000.
```

```
HP0120 Job aborted: JOB ID 0000.
```

```
HP1230 HOST OFFLINE.
```

HIP SHOW BUFFER

This command displays the current status of the HIP buffer with the following message:

```
HP1330 Buffersize = xxxxx, blocks, Utilization =
uu%.
yyyyy blocks used, zzzzz blocks available.
```

Syntax HIP SHOW BUFFER

HIP SHOW HOSTS

This command displays the current host and a list of selectable hosts. The following information is displayed:

```
HP1340 Known and current HOSTS are:
XPF XPS NONE
```



Note: <-- indicates the current host. NONE is displayed when a host has not been previously selected by the HIP HOST command or by the HOST command in HIP.LIB. NONE also displays when there is a system error, causing the system to deselect all hosts.

Syntax HIP SHOW HOSTS

HIP SHOW JOBS

This command displays the current status of the HIP task and a selected list of the internal jobs. The options for the parameter are described below. The format and the contents of the display are discussed in the “Job queue display” section.

Syntax HIP SHOW JOBS [*parameter*]

Parameters Table 8-5 shows the parameters of the HIP SHOW JOBS command.

Table 8-5. **Parameters of the HIP SHOW JOBS command**

Parameter	Description
ABORTED	Displays all jobs that have been aborted by the HIP ABORT command.
ACTIVE	Displays all jobs in the active state: QUEUED, ACTIVE, OQUEUED, or PRINTING.
ALL	Displays all jobs in all queues.
CANCELLED	Displays all jobs that have been cancelled by a host restart.
COMPLETED	Displays jobs in a completed state: PRINTED or ABORTED.
CONTINUE	Continues a previously requested display if all jobs in the queue were not displayed.
PAGE[<i>n</i>]	Specifies a particular page of job information. The page number <i>n</i> is from 1 to 6 specifying the number of pages. If you do not specify <i>n</i> or if <i>n</i> =0, the default is the 20 jobs starting with the last COMPLETED job.
PENDING	Displays all jobs in the pending state: PENDING, RECEIVING (but not submitted to the OCS).
PRINTED	Displays the jobs in the printed state: PRINTED, or TRUNCATED.
SENT	Displays all files that have been sent to the host.
STORED	Displays all files that have been stored by the host.
SUMMARY	Displays a summary of the job queue. (This is the default.)

Internal job queue display

The current status of jobs in the internal queues may be reviewed by using the following command:

```
HIP SHOW JOBS [parameter]
```

These commands display the current status of the selected jobs in the HIP internal job queue, as well as the current status of the HIP processor itself. The format of the display depends upon the parameter given.

A display in the response to a SHOW JOBS request is called a screen. The number of jobs that can be displayed in a screen is limited to the number of lines available on the operator control console. Allowing two lines for the HIP status line and the column header, and two lines for the HP1110 End of job display and OS1000 READY FOR COMMANDS messages, there are 20 lines available for job status information in a screen.

SHOW JOBS PAGE display

The queue is divided into groups of 20 jobs called PAGEs, which are ordered from the oldest completed job to the newest job received. PAGE 1 consists of jobs 1 through 20 in the queues; PAGE 2 of the jobs 21 through 40, and so on.

The maximum page number is 6, corresponding to jobs 101 through 120, when the queue size is 120. If the queue size is less than 120, page numbers corresponding to the unused portions of the queue refer to the highest used page number. For example, if the queue size is 80, PAGEs 4, 5, and 6 refer to jobs 61 through 80.

A SHOW JOBS PAGE command always attempts to display a full screen of job information. If there are more than 20 jobs total in the HIP queue, SHOW JOBS REQUEST displays the entire queue. If there are fewer than 20 jobs in the page requested, one or more of the jobs from the previous page is included to make the display a full screen of 20 jobs.

PAGE 0 and PAGE with no parameter are exceptions because they produce a display screen of up to 20 jobs, starting with the most recently completed job. If there are fewer than 19 jobs after the completed job, additional completed jobs are included in the display.

Job queue display

The job queue display screen consists of two lines of header information, up to 20 lines of jobs status information, and a message to indicate the end of the display. If there are fewer than 20 jobs in the selected state, only that number will be displayed. If there are more than 20 jobs in the selected state, all jobs are not displayed, and the following message is displayed:

```
HP2145 Enter 'HIP SHOW JOBS CONTINUE' for more.
```

The abbreviated command HIP SJC can be used in place of HIP SHOW JOBS CONTINUE to continue the display. In this case, HIP SHOW JOBS CONTINUE continues to display.

The format of a normal display screen is shown in figure 8-1.

Figure 8-1. **Screen display for HIP SHOW JOBS CONTINUE**

```

HOST:hst:linemode HIP:runmode FILEMODE:filemode
JID# HOSTNAME RECEIVED STATUS JOB# TIME
nnnn hostname hh:mm:ss status 0000 hh:mm:ss
HP1110 END OF JOB DISPLAY

```

Table 8-6 describes the fields of the screen display.

Table 8-6. **Data provided by HIP SHOW JOBS**

Parameter	Description
hst	Specifies one of the possible hosts (XPF or XPS).
linemode	Specifies the status of the host interface (OFFLINE, ONLINE, PEND, READY, RECEIVING, SENDING, or SUSPENDED).
runmode	Specifies the jobs run mode (STARTED, STOPPED, DUMP, or CONCATENATED).
filemode	Specifies the acceptable file types (ALL JOBS or PRINT JOBS).
nnnn	Specifies the internal job identification number (JID) for HIP.
hostname	Indicates the host name on the file label.
0000	Specifies the job number (JN) assigned by OCS when the job is submitted for printing. It is 0000 for nonprint and pending jobs.
hh:mm:ss	Specifies the time a job was received and when its status was last changed.
status	Specifies the current status of a job.

Table 8-7 lists the processing job status

Table 8-7. **HIP job processing status**

Status	Description
RECEIVE	The job is currently being received.
SENDING	The job is currently being sent.
PENDING	The job is waiting to be sent.
QUEUED	The job is in OCS job queue waiting to be processed.
ACTIVE	The job is being processed by input and output.
OQUEUED	Input phase is complete and the job is queued for printing.
PRINTING	Input phase is complete and printing is in progress.
TRUNCATED	The transfer was truncated.
COMPLETED	The job was completed normally.
STORED	The file was stored in a permanent file directory.
SENT	The file was successfully sent to the host.
ABORTED	The job was aborted with the host command.
HOSTABORT	The job was aborted with the host command.
CANCELLED	The transfer was cancelled by a host restart.

SUMMARY job queue display

The SUMMARY job queue display screen consists of two lines of header information, up to 20 lines of job status information, and a message to indicate the end of the display. If there are fewer than 20 jobs to be displayed, a normal job queue display is given.

Figure 8-2 illustrates the display if there are more than 20 jobs in the queues.

Figure 8-2. **Summary display for HIP SHOW JOBS**

HOST:hst:linemode	HIP:runmode	FILEMODE:	filemode
JID# HOST NAME	RECEIVED	STATUS	JOB# TIME
nxxx<host name>	hh:mm:ss	COMPLETED	0000 hh:mm:ss
.	.	COMPLETED	.
.	.	COMPLETED	.
.	.	COMPLETED	.
.	.	STORED	.
...	and n more	COMPLETED	jobs.
.	.	PRINTING	.
.	.	QUEUED	.
.	.	QUEUED	.
.	.	QUEUED	.
.	.	QUEUED	.
.	.	QUEUED	.
...	and n more	QUEUED	jobs.
.	.	ACTIVE	.
.	.	PENDING	.
.	.	PENDING	.
.	.	PENDING	.
.	.	PENDING	.
.	.	PENDING	.
...	and n more	PENDING	jobs
.	.	RECEIVE	.
HP1110 END OF JOB DISPLAY			

There are 5 lines dedicated to jobs in the COMPLETED state, 8 lines dedicated to jobs in the ACTIVE state, and 7 lines dedicated to the jobs in PENDING stage. If there are fewer than this number of jobs in a given state, all jobs are displayed, and the entire display is less than 20 lines. If there are more than this number of jobs in a given state, the line which indicates the specified number of jobs displays the number of jobs that are not displayed.

HIP SHOW VERSION

This command displays the version and the system generation information for the HIP subsystem. The following message is displayed:

```
HP1190 HIP Version vvvv;gg
```

Syntax HIP SHOW VERSION

HIP START

Use this command to start or restart processing of jobs in the HIP queue.

The XPF host may submit jobs in a pass-through mode, where the job passes directly to the LPS processing. Although HIP is not involved in receiving the job, you must start HIP in order for pass-through jobs to be processed. If a message appears indicating that a pass-through job is pending, enter the HIP START command. While the pass-through job is processing, HIP is in a suspended state. After the pass-through job has finished, HIP returns to the start state.

HIP remains in START mode until you enter a HIP STOP command except in the following situations:

- If you enter the START command with a job-id, HIP exits START mode after submitting the last job-id.
- If you enter the START command with the DUMP parameter, HIP exits START mode after submitting one job.
- If you abort a HIP job during Input or Output processing, or using the ABORT command, HIP exits the START mode.
- If you enter the RESET command, HIP exits the START mode.

Syntax HIP START [DUMP][*job-id*[,...]]

Parameters Table 8-8 shows the parameters of the HIP START command.

Table 8-8. **Parameters of the HIP START command**

Parameter	Description
No parameters	When you enter the HIP START command with no parameters, HIP submits all jobs in the queue in the order received (or the order set in previous HIP START commands).
DUMP	If you specify this parameter, the system processes the next job in the queue (or the job specified in the command string if you have specified one) using the START command parameters defined in the DUMPJOB = field of the HIP.LIB file. Only one job is submitted (either the first job in the queue or the first job-id specified in the command string). After the specified job is processed, HIP issues a STOP command.
<i>job-id</i>	Use this parameter to start only specified jobs within the HIP queue. After HIP has submitted the last job in the job-id list, HIP issues a STOP command.

Procedure If the optional DUMP parameter is specified, the specified job or the next job in the queue is submitted with the START command parameters specified by the DUMPJOB = command in the HIP.LIB file. After the job is submitted, HIP effects the HIP STOP command. See also the "HIP library commands" section of this chapter.

If any listed jid is not eligible, it will be listed in the message:

```
HP1150 Cannot start job(s): | [,...].
```

HIP may be taken offline temporarily while HIP START is invoked. Should this occur, processing will automatically continue where it left off when HIP is put online again; that is, files will be immediately accepted from the host and printed from HIP's internal job queue.

If printing is to remain stopped while the printing system is taken offline and brought back online, the HIP STOP command must be entered.

Restriction When HIP submitted jobs are running, the commands START and SAMPLE (with parameters) are not allowed.

HIP STOP

This command causes HIP to stop submitting jobs to OCS for processing. If jobs are being processed in job concatenation mode, a HIP DRAIN is effected. Jobs already submitted to OCS for processing are not affected and are processed.

If the system is online, it continues to accept and queue jobs until it runs out of buffering or queuing space. One or more of the following messages is displayed:

```
HP0100 STOPPED ONLINE.
HP0110 STOPPED OFFLINE.
HP2100 HOST file buffer full. Enter 'HIP START'.
```

Syntax HIP STOP

Procedure Concurrent operation of HIP and online subsystems functions as follows:

The LPS can receive and print online (3211/4245 channel) jobs while HIP receives jobs in background processing. When online jobs complete, OFFLINE followed by HIP START sends the queued HIP jobs to print processing. The following commands are used to activate and deactivate these subsystems:

- ONLINE activates the online subsystem.
- HIP ONLINE activates HIP.
- OFFLINE deactivates the online subsystem.
- HIP END takes HIP offline so that ONLINE can be activated.

HIP library commands

A library (HIP.LIB) of stored parameters can be created that will automatically configure HIP when it is loaded. The HIP task uses the control file HIP.LIB to define parameters and select options that control the operational characteristics of the task. These include, but are not limited to, the JDE and JDL parameters of the START commands sent to OCS to process print jobs, the sizes of the internal job queue and buffering file, and the processing of non-print files.

The file is interpreted and acted upon only when the HIP task is loaded into memory. If HIP re-initializes itself, as following a sysgen, it applies the defaults indicated in this section if a command or the file is missing. Thereafter, for normal task loads, explicit commands with new parameter settings must be given if a parameter is to be changed from its previous setting.



Note: Only the following commands are accepted when there are jobs in the PENDING state: ACCEPT =, CONCATENATE =, and DUPLICATEFILE =. The other commands alter the pending job's processing parameters.

For most commands there are defaults defined in case that command cannot be used because it has an error, it is not present in the file, or the file does not exist. If an option is not wanted, or the default is wanted, the command line can be omitted.

The commands in the control file are keyworded commands; they may be in any order. The first three characters are unique, and can be specified by using just the first three characters. Optionally, they can be spelled out with any number of the additional characters, as long as the spelling is correct. Spaces between words in commands with more than a one-word keyword are optional, but they are recommended for clarity. Lines that are entirely blank are ignored. Lines whose first non-blank character is a semicolon are comment lines, and will also be ignored.

Commands to specify parameters are described below.

ACCEPT

This command specifies the file types that HIP accepts for permanent storage.

Syntax ACCEPT=*typ*[,*typ*,...]

Acceptable *file-types* are CMD, CME, DAT, FIS, FNT, FRM, FSL, ICT, IDR, IMG, JDL, JSL, LIB, MSC, LGO, PCH, PDE, STK, TMP, TST, and XCS.

Unacceptable *file-types* are SYS, \$Y\$, TPF, TSK, SAF, OSD, and LOG.

If the *file-type* is not listed as part of the ACCEPT command, it is rejected, and the following message is displayed:

```
HP2200 Rejected non-print file transfer
```

If the *file-type* or the line is missing, it defaults to ACCEPT=CMD, FSL, JSL, TMP, MSC, XCS, JDL.

XPAF hosts require access to file types not listed in the default HIP.LIB ACCEPT command. These file types are .FNT, .FRM, and .LGO files. It is best to code these file types at the beginning of the command.

Example ACCEPT = FNT,FRM,LGO,IMG,CMD,JSL,FSL,TMP,XCS;

Restriction It is the user's responsibility to modify the list in the HIP.LIB file to specify acceptable file types.

ACFILES

To configure the number of XPS accounting files that can be stored on the printer, the following entry must be made in the printer-resident HIP.LIB file:

Syntax ACFILES = *number*

Parameter Table 8-9 shows the parameter of the ACFILES command.

Table 8-9. **Parameter of the ACFILES command**

Parameter	Description
<i>number</i>	The decimal number (minimum of 1, default of 32, and maximum of 64) of accounting reports to be stored in each accounting file.

Limitations If the number entered is less than the minimum or greater than the maximum, the default of 32 will be enforced.

One accounting report (entry) requires 512 bytes (one sector), which means, the number of reports per file also specifies the file size (in sectors).

ACTREPORTS

To configure the number of XPS accounting reports (entries) per accounting file, the following entry must be made in the printer-resident HIP.LIB file:

Syntax ACTREPORTS = *number*

Parameter Table 8-10 shows the parameter of the ACTREPORTS command.

Table 8-10. **Parameter of the ACTREPORTS command**

Parameter	Description
<i>number</i>	The decimal number (minimum of 1, default of 32, and maximum of 64) of accounting reports to be stored in each accounting file.

Limitations If the number entered is less than the minimum or greater than the maximum, the default of 16 will be enforced.

One accounting report (entry) requires 512 bytes (one sector), which means, the number of reports per file also specifies the file size (in sectors).

BLOCKSIZE

Use this command to define the maximum amount of data the host can transfer in 512-byte disk sectors. The transmission of data from the host is optimized if the host block size is a multiple of 512 bytes. If the block size is not in multiples of 512, this parameter must specify a size greater than the block size to allow HIP to allocate buffers. If the blocks are in multiples of 512 bytes, this parameter can specify a larger buffer to allow the host to send multiple blocks. The default is 1 if the line is missing or is greater than 4. The default value for the host is 1.

For the XPAF/XPS host, the block size is an index as follows:

- 1 represents a 512-byte buffer
- 2 represents a 1024-byte buffer
- 3 represents a 2048-byte buffer
- 4 represents a 4096-byte buffer.

Syntax BLOCKSIZE = *n*;

Parameter Table 8-11 shows the parameter of the BLOCKSIZE command.

Table 8-11. **Parameter of the BLOCKSIZE command**

Parameter	Description
<i>n</i>	XPF or XPS hosts, the parameter should be set to 3 or 4.

BUFFERSIZE

Use this command to define the size of the file which temporarily buffers print data for print jobs.

Syntax BUFFERSIZE = *n*;

Parameter Table 8-12 shows the parameter of the BUFFERSIZE command.

Table 8-12. **BUFFERSIZE command parameter**

Parameter	Description
<i>n</i>	Specifies the size of the file in the range of 1024 to 65504 (values greater than 65504 give unexpected results). If the size specified is not a multiple of 32, it is truncated to the next smaller multiple of 32 to optimize HIP's disk utilization.

The default is 2048 if this line of the HIP.LIB file is missing, or *n* is less than 1024. HIP may allocate a smaller file if there is insufficient free space in the file system to allocate the specified size. In such a case, it allocates to the largest possible file and the HP1130 Buffer file allocation less than specified is displayed.

The size of the buffer file can be changed whenever the system is idle and there are no jobs in the HIP queue. HIP interrogates this specification each time it is manually loaded and reallocates the buffer file if the file either does not exist, or is empty, or is a different size than specified.

If there is insufficient disk space for even the minimum size spooling file, HIP cannot run. In this case, the current host is cleared as an available host. HIP should be ended immediately and disk space made available. The message HP2120 Insufficient disk space for HOST buffering is displayed.

CONCATENATE

Use this command to specify how jobs are submitted to the OCS for processing by the output and input processors.

Syntax `CONCATENATE = YES | NO, OPRINFO=YES|NO;`

Parameters The parameter is either YES or NO. If NO is given, each host job is submitted to OCS as a separate job. If the parameter is YES, HIP concatenates host jobs into reports as part of a larger job. A special concatenated job is started through OCS. If the concatenated job is defined by the SCI/JDE/JDL parameters, it becomes the next report of the current job. When a job with different SCI/JDE/JDL parameters is encountered, the current concatenated job is completed and a new one is started. This mode of input and output processing provides better performance.

Following the `CONCATENATE =` parameter is the optional subcommand `,OPRINFO =`, which has a parameter of YES or NO. The subcommand has meaning only if `CONCATENATE = YES`. It instructs HIP to display or not to display HIP information when a concatenated report changes state.

The default is `CONCATENATE = NO, OPRINFO = NO`.

While printing HIP jobs in concatenate mode with the job display mode ON (`HIP SHOW JOBS`), the `OUTPUT STARTED` message may be seen twice for an individual job. This occurs only if the printer is cycling up at the time this job is started by output (typically the first job that causes the printer to cycle up). This is an extraneous message and does not indicate that the job will be printed twice.

DRAINTIMER

Use this command to specify the number of seconds HIP is to delay before terminating the concatenated job.

Syntax DRAINTIMER = *n*;

Parameter Table 8-13 shows the parameter of the DRAINTIMER command.

Table 8-13. **Parameter of the DRAINTIMER command**

Parameter	Description
<i>n</i>	Can be any value from 0 to 65504. A zero value terminates the job immediately, if there is no job pending. The default for this command is DRAINTIMER = 30.

Concatenated jobs

In job concatenation mode, input processing waits for another job from the host when all jobs previously received from a host have been formatted. Some of the formatted data will not have printed, as it is less than the engine start-up threshold. The operator can print this data by entering the DRAINTIMER command, which terminates the current job and prints its data.

Concatenated jobs are terminated automatically, if no additional jobs are received from the host within the time interval specified by the DRAINTIMER command.

HOST

Use this command to specify user-defined host defaults that are to be used in processing HIP.LIB. It is automatically selected when the HIP task is loaded.

Syntax HOST = {XPF | XPS};

Parameters If only one host type is configured, that host type will be the default.

Table 8-14 shows the possible hosts for the HOST command.

Table 8-14. **Parameters of the HOST command**

Parameter	Description
XPF	Xerox printer access facility host for XPAF systems.
XPS	Xerox print service host for DocuPrintServer (formerly XPSM) systems.

LOGENTRIES

Specifies the maximum number of entries in a log file. The last entry of the log file marks the end of the file.

Syntax LOGENTRIES = *n*

Parameter Table 8-15 shows the parameter of the LOGENTRIES command.

Table 8-15. **Parameter of the LOGENTRIES command**

Parameter	Description
<i>n</i>	Specifies the number of entries in the range of 16 ₁₀ to 4992 ₁₀ . Specify this number as a multiple of 16.

If there is insufficient disk space, the system displays the following message and creates a smaller log file less than the one specified:

```
HP1170 Log file allocation less than specified
```

NAME

Use this command to specify the symbolic printer name that is returned upon host request. The default is NAME=XEROXHIP.

Syntax NAME = *printer-name*;

Parameter Table 8-16 shows the parameter of the NAME command.

Table 8-16. **Parameter of the NAME command**

Parameter	Description
<i>printer-name</i>	The symbolic printer name returned upon host request. The default is XEROXHIP.

QUEUESIZE

Use this command to specify the size of the internal job queue which HIP maintains.

Syntax QUEUESIZE = *n*;

Parameter Table 8-17 shows the parameter of the QUEUESIZE command.

Table 8-17. **Parameter of the QUEUESIZE command**

Parameter	Description
<i>n</i>	<p>Specifies the size of the queue in increments of 20 jobs, in the range of 20 through 120. If the value is not a multiple of 20, it is rounded up to the next multiple of 20; for example, QUEUESIZE = 55 will result in a queue size of 60. The queue size is adjusted on the first system load following a system generation or when the task is reloaded and there are no jobs in HIP's internal queues (such as following a reset). It is ignored upon HIP initiation following a system recovery and any time there are entries in HIP's queue.</p> <p>The default is QUEUESIZE = 20.</p>

Restriction Job numbering is reset only with the RESET or by rebooting the system controller. The queue is stored in the file HIPDR\$.SYS; each queue entry occupies one sector.

STARTCOMMANDS

Specifies one or more of the JDE/JDL parameters. Because the HIP task is responsible for processing Ethernet jobs, only documented HIP START commands should be used to start an Ethernet job. HIP file types HST and XDP cannot be used to start jobs with the ENET JDL. In order to selectively start jobs in the HIP queue, you should visually inspect the HIP job queue and manually start the desired jobs with the HIP START *jid, ..., jid* command.

STARTCOMMANDS is used in the START command which is sent to OCS when the HIP job is submitted for printing. Sixteen different pairs of JDE/JDLs may be defined. Each pair is identified by a START command index number, which is then used by other commands in the HIP.LIB file. Several of the indices are predefined to describe standard formats for various hosts.

Syntax STARTCOMMANDS = *sci*₁ :*jde*₁,*jdl*₁ ;*sci*₂:*jde*₂ ,*jdl*₂;...

Parameters Table 8-18 shows the parameters of the STARTCOMMANDS command.

Table 8-18. **Parameters of the STARTCOMMANDS command**

Parameter	Description
<i>sci</i>	A START command index in the range of 1 to 16.
<i>jde</i> and <i>jdl</i>	The job descriptor entry and the job description library which describe the format of the data.

As many multiple *sci:jde,jdl* parameter sets can be given on a line as will fit. Multiple command lines can reside in a file. When an SCI is specified more than once, the last one is used.

Defaults The following commands are used as the default if the needed command, or the file itself, is missing:

```
STARTCOMMANDS = 1:HIP ,HIP;
STARTCOMMANDS = 8: ,ONLINE;
STARTCOMMANDS = 13:EBCDIC,HIP; 14:ASCII,HIP;
15:HEXDMP,HIP;
```

For each possible host, there is a command that specifies the STARTCOMMANDS index which is used to submit jobs from that host. The format of the command is *xxxJOB=SCI*, where *xxx* is the host name as specified in the HOST command. The defaults are:

Host	Start command
XPF	XPFJOB = 7
XPS	XPSJOB = 1

XPAF host jobs that are to be printed in the pass-through mode are started with the start command index (SCI) specified by the PASSTHROUGHJOB= command. For the pass-through mode to function, HIP must be in the START state. Also, the HIP online interface goes into a suspended state whenever a pass-through job is received. The default is:

```
PASSTHROUGHJOB = 8 XPAF jobs to be processed in the
online processing mode
```

Jobs started with the command START DUMP JID are processed in dump mode, not as user data. The DUMPJOB command specifies the STARTCOMMANDS index JDE/JDL to be used.

The defaults for these commands are as follows:

DUMPJOB = 15 Jobs started from the HIP START DUMP *jid*

- Restrictions**
- Use only documented HIP START commands to start an Ethernet job.
 - Do not use HIP file types HST and XDP to start jobs with the ENET JDL.
 - The default JDE/JDL for XPAF pass-through jobs is DEFAULT. To use the correct JDE/JDL for the current version of XPAF, you may have to edit the HIP.LIB file from ONLINE to DEFAULT in the STARTCOMMANDS index.
 - The default JDE/JDL for XPS pass-through jobs is DFLT, XPSM. To use the correct JDE/JDL for the current version of XPS, you may have to edit the HIP.LIB file from ONLINE to DocuPrintServer (formerly XPSM) in the STARTCOMMANDS index.

XPFJOB

Use this command to specify the START command index defined in STARTCOMMANDS used for XPF jobs.

- Syntax** XPFJOB = *n*;
 The default is XPFJOB = 7

XPSJOB

Use this command to specify the START command index defined in STARTCOMMANDS used for XPS jobs.

- Syntax** XPSJOB = *n*;
 The default is XPSJOB = 1.

Nonprint transfer file control

The acceptance for nonprint files for storage is controlled by the current host, the system logon level, the ACCEPT command, and the DUPLICATEFILE command.

The file types that are accepted can be changed whenever the system is idle and there are no active jobs in the HIP queues. Each time HIP is loaded, it regenerates its lists of file types accordingly.

The file types that are never accepted are \$\$\$, SYS, TSK, SAF, OSD, TPF, and LOG.

Care must be taken in transferring nonprint files interspersed with print files. Once font, form, logo or other types of data files are referenced by a print job, their physical disk addresses, rather than their names, are associated with the job. If one of these files is replaced by a nonprint file transfer, it is indeterminable what data will actually be used during printing; it could be the original file, the new file, part of each, or some other file if more than one nonprint file was transferred.

File recovery

There are two types of file recovery.

Crash HIP recovery

The DocuPrint 180 LPS recovers the HIP JOB queue after a system crash (such as a CTRL V) because the OSEXEC task saves all memory previous to the crash. All data is saved.

No boot or power failure HIP recovery

The OSEXEC task does not service a reboot or a power failure. This means the DocuPrint 180 LPS loses the HIP JOB queue after a boot or power failure.

You can save all HIP data before rebooting by entering the following commands (HIP END, HIP OFFLINE) before the boot command.

9. Graphics on the LPS

The Advanced Imaging Subsystem (AIS) enables a laser printing system to image encoded digitized graphic images in addition to other text and forms.

LPS-resident graphic software supports input from:

- Open-reel tapes with fixed or variable record/block structures (such as RECORD STRUCTURE = F, FB, V, or VB only) in an unpacked format, which are compatible with use on the LPS
- IBM online host systems equipped with the XPPI-XDGI software package or equivalent.

A job with complex graphic output may require modification to run on the Xerox DocuPrint 180 LPS if the job was coded for a 4050F LPS which runs V3.5.

Input for graphics

Named graphics may be input to the LPS with open-reel tape or the online channel. They may be input either separately from or interleaved with textual data. If input separately, they are recorded as permanent disk files. If interleaved with textual data, they may or may not be recorded as permanent disk files. (Refer to the "Processing modes" section for a discussion of this operation.)

A graphic is typically incorporated into a document by a DJDE record occurring with the data targeted for the same simplex or duplex page on which the graphic appears. Additionally, a graphic may be referenced in a form, so that use of the form automatically includes the graphic.

Processing modes

A graphic is handled by the software in one of the following four modes:

- Random
- Move
- Block
- Batch.

The user selects the mode depending upon how the graphic is made available for report processing, as well as user performance requirements and disk resources.

Random mode

In Random mode, a graphic is obtained from a named disk file. This occurs if the file already exists at the start of report processing or is read from the report's input data stream before the first record of the report text is processed. In the latter case, each graphic is written to the disk as a permanent .IMG file. In Random mode, the output software reads the various graphics for a page from their individual files into graphics memory.

Move mode

In Move mode, a graphic is obtained from a named disk file (just as in Random mode), but the graphic is copied into the print file and used as in Block mode. This feature may be invoked by the GRAPHICS command of the OUTPUT PDL statement. (Refer to the "OUTPUT statement" section in the *Xerox DocuPrint 180 LPS PDL Reference*.)



Note: Under some circumstances, Move mode graphics are overridden and treated as Random mode. In particular, if for any page there is exactly one graphic and that graphic is on a disk drive other than the print file's disk drive, Random mode is invoked.

Block mode

In Block mode, a graphic is obtained from the input data stream immediately following the textual data to be imaged on the same page (refer to the "Page interleaved" section of this chapter). The graphic is copied directly to the print file adjacent to the page's formatted textual information. Separate graphic disk files are not created. In Block mode, the output software reads graphics from the print file into graphics memory. Block mode graphics exist on disk only in the print file and only while the report is being processed. No disk file is created.

A given page of a report which includes Block mode graphics may also include either Random mode graphics or Move mode graphics.

Batch mode

In Batch mode, the input stream contains graphics which are to be imaged, one per page, with either default or optional imaging specifications. Batch mode graphics are copied to the print file and processed the same as Block mode graphics. In Batch mode, only one graphic is imaged on each page (for example forms may not be invoked in Batch mode). This feature may be invoked by the GRAPHICS command of the OUTPUT statement. Refer to the *Xerox DocuPrint 180 LPS PDL Reference* for more information.

Tape formats

This section describes the tape formats used for transferring files and reports.

Noninterleaved

To transfer .IMG files from tape to an LPS system disk independent of printing, the COPY TAPE LABEL *file-name* command may be used. Refer to the "Editing source files" chapter for more information.

Document interleaved

In this format, a report's data stream begins with a set of one or more graphics prior to the first record of text. These graphics are copied one at a time to named .IMG disk files. When the report has finished printing, the .IMG files created are automatically deleted, unless the user has explicitly specified otherwise.

The rules for formatting a document interleaved graphic report tape file are:

1. The tape format may be any legal fixed or variable length block, or record format which is otherwise supported by the LPS (such as RECORD STRUCTURE = F, FB, V or VB). It may not be an undefined block or record format (such as RECORD STRUCTURE = U or UB). Packed data formats are not supported.
2. Use of block and record constants in an appropriately structured file (type F, FB, V, or VB) containing interleaved graphics is supported, but the actual delimiter searching is suspended while an .IMG file is read.
 - a. For record delimiters, this means that delimiter searching is suspended from the first record of graphic data until the entire graphic has been read. Extraneous bytes at the end of the last record are ignored. Record delimiter processing resumes with the next record.
 - b. For block delimiters, this means that the first byte of a graphic must be the first data byte of a block and that the first byte of data following the graphic must also be the first byte of a block. Extraneous data in the last block following the last byte specified by the IMG byte count are ignored.
3. Each graphic must be immediately preceded by a GRAPHIC DJDE.
4. Improved tape-to-disk processing time is obtained if the graphic data portion of each record is an integer multiple of 512 bytes and begins at an even byte offset from the beginning of the block. Processing of each graphic begins in this mode. If a record is not a multiple of 512 bytes or does not begin at an even byte offset from the beginning of the block, the mode is terminated.

Page interleaved

In this format, one or more graphics imaged on a physical page are included in the data file immediately following the textual data for that page. These graphics, if referenced, are processed in Block mode (such as copied one at a time directly to the print file adjacent to the page's formatted textual data). If not referenced, a page interleaved graphic is skipped. The rules for page interleaved graphics are as follows:

1. The graphics must occur on the tape immediately following the last text to appear on the same physical page.
2. Rules 1 through 4 for document interleaved graphics apply (refer to the "Document interleaved" section).

Batch mode

In this mode, a report contains only graphics imaged one per page and an optional IMAGE DJDE preceding each graphic. These graphics are copied to the print file and processed as Block mode graphics.

Rules 1 and 4 for document interleaved graphics apply (refer to the "Document interleaved" section). Note that block or record delimiters are not permitted.

Except for the graphics themselves, the input stream may contain only optional DJDE IMAGE records. When a DJDE IMAGE record is encountered, it defines the scaling and positioning parameters for subsequent graphics.

Online formats

This subsection describes the online tape formats.

Noninterleaved

HOSTCOPY is the mechanism for transferring noninterleaved graphics with the online channel to disk. The HOSTCOPY utility supports .IMG files in Xerox system tape format.

Document and page interleaved

Online report processing, which includes document and page interleaved graphics, functions as tape input processing functions. The rules for online interleaved graphics are:

1. Each graphic must be immediately preceded by a DJDE GRAPHIC record.
2. There is no change in the maximum record length when processing interleaved graphic records, but every record of an online interleaved graphic must have an appended terminating character which is the same for all records of the graphic. If an

interleaved graphic record is encountered which does not have the same terminating character as the first record of the graphic, the user has the option of aborting or continuing the report. If the report is continued, the graphic will be unprintable and the results may be unpredictable.

3. Document interleaved graphics must begin immediately following the banner pages.

Batch mode

Online processing of Batch mode reports, except for media-imposed differences, is the same as for open-reel tape (refer to the "Batch mode" subsection of the "Tape formats" section in this chapter).

The input data stream format must comply with the following rules:

1. Even though a JDE specifying Batch mode is in effect, the first graphic must be preceded by a DJDE specifying BATCH=START. The last graphic must be immediately followed by a DJDE specifying BATCH=END. These Batch stream delimiters are necessary to distinguish banner page data (header and trailer) from graphic data.
2. Except for the graphics themselves, the input stream may contain only optional DJDE IMAGE records. When a DJDE IMAGE record is encountered, it defines scaling and positioning parameters for subsequent graphics. The actual batch mode data stream is defined by the BATCH=START and BATCH=END DJDEs.
3. There is no change in the maximum record length when processing Batch mode graphic records, but every record within the graphic must be terminated by a byte containing an appended terminating character which is the same for all records of the graphic. If a Batch mode graphic record is encountered which does not have the same terminating character as the first record of the graphic, the user has the option of aborting or continuing the report. If the report is continued, the graphic is unprintable and the results may be unpredictable.

Document interleaved graphic file transfers

The document interleaved format copies graphics down the channel from the host to LPS .IMG files. This is done by running a document interleaved report containing a trivial amount of text using a JDE whose OUTPUT statement specifies graphics with PURGE=NO. This function is available for offline use also.

Management of image files

To produce print files consisting of interleaved textual and imaginal data, the LPS file id must be saved in ASCII in the LPS 128-byte file label.

Scaling

Scanned graphic images can be scaled to print at an enlarged (or reduced) size subject to certain constraints. When a graphic image file is created by a raster encoding scanner, the image file header is required to contain the resolution (such as number of scan lines per inch) at which the image was scanned. Each picture element (pixel) in the raw or unencoded rasters is assumed to be isometrically formed (such as equally dimensioned in the x and y directions). The imaging hardware scales the graphic images by representing each pixel in a graphic image file by one or more spots in the printed image. The scale factor applied by imaging hardware is calculated as follows:

$$S_e = S_r \times (R_p \div R_i)$$

where

S_e = effective scale factor (rounded to nearest integer value:
1£ S_e £8)

S_r = reference scale factor (default value = 1)

R_p = print job resolution

R_i = image resolution

For example, if S_e is computed to be =3, each pixel in a graphic image file will be represented by a 3 by 3 matrix of 9 spots. The reference scale factor (optionally specified when the graphic image is referenced by DJDE) is used to describe a size relative to the original graphic. The R_p/R_i ratio serves to make the image file and the reference scale factor virtually resolution independent.

It should also be noted that graphic images may be scaled to print at a reduced size only when the R_p/R_i ratio is greater than 1. For example, a graphic image file created with an image resolution of 150 scan lines per inch can be scaled to print at one-half the size of the original graphic by specifying a reference scale factor of 1/2 in a 300 spi print job. Also, a graphic image scanned at 240 scan lines per inch will print at 4/5ths the size of the original, even though a reference scale factor of 1 is specified in a 300 spi print job.

If a scaled graphic image is too large or is positioned to print outside the page boundary, input reports an error, and the graphic is not imaged.

PDL graphics options

Several PDL commands are available for graphic handling. These include four job descriptor entry (JDE) OUTPUT statement commands and six dynamic job descriptor entry (DJDE) commands.

Tables 9-1 and 9-2 summarize these commands. The JDE OUTPUT statement commands and the DJDE commands are further described in the *Xerox DocuPrint 180 LPS PDL Reference*.

The DJDEs are record oriented because they take effect immediately. For example, the ALTER, CANCEL, and IMAGE commands may appear anywhere on the page and are applied to the page on which they occur. The GRAPHIC, BATCH, and SAVE DJDE commands also take effect immediately.

Page-oriented DJDEs applied to the next page must not precede page interleaved graphics to be applied to the current page.

Table 9-1 summarizes the JDE OUTPUT statement commands.

Table 9-1. **JDE OUTPUT statement commands**

Command	Function
GRAPHICS	Specifies graphics may be used in current job.
IMAGE	Specifies initial scaling and positioning parameters for Batch mode.
PURGE	Specifies default disposition of document interleaved graphics.
UNITS	Specifies user-selected units for positioning graphics.

Table 9-2 summarizes the DJDE commands.

Table 9-2. **DJDE commands**

Command	Function
ALTER	Specifies new imaging parameters for graphics previously referenced using HOLD statement.
BATCH	Marks beginning and end of online Batch mode graphic data.
CANCEL	Terminates HOLD statement.
GRAPHIC	Precedes interleaved graphic in data stream.
IMAGE	Specifies name, scaling, and positioning of a graphic printed.
SAVE	Identifies document interleaved graphics not to be deleted (purged)

FDL GRAPHIC statement

Use this statement to associate one or more Random mode graphics with a form so that imaging the form includes imaging the associated graphics.

Graphics referenced by forms are invoked only in reports for which graphics processing has been previously specified by the OUTPUT statement GRAPHICS = YES or MOVE option. If graphic processing has not been specified, these graphic references are ignored and not considered an error condition. (Form sampling implicitly specifies graphic processing.)

One to 16 graphics may be associated with a single form (over 16 with Interpress and RIP options). Each graphic to be associated with a form must be specified in the form's .FSL file by a statement with the following syntax:

Syntax GRAPHIC *name* AT *vpos* [*cm* | *in* | *dot*] , *hpos* [*cm* | *in* | *dot*] [SCALE[S]
n [*d*]]

Parameters The parameters, except for positioning precision, are the same parameters used by the DJDE IMAGE command. In FDL, the precision of any position specification is two rather than three places to the right of the decimal point. User-specified positioning units (UN) are not available in FDL. Standard FDL grid specification and unit defaults apply. A form may consist solely of graphics.

Please refer to the *Xerox DocuPrint 180 LPS PDL Reference* and to the *Xerox DocuPrint 180 LPS Forms Creation Guide*.

Operational considerations

This section identifies some operational considerations.

Recovery

Recovery is the same as for standard LPS processing and includes graphic data as well as textual and form data.

Accounting

The accounting sheet includes four graphics-oriented fields.

Table 9-3 shows the graphics fields.

Table 9-3. **Graphics-oriented fields**

Field	Description
GRAPHIC PAGES PRINTED	A count of all pages containing at least one graphic impression delivered to the tray.
GRAPHIC IMAGES READ	A count of all graphics read in the input stream.
GRAPHIC COMPLETION CODE	A composite code identifying the types of abnormal graphic processing conditions encountered by the input processing task. Refer to table 9-4.
GRAPHIC IMAGES MOVED	Total count of the times one or more images were copied from one disk location to another (for example Move mode, page interleaved with HOLD).

Graphic messages

The following explanation of messages is provided to help the user monitor the progress of the job and identify and correct any errors or difficulties encountered:

- A summary graphic completion code appears on the accounting sheet.
- DJDE errors are reported on a printed error sheet.
- A list of .IMG files created or replaced is printed. Those files not purged are marked with an asterisk.
- A list of numbers of those pages having an input task detected error is printed along with a code for each page identifying the error(s) detected.

Table 9-4 lists the keys for interpreting the codes.

Table 9-4. **GRAPHIC completion code**

Code	Condition
1	Referenced graphic not found.
2	Graphic positioned off page—not printed.
4	Too many graphics specified for page.
8	Warning only—graphic data for page exceeds one megabit.
16	Page interleaved graphic not referenced.
32	CANCEL or ALTER of graphic not held.
64	Graphic data for page exceeds graphic memory—excess not printed.
128	Illegal .IMG file header format.
256	Copy-sensitive form illegally referenced a graphic.
512	Document interleaved graphic sentinel contains positioning information.
1024	Online interleaved or batch graphic record not terminated by delimiter character.
2048	Permanent file saturation with no purges pending (such as no space on disk for document interleaved graphic).

It should be noted that combinations of codes can occur. If combinations of codes do occur, they are added together into a unique code, which must be decoded by the user. For example, a code of 11 would indicate exception codes 1, 2, and 8.

Disk saturation

Two kinds of disk saturation are possible during report processing. The first can occur at the beginning of a report while a set of document interleaved graphics (refer to the “Document interleaved” section in this chapter) are being copied from the input stream to disk files. If the disk becomes saturated during this process, the operator is notified. If there are graphic file purges pending the completion of printing of a previous report, the copying process waits for the file purge before continuing. If there are no graphic file purges pending, the operator is asked to specify that the report be either aborted or continued without the graphic currently being read.

The second kind of disk saturation, print file saturation, can occur in multiple copy reports when the print file is not large enough to contain the entire report. This may occur in Block, Batch, and Move modes where the graphics are copied into the print file. Standard LPS software allows the user to specify (with the PDL RSAT command) what to do in the event of print file saturation offline. Use of the RSAT=SPLIT command is recommended since the reprocessing required when the REMOUNT command is requested will significantly degrade throughput because of the time it takes to move the large amount of data that graphics represent.

Imaging dump utility

The purpose of the Image Dump utility (IMD) is to fix, display, or print the content of IMG files in compressed and decompressed format. The Xerox DocuPrint 180 LPS is designed to handle RES format image files. RES is stricter than the format supported by older version systems (such as, 4135 Release V3.5 or 3.6 with GVG). This means, not all images that work on older systems will work on Xerox DocuPrint 180 LPS.

Image Dump or Print

The image dump or print tools displays or prints, respectively, the content of IMG files in compressed and, if specified, decompressed format. These tools only display or print Interpress 1.0, 2.0, and RES format image files.

Syntax IMAGE {DUMP | PRINT} *filename.filetype* [(*scanline start,scanline end*) [/EXPAND] [/STATISTICS] [/VERBOSE]

Parameters Table 9-5 shows the parameters of the IMAGE DUMP or PRINT command.

Table 9-5. **Parameters of the IMAGE DUMP or PRINT command**

Parameter	Description
DUMP	Displays the content of IMG file in compressed and decompressed format.
PRINT	Prints the content of IMG file in compressed and, if specified, decompressed format.
<i>filename.filetype</i>	Names an existing file or files in the .IMG or .IMG directory. <i>filename</i> may contain wild cards such as '*' matching 0 or more characters, or '?' matching exactly one character. Length should be at least one and at most six characters. The only valid <i>filetype</i> is IMG.
<i>scanline start,scanline end</i>	Specifies a range to dump or print the image data. Notice that <i>scanline start</i> and <i>scanline end</i> must be separated by one comma; any other delimiter is invalid.
/EXPAND	Displays or prints compressed scan line followed by decompressed scan line. Without this switch, no decompression will be performed on the compressed data.
/STATISTICS	Displays or prints the following statistics summary: <ol style="list-style-type: none"> 1. scan line length 2. total number of scan lines in image file 3. total number of ENC scan lines 4. total number of HTN scan lines 5. total number of LIN scan lines 6. total number of RAW scan lines 7. total number of EOI codes encountered 8. total number of bad nibbles 9. total number of bad scan lines 10. x pixel value 11. y pixel value 12. image file size (in bytes) 13. actual Compressed vector length 14. compressed vector length in header 15. compressed vector length in trailer
/VERBOSE	Displays or prints detailed diagnostic messages before and after compressed image data. All information in the image header and trailer is displayed in decimal value. This switch has no effect on RES format image files. In other words, for RES format file, no diagnostics message will be displayed.



Note: Optional parameters may start with a '/', or without. Also, any substring of the option command is valid. For example: '/VERBOSE', 'VERBOSE', 'VERBO', 'VER', 'V', '/V' are valid commands to indicate the 'VERBOSE' option.

Example 1 When IMAGE DUMP A.IMG is executed the following information is displayed one page at a time:

```
Image:a.img,break table:0,Range:8,Scan Length:300
Page 3
```

```
Xpixel:1024, Ypixel:300, resolution:(300/1)
```

```
Ln# :Bk# :Nb# :lbcode !Data
55 :2 :100 :HTN !5F62!4562!51FC!CEA1...
56 :2 :357 :RAW !7321!894E!998A!....
57 :3 :28 :LIN !3376!AAFC!56FF!9956!....
```

.

.

.

<Hit return to continue or (q/Q) to quit>

After the user presses <ENTER>, the next page will be displayed.

Example 2 When IMAGE DUMP A.IMG EXPAND is executed the following information is displayed one page at a time:

```
Image:a.img,break table:0,Range:8,Scan Length:300
Page 3
```

```
Xpixel:1024, Ypixel:300, resolution:(300/1)
```

```
Ln# :Bk# :Nb# :lbcode !Data
55 :2 :100 :HTN !5F62!4562!51FC!CEA1...
55 :2 :100 :HTN==> !4535FB3C!4D....
56 :2 :357 :RAW !7321!894E!998A!....
57 :3 :28 :LIN !3376!AAFC!56FF!9956!....
57 :3 :28 :LIN==> !C7DEC906!CE468FDE!....
```

.

.

.

<Hit return to continue or (q/Q) to quit>

After the user presses <ENTER>, the next page will be displayed.

Example 3 When IMAGE DUMP A.IMG VERBOSE is executed the following header information is displayed one parameter at a time; all values displayed are in decimal:

Figure 9-1. Header information

```

Begin value is <value>

<Hit return to continue or (q/Q) to quit>

Comment header is <value>

<Hit return to continue or (q/Q) to quit>

Xpixel value is <value>

<Hit return to continue or (q/Q) to quit>

Ypixel header is <value>

Ypixel value is <value>

<Hit return to continue or (q/Q) to quit>

Resolution header is <value>

Resolution numerator = <value>

Resolution denominator = <value>

<Hit return to continue or (q/Q) or quit>

R1 header is <value>

<Hit return to continue or (q/Q) to quit>

CPV header is <value>

compressed pixel vector is <value>

Break table is <value>

range is <value>

scan line length is <value>

start of image is <value>

```

Figure 9-2. Header information

```

<Hit return to continue or (q/Q) to quit>
.
.
.
Image:a.img,break table:0, Range:8, Scan Length:300 Page 3
Xpixel:1024, Ypixel:300, resolution:(300/1)
Ln#  :Bk#  :Nb#  :lbcode !Data
55   :2    :100  :HTN   !5F62!4562!51FC!CEA1...
56   :2    :357  :RAW   !7321!894E!998A!....
57   :3    :28   :LIN   !3376!AAFC!56FF!9956!....
.
.
.
<Hit return to continue or (q/Q) to quit>
.
.
.
Signature header is <value>
<Hit return to continue or (q/Q) to quit>
CPVL header is <value>
compressed pixel vector is <value>
<Hit return to continue or (q/Q) to quit>
END value is <value>
<Hit return to continue or (q/Q) to quit>

```

Example 4 When IMAGE DUMP A.IMG STATISTICS is executed the following summary data is displayed after image data dump:

Figure 9-3. **Summary data**

```

.
.
.Image:a.img,break table:0,Range:8,Scan Length:300Page 3
Xpixel:1024, Ypixel:300, resolution:(300/1)
Ln# :Bk# :Nb# :lbcodes !Data
55 :2 :100 :HTN !5F62!4562!51FC!CEA1..
56 :2 :357 :RAW !7321!894E!998A!....
57 :3 :28 :LIN !3376!AAFC!56FF!9956!....
.
.
.
<Hit return to continue or (q/Q) to quit>
.
.
.
***IMAGE statistics Summary:***
scan line length: 300
Total number of scan lines 1024
Total number of ENC: 300
Total number of HTN: 400
Total number of LIN 300
Total number of RAW: 24
Total number of EOI 1
Total number of bad nibbles: 0
Total number of bad scan lines: 0
xpixels: 1024
ypixels: 300
Total bytes count: 27392
Actual compressed bytes count: 26891
compressed pixel vector (header): 26891
compressed pixel vector (trailer): 26891

```

Image Fix

The image fix tool fixes the format of known bad IMG files which cause problems when imaging with the decompressor chip on the EAIPX board.

After an image file is fixed the following information is updated:

- Image file's compressed pixel vector length, both in the header and in the trailer section of the file.
- Xpixel value, number of scan lines in the image.

Syntax IMAGE FIX *filename,filetype* [/STATISTICS] [/VERBOSE] [/ALL]

Parameters Table 9-6 shows the parameters of the IMAGE FIX command.

Table 9-6. **Parameters of the IMAGE FIX command**

Parameter	Description
<i>filename.filetype</i>	Names an existing file or files in the .IMG or .IMG directory. <i>filename</i> may contain wild cards such as '*' matching 0 or more characters, or '?' matching exactly one character. Length should be at least one and at most six characters. The only valid <i>filetype</i> is IMG.
/STATISTICS	Displays or prints the following statistics summary: <ol style="list-style-type: none"> 1. scan line length 2. total number of scan lines in image file 3. total number of ENC scan lines 4. total number of HTN scan lines 5. total number of LIN scan lines 6. total number of RAW scan lines 7. total number of EOI codes encountered 8. total number of bad nibbles 9. total number of bad scan lines 10. x pixel value 11. y pixel value 12. image file size (in bytes) 13. actual Compressed vector length 14. compressed vector length in header 15. compressed vector length in trailer
/VERBOSE	Displays or prints detailed diagnostic messages before and after compressed image data. All information in the image header and trailer is displayed in decimal value. This switch has no effect on RES format image files. In other words, for RES format file, no diagnostics message will be displayed.
/ALL	Fixes all image files including RES format. Without the /ALL switch, no files will be fixed.



Note: Optional parameters may start with a '/', or without. Also, any substring of the option command is valid. For example: '/VERBOSE', 'VERBOSE', 'VERBO', 'VER', 'V', '/V' are valid commands to indicate the 'VERBOSE' option.

Example When the IMAGE FIX process is executing the following messages may be displayed:

```
<parameter> header corrected... <new value>
```

```
.  
.
.
```

```
Fixed (header) compressed pixel vector,  
( <value><value>)
```

```
Fixed xpixel value, ( <value>><value>)
```

```
Fixed (trailing) compressed pixel vector,  
( <value>><value>)
```

After the image file is fixed the following message may be displayed:

```
Image <filename>.<filetype> fixed
```

```
Image <filename>.<filetype> skipped; Xerox  
Interpress 2.0 Format
```

```
Image <filename>.<filetype> bad; suggest deleting.
```

Performance

This section outlines three performance considerations.

Random mode

Random mode requires a separate disk access by the output processing task for every graphic on a page. As more random mode is used, the set up time increases. This may result in more paper path holes.

Online

It should be noted that sending interleaved graphics across the online interface impacts system performance. The time to process graphic information across the channel, block it, and write it to disk is significant. As the average density of graphic information decreases, throughput should increase.

Document interleaved file creation

The time required to create (or replace) and subsequently purge document interleaved graphics is significant.

Restrictions

This section summarizes graphic restrictions, some of which have been mentioned elsewhere in this document. These restrictions are grouped according to whether they affect graphic features.

Graphic feature restrictions

Tape formats supported by interleaved graphic processing are restricted to LPS compatible fixed or variable record structures (such as RECORD STRUCTURE = F, FB, V, or VB only). Undefined length record structures (such as types U and UB) are not supported. Packed data formats are not supported.

The maximum number of graphics that may be imaged on one physical page is 16. This includes graphics associated with forms.

Graphic images may be overlapped on a page with each other and with text.

The maximum amount of graphic data which can be imaged on a simplex page is limited by the size of graphics memory. Throughput may be reduced if more than one megabit of graphic data is imaged on each page.

No online banner page (neither header nor trailer) may include graphic references.

A JDE which is selected with the DJDE option JDE= or JDL= may not specify GRAPHICS=NO if the current report has already processed at least one graphic.

In online Batch mode, PCC commands are ignored beginning with the BATCH=START DJDE until the BATCH=END DJDE.

Graphics processing is not copy sensitive. A copy-sensitive form may not reference graphics.

If RPAGE is used to move a logical page from one physical page to another, graphic references are handled as follows: Graphic references encountered after the record which satisfied the RPAGE statement are imaged on the physical page to which the logical page is moved. Any other image references are applied to the physical page being formed at the time the RPAGE statement was satisfied. For example, if RPAGE WHEN=TOP is specified, it is possible for a graphic to appear on physical page n even though text which may have preceded its reference in the input data stream appears on physical page n+1.

Use caution when using RPAGE with page interleaved graphics.

A document interleaved graphic replaces an existing graphic with the same name. This is true even if the existing graphic is used in a report which is being printed.

Forms are not supported for Batch mode processing.

This chapter contains the following sections:

- Compare V3C with other Xerox software versions
- Controlling stock input to print jobs
- Setting up output from print jobs

Compare V3C with other Xerox software versions

This section highlights how V3C performs in comparison with other Xerox software.

Compare job processing

This subsection outlines how print jobs run on V3C.

Identify how print jobs run on V3C

Print jobs created for LPS running software versions 2.1, 3.0, 3.5, 3.7, 3.8, 4.0, and V3A run well on the Xerox DocuPrint 180 LPS. Jobs created for the 4850 Highlight Color laser printing system print in blacks and grays. If the job includes a very light tint, the LPS may not be able to print the tinted areas. Two-color graphics cannot be printed on the Xerox DocuPrint 180 LPS.



Note: A job which includes 600 dpi fonts, forms, or logos can be printed only on the 4650 LPS; it cannot be printed on the Xerox DocuPrint 180 LPS.

For additional compatibility information refer to the *Xerox DocuPrint 180 LPS Product Reference*.

Compare file compilation

You can compile any job source libraries (JSLs) created on V3.5 or V3.7 on the Xerox DocuPrint 180 LPS.

Table 10-1 shows how form source files (.FSL files) created on an LPS running other OSS versions are compiled on an LPS running V3C.

Table 10-1. **How .FSL files created on other OSS versions compile on V3C OSS**

OSS	LPS	V3C result
V2	9790	Compile as on original OSS.
V3	4050	Compile as on original OSS.
V3.5	4050/4090	Compile as on original OSS.
V3.6	4135	Compile as on original OSS.
V3.6	4635	Compile as on original OSS.
V3.7	4850	Do not compile.

Compare OSS command execution

This subsection outlines how V3C handles OSS commands.

Identify how V3C handles other OSS commands

Table 10-2 shows how the OSS commands available for an LPS running other OSS versions are executed on a Xerox DocuPrint 180 LPS running V3C.

Table 10-2. **How OSS commands for other OSS versions execute on V3C OSS**

OSS	LPS	V3C result
V2	9790	Execute as on original OSS.
V3	4050	Execute as on original OSS.
V3.5	4050/4090	Execute as on original OSS.
V3.6	4135	Execute as on original OSS.
V3.7	4850	Execute as on original OSS unless the command is LIGHT, PQA, or SUBSTITUTE INK, in which case the LPS displays OS1726 This system is not configured for this command.

Converting Xerox DocuPrint 180 files to print on the 4650 LPS

Use the SCALE command to convert Xerox DocuPrint 180 files.

SCALE

Use this command to scale 300 spi fonts, forms, and logos to 600 spi equivalents. To scale fonts and logos, SCALE doubles all positioning metrics in the output file and employs the same set of interpolation rules used by the ROS to render the 600 spi character bit maps. The appearance of the fonts and logos produced by the SCALE and printed at 600 spi is indistinguishable from the appearance of the same unscaled 300 spi files interpolated by the ROS. Forms are scaled by changing all internal references to font, logos, and images in the form header to refer to FN6, LG6, IM6 entities.

Logon level Levels 1 through 5.

Syntax SCALE *input-filename.filetype1 output-filename.filetype2* [/b]

Parameters Table 10-3 shows the parameters of the SCALE command.

Table 10-3. **Parameters of the SCALE command**

Parameter	Description
<i>input-filename</i>	Specifies the font (.FNT), form (.FRM) or logo (.LGO) file to be converted to 600 spi.
<i>filetype1</i>	Specifies the file type as .FNT, .FRM or .LGO.
<i>output-filename</i>	Specifies the name for the new 600 spi file.
<i>filetype2</i>	Specifies the file type as .FNT or .FN6, .FRM or .FR6, .LGO or .LG6, .MSC or .TMP; usually "6" is substituted for the last character of the original file type.
/b	Doubles the font or logo bitmaps instead of using the interpolation formula to scale the input file.
/O	Overwrites the output filename without prompting.
/A	Overrides the escapement error message ** DiagChar NNN in XYZABC.FN6 exceeds byte escape, set to 0.425 in ** if the font is not used Interpress and justification is not required. If the switch /A assign escapement is added to the command line (that is , SCALE XYZABC.FNT /A), SCALE sets the width of the large characters to 255/600 (.425) inches. Widths overridden in this manner may yield "compressed" characters when used in Interpress printing. Problem characters will typically be in 36 point fonts and above.

Restrictions The following restrictions apply:

- When large 300 spi proportional spaced fonts (or logos) are scaled to 600 spi and used in printing an Interpress job, the results may not be correct. If any scaled character widths are larger than 255/600 (.425) inches, the output font will not generate and the `FONT FILE, %s, CANNOT BE SCALED(BYTE ESCAPEMENT)**` message appears.
- Do not interrupt SCALE processing. If an interruption occurs, enter a carriage return to resume the process.

Controlling stock input to print jobs

This section identifies several methods for controlling stock input to print jobs.

High-capacity feeder (HCF)

In addition to the two paper feeder trays resident in the LPS (identified as tray 1 and tray 2), the Xerox DocuPrint 180 LPS comes with two feeder-stacker modules. Each feeder-stacker module has a high-capacity feeder tray. These trays are referred to as tray 3 and tray 4. Optionally, two additional feeder stackers can be added. These feeder trays are referred to as tray 5 and tray 6. These additional trays add flexibility by allowing greater stock loading and increasing production due to less required operator intervention.

Each of the six trays on the LPS can be addressed and used individually. They can also be addressed in groups (such as pairs, groups of three, or all six) and used in groups as clusters. Refer to the cluster information below.

What clusters are

Clusters are a systematic method provided for feeding specific paper stock to print jobs.

Cluster features

The following are major cluster features:

- One or more feeder trays are identified as a single logical group called a *cluster*.
- Each cluster feeds one stock type to the LPS which automatically selects between the cluster's trays as they become empty or as page size requirements change.
- A cluster can have stock in different sizes.
- One or more trays in the cluster may be *preferred*. Preferred trays are trays recommended by the application programmer or by the site supervisor for the operator to load the stock.
- The trays actually loaded with the stock are the current trays.
- The operator can override the preferred tray recommendation.

- The data stream can specify the stock used to feed a page.
- The operator can override the data stream instructions by selecting a specific cluster or tray.

Cluster processing overview

Briefly, the programmer creates applications which tell the system controller on which stock jobs are to be printed. The operator loads the stock and informs the system controller which stock is in which tray. The system controller then selects the stocks as requested by the application, selecting a feeder tray for each page by checking the cluster name and paper size associated with the page.

In response to the CLU command the LPS displays cluster status as shown in figure 10-1.

Figure 10-1. **Cluster information display**

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
MAIN	A	1	1	Has Trays
AUX	A	2	2	Has Trays
AUTO		1 2 3	1 2 3	Has Trays
WHITE	A	1 2	Is Defined	
RED		3	Is Defined	

The information in the CLUSTER, ASIZ, and PREFER columns is set up by the programmer. The CURRENT column information is updated by the operator; and the STATUS column information is updated by the system controller.

- The CLUSTER column displays the name of the cluster.
- An A in the ASIZ (autosize) column indicates that the cluster may have multiple sizes of paper.
- The PREFER column indicates in which tray the programmer recommends that the stock be loaded for greatest efficiency.
- The CURRENT column lists which trays are currently assigned to the cluster.
- The STATUS column displays Has Trays for clusters with loaded stock and Is Defined for clusters without loaded stock. (See Cluster classification below.)

Cluster classification

Clusters are classified in the STATUS column.

Table 10-4 shows the cluster status categories.

Table 10-4. **Status categories for clusters**

Term	Cluster status
Undefined	The cluster is not in the printer's cluster database; it cannot be used until it is defined.
Is Defined	The cluster is in the printer's cluster database and has defined preferred trays, but the cluster has no trays currently assigned to it.
Has Trays	The cluster is in the printer's cluster database and has defined current trays. At least one tray has been assigned to the cluster.

Table 10-5 shows the cluster processing status.

Table 10-5. **Processing status for clusters**

Term	Cluster status
Active	The cluster is shown as Has Trays in the STATUS column.
Inactive	The cluster is shown as Is Defined in the STATUS column.
In Use	The cluster is active or inactive; and it was called by stock name by the most recent STOCKSET command in the job's data stream.
Missing	The cluster is undefined; and it was called by stock name by the most recent STOCKSET command in the job's data stream.

What clusters do for the programmer and operator

Clusters enable the programmer and the operator to improve their performance.

Programmer

As the programmer, you can group any combination of trays into a cluster, as long as they hold the same stock or stock which differs only in size. You can specify the cluster's paper stock in any combination of weight, color, size, and so on.

Use clusters to include instructions within the report, identifying the following:

1. The clusters needed by the report (such as the report's "stockset")
2. The particular cluster to be used to print each particular page
3. The recommended trays for loading the stock.

Then you can store the defined clusters in the CLUSTR.LIB file with the cluster (CLP or CLU) commands described in the "Cluster commands" section later in this chapter. CLUSTR.LIB, the printer's

cluster database, stores up to 159 clusters. The clusters can be called by all jobs run on the system. Since jobs often require various stocks, you can also define cluster groups as “stocksets.” The stocksets can also be called by all jobs run on the system.

Operator As the operator, you can use clusters to decrease how often you load stocks. You can verify job and cluster status, and override tray specifications for clusters with the cluster (CLU) commands described in the “Cluster commands” section later in this chapter. The cluster definitions inform you which stocks to load in which trays for queued jobs.



Note: When you load trays without entering CLU commands, the system assumes that the trays have been refilled and no cluster changes have occurred.

Where clusters are stored

The CLUSTR.LIB file is the cluster database for the LPS. Each cluster definition is a record in the file. This file is created when a print job is run and no CLUSTR.LIB exists.

CLUSTR.LIB is a user file which you can copy from one system to another even if the first LPS has trays which the second LPS does not have. (The trays not in the second system are deleted from the cluster definitions which reference them.)

The default CLUSTR.LIB created contains definitions for the MAIN, AUX, and AUTO clusters. These clusters can be used as they are or modified before use, but they cannot be deleted.

MAIN is predefined as tray 1

AUX is predefined as tray 2

AUTO is predefined as trays 1, 2, 3, 4

The OUTPUT FEED = statement is valid with FEED = MAIN and FEED = AUX because the system defines the MAIN and AUX trays as clusters to support applications designed for earlier software versions. Both clusters are always active regardless of whether they are specified in the active stockset.



Note: Do not use FEED = MAIN and FEED = AUX in conjunction with stocksets.

You cannot edit CLUSTR.LIB. It is automatically updated by the CLU and CLP commands which enter clusters in the database or modify cluster definitions.

How applications use clusters

Some applications do not change stock types (simple applications). Some applications change stock types but do not use stockset commands to do so (OTEXT applications). Some change stock types using stockset commands (stockset applications). The appropriate cluster usage for these three types of applications is discussed below.

Simple and OTEXT applications

These applications do not provide for the use of different stocks. If the application is online, then it uses the default stock for the site. If the application is offline, and the operator knows from the START command which stock to load, the operator loads the stock. Simple applications are fed stock from the MAIN and AUX clusters. MAIN and AUX are defined by the programmer to improve printer performance for the jobs run by the site, ensuring that paper can be loaded during printing.



Note: If your site runs simple, OTEXT, and stockset jobs, then it is useful to maintain a cluster for each stock loaded using cluster/stockset definitions and changing MAIN and AUX to logicals for each cluster as required by the jobs.

Stockset applications

Each page has a particular stockset and a particular cluster associated with it. The cluster is always verified because it determines from which trays a page may feed. The stockset is verified only if the current page calls a different stockset than the previous page called.

Table 10-6 shows the cluster checks required for stockset changes.

Table 10-6. **Cluster checks for stockset changes**

Reported error	Check
Missing	Is the cluster defined in the CLUSTER.LIB?
No trays	Does the cluster have current trays?
Tray overlap	Do any of the clusters have the same current trays?
Sizing error	Are all trays the same size? If not, is autosize set?

A cluster without these errors has an “in use” status, such as it has trays and is in the stockset last called by the job. Any one of these errors in any clusters in the stockset gives “in error” status to the stockset. Tray Overlap and Sizing Error are fatal errors: the printer cycles down if it is active or it does not cycle up if it is inactive; it cycles up only after you remedy the problem. Missing and No Trays are not fatal errors: the printer cycles down only if the cluster in error is called by a report in the job.

The following errors are declared only the first time they are encountered in the stockset:

Sizing error when paper is too large for data
(requires OUTPUT SIZING = SEMIAUTO)

Data offpage errors

Operator FEED command overrides

As jobs print the output task uses the application's specified cluster to choose a tray for each page. Occasionally it provides a cluster status display. If you enter FEED commands that override the data, for example FEED = DJDE, then the printer cycles down and displays:

OS2289 Warning: Current FEED command overrides data
FEED= value.

This confirmation is issued once per stockset. To respond, enter a FEED command which allows the LPS to select the feeder tray (refer to the FEED override matrix) or enter CONTINUE O. If stockset or cluster problems are found, the output task informs you with appropriate messages and blinking status values for the problem clusters. Once you correct the error or decide to override the status, enter CONTINUE O.

Mixing applications

When a site runs simple and OTEXT applications with stockset applications the LPS executes two additional stockset checks to help ensure the correct paper is loaded. They are as follows:

- If a report has no stockset information, then a stockset change is assumed.
- A stockset change is assumed at each job boundary, regardless of the stocksets used.

To run a simple or OTEXT application correctly after a stockset application while avoiding printer cycle downs, ensure that the current trays are always in the MAIN and AUX clusters.

A stockset application run after a simple or OTEXT application always causes a printer cycle down, allowing you to load the correct trays (if necessary) before continuing.

Defining clusters and stocksets with PDL and DJDE

This subsection describes how to define stocksets with PDL and DJDE.

Basic considerations

Assign each cluster a unique logical name which begins with a letter and consists of 1 to 6 alphanumeric characters. Make certain that the cluster name is not a keyword, parameter, or option used by the LPS. Group the clusters used by each application into a stockset.

Stocksets convert cluster references to cluster names. (Cluster references are recommended but are not required.) The cluster name appears first in quotes, followed by its cluster reference. The cluster is identified to the operator by cluster name. Try to define stocksets with a maximum of one cluster for each feeder tray on the LPS since the printer often cycles down when a report calls a stockset with more clusters than the number of feeder trays on the system. You may define stocksets within JDLs or catalog stocksets separately in an .STK file. Like .PDE files, .STK files are accessible to all .JDL files. If you define the stockset in an .STK file, you must recompile only the .STK file to effect a stock change. If you define the stockset in the .JSL files which use it, you must recompile all the .JSL files to effect a stock change.



Note: Avoid using stocksets with the cover and RAUX options. Cover pages and pages which meet the RAUX criteria are fed from the AUX cluster, overriding the FEED = option (or any other option) specified for these pages.

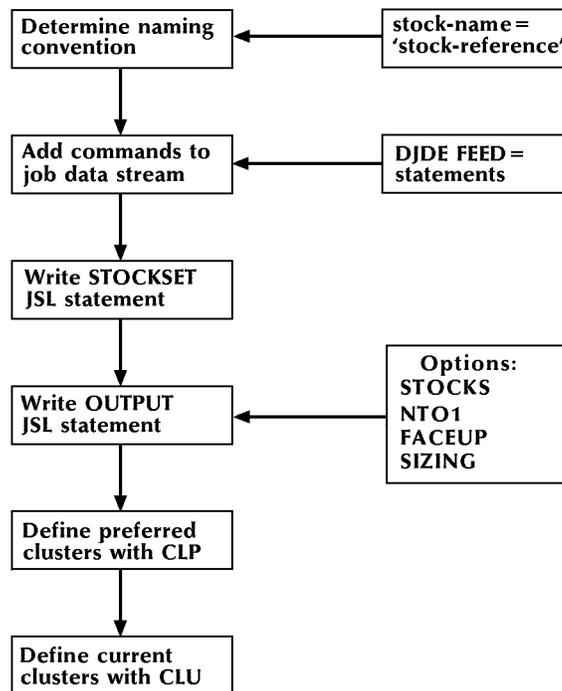


Note: Refer to the *Xerox DocuPrint 180 PDL Reference* for instructions on using the RFEED command to select a cluster for a page on the basis of record content.

Steps for creating clusters

Figure 10-2 shows the steps required for creating clusters.

Figure 10-2. **Steps for creating clusters**



Procedure Follow these steps:

- Step 1.** Determine the stocks the application will use.
- Step 2.** Create a .JSL file and compile it to produce an .STK file: Define a logical mnemonic name (a cluster reference) for each stock (for example COVER in the example). Associate each cluster reference with a cluster name easily recognized by the operator as a particular stock, such as CV102 in this example:

```

BILLS: STOCKSET  ASSIGN = ('CV102', COVER),
                  ASSIGN = ('BL231', BILPAG),
                  ASSIGN = ('SM233', SUMMRY);
  
```

- Step 3.** Use the cluster reference rather than the cluster name in .JSL files and DJDEs. This way you can make cluster stock changes by modifying the cluster name in the STOCKSET command and recompiling it (without modifying the data stream or the application.) The cluster reference in .JDL files and .STK files remains valid.

Step 4. Use the FEED = DJDE option in the .JSL file in order to specify the stock to be used. For example:

```
...
...data for cover letter...
DJDE FEED = BILPAG,END;
...data for bill...
DJDE FEED = SUMMRY,END;
...data for summary sheet
(and in the .JSL file)
OUTPUT STOCKS = BILLS, FEED = COVER;
```

Restrictions The following restrictions apply:

- The specified cluster must be identified in the active stockset.
- If clusters are to be used in a job, the OUTPUT statement requires the STOCKS = option; the FEED = option is optional. The INIFEED = option of the stockset takes effect when no FEED = option is coded. If the stockset has no INIFEED = option, then the first cluster specified in the stockset is used. The following example produces the same results as the previous example.

```
BILLS: STOCKSET  ASSIGN = ('CV102', COVER),
                  ASSIGN = ('BL231', BILPAG),
                  ASSIGN = ('SM233', SUMMRY);
                  INIFEED = COVER;

OUTPUT          STOCKS = BILLS;
```



Note: The STOCKS=DJDE makes the stockset's INIFEED = option take effect, thereby canceling the previous FEED = option.

The STOCKSET statement has an optional SYSPAGE = option. It feeds the system pages (for example PLABEL, OPRINFO, rollover marker pages, accounting pages) from the specified cluster; otherwise these pages are printed from the MAIN cluster. The following example illustrates its use:

```
BILLS: STOCKSET  ASSIGN = ('CV102', COVER),
                  ASSIGN = ('BL231', BILPAG),
                  ASSIGN = ('SM233', SUMMRY);
                  ASSIGN = ('RED00',RED),
                  INIFEED = COVER;

OUTPUT          STOCKS = BILLS;
```

RFEED (record feed)

You can use the RFEED statement to feed a print job on a page basis without using DJDE. The RFEED command tests records for certain criteria. If the record meets the test, the cluster change specified in the command occurs.

Syntax RFEED TEST = *{{(test-exp,cluname)}*

Parameters Table 10-7 shows the parameters of the RFEED command.

Table 10-7. **Parameters of the RFEED command**

Parameter	Description
<i>test-exp</i>	Specifies the criteria table.
<i>cluname</i>	Specifies the cluster to feed the page. To specify a cluster name, use quotes around the cluname variable. To specify a cluster reference, do not use quotes around the cluname variable.

When a criterion is met, the job feeds from the cluster until the next RFEED criterion is met or until the end of the job.

Example 1 T1: TABLE CONSTANT = ('ABC');

C1: CRITERIA CONSTANT = (4,3,EQ,T1);

RFEED TEST = (C1,'RED');

If the C1 criterion is met, then the paper for the current page is fed from the RED cluster.

Example 2 T1: TABLE CONSTANT = ('ABC');

C1: CRITERIA CONSTANT = (4,3,EQ,T1);

T2: TABLE CONSTANT = ('DEF');

C2: CRITERIA CONSTANT = (4,3,EQ,T2);

RFEED TEST = ((C1,OR,C2),'RED');

If the C1 or C2 criterion is met, then the paper for the current page is fed from the RED cluster.

If multiple criteria are met for a page, the last criterion overrides the previous criteria.

For a duplex job, if logical page *n* (where *n* is even such as the back side of the page) meets the criteria, then the input task forces a blank back side and moves the logical page to the front of the next physical page. If the cluster specified for *n* is the same cluster used by the previous page, then the input task does not force a blank page. Refer to the *Xerox DocuPrint 180 LPS PDL Reference*.

The operator entered FEED = *tray-number* overrides the RFEED statement; but the RFEED statement overrides the operator entered FEED = *cluname*.

Keeping stockset changes to a minimum

Each formatted page is associated with the active stockset and the active FEED = option stock identified by INIFEED, FEED = in JSL, or the FEED = DJDE. A stockset change causes the LPS output task to make additional checks when a page is to be printed. These checks can result in a printer cycle down if any cluster in the new stockset presents a tray overlap or sizing problem. Thus you should keep stockset changes within applications to a minimum. For example:

```
DJDE STOCKS = BILLS1, END;
...data...
DJDE STOCKS = BILLS1, END;
...data...
DJDE STOCKS = BILLS2, END;
...data...
```

This data stream has three STOCKSET = callouts, but only one stockset change. Report boundaries are not treated as stockset changes unless use of the stockset specified by JDE/JDL makes a page print on another stockset. For example:

```
DJDE STOCKS = BILLS1, END;
...data...
REPORT BOUNDARY
...data...
DJDE STOCKS = BILLS1, END;
...data...
DJDE STOCKS = BILLS2, END;
...data...
```

The data following the report boundary is printed using the JDE/JDL of the START command. If it does not specify OUTPUT STOCKS = BILLS1, the stockset changes.



Note: If the stockset changes but the second stockset consists of the same clusters as the first stockset, no cycle down occurs, since for printing purposes the stockset has not changed.

Using clusters with ordered or tab stocks

Ordered stocks consist of any repeating sequence of paper stock types stacked in a tray. These clusters function in the same way other clusters function. A special case of ordered stock is a tab stock cluster: the tab positions of the tab stock are stacked in a repeating sequence. Refer to the *Xerox DocuPrint 180 PDL Reference* for further information.

Cluster commands

The **CLUSTER (CLU) command** establishes or deletes **current** trays for a cluster; the **CLP command** defines or deletes **preferred** trays for a cluster.

CLP displays all clusters with all their associated information. CLU displays all information for all clusters with current trays.

A cluster exists permanently in the database if it has at least one preferred tray. If you delete an active cluster's preferred trays, by assigning them to another cluster with the CLP command, you can use the cluster until you delete its current trays by assigning them to another cluster with the CLU command.

Syntax {CLUSTER | CLP} [*clu-name*] [[/][-]A] [[n] | [*n*₁,...*n*_n]] [+ | -] [SET] [ALL] [/NR]

Parameters Table 10-8 shows the parameters of the CLU/CLP command.

Table 10-8. **Parameters of the CLU/CLP command**

Parameter	Description
CLUSTER	Displays information for, creates, or modifies current clusters.
CLP	Displays information for, creates, or modifies preferred clusters.
<i>clu-name</i>	Specifies a cluster by name, rather than by tray number. Must be 1 to 6 alphanumeric characters which begin with a letter.
[/]A	Sets the autosize flag to True.
[/] -A	Sets the autosize flag to False.
[n]	Denotes a feeder tray on the printer by a number within the range 1 to the highest configured feeder, a bulk feeder being number 7.
[<i>n</i> ₁ ,... <i>n</i> _n]	Denotes feeder trays on the printer by their numbers within the range 1 to the highest configured feeder, a bulk feeder being number 7.
+	Adds trays as specified by the other parameters.
-	Deletes trays as specified by the other parameters.
[SET]	Assigns current tray status to all the preferred trays in the cluster.
[ALL]	Specifies all the trays on the printer.
[/NR]	Specifies that the assigned current trays are not removed from other clusters.

Examples Table 10-9 shows examples of the commands which control clusters.

Table 10-9. **Cluster command examples**

Command	Function
CLUSTER	Displays all tray information for all clusters with current trays.
CLP	Displays all tray information for all clusters.
CLUSTER <i>cl-name</i>	Displays all tray information for the specified cluster.
CLUSTER <i>n</i>	Displays all tray information for all clusters which have the specified tray as a current tray.
CLP <i>n</i>	Displays all tray information for all clusters which have the specified tray as a preferred tray.
CLUSTER <i>cl-name</i> <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}]/[NR] CLUSTER <i>cl-name</i> ALL [NR] CLUSTER [<i>cl-name</i>] SET/[NR] CLUSTER SET ALL [NR]	Assigns current trays to the specified cluster. ALL specifies all the trays on the printer. NR (not remove) specifies that the assigned <u>current trays</u> are not deleted from other active clusters (such as these trays may be current trays in multiple clusters). SET assigns current tray status to all the preferred trays in the specified cluster.
CLP <i>cl-name</i> [/[A] <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}] CLP <i>cl-name</i> [/[A]ALL	Replaces the preferred trays of the specified cluster with the specified trays. ALL refers to all the trays on the printer.
CLUSTER <i>cl-name</i> + <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}]/[NR]	Adds the specified trays as current trays to the specified cluster. NR specifies that the assigned current trays are not deleted from other active clusters (such as these trays may be current trays in multiple clusters).
CLP <i>cl-name</i> [/[A]+ <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}]	Assigns the specified trays as preferred trays to the specified cluster.
CLUSTER <i>cl-name</i> - <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}] CLUSTER <i>cl-name</i> -ALL	Removes the specified current trays from the specified cluster. ALL refers to all the trays on the printer.
CLP <i>cl-name</i> [/[A] - <i>n</i> ₁ [... <i>n</i> _{<i>n</i>}] CLP <i>cl-name</i> [/[A] -ALL	Removes the specified preferred trays from the specified cluster. ALL refers to all the trays on the printer.



Note: You can protect the preferred trays in the cluster database by restricting the CLP command with RCU.

Using cluster commands

Use cluster references only to set up print jobs, since all the commands you issue to the LPS refer to cluster names. Define clusters and enter them into the cluster database with the CLU and CLP commands. First determine each cluster's preferred tray settings and its autosizing attribute, as follows:

1. Consider the characteristics of the jobs printed at the site; then assign preferred trays to clusters in order to establish a

standard loading pattern across the site's printers. Assign larger and more trays to heavily used clusters. Avoid assigning the same trays to clusters which are often used together.

For example, if clusters RED, WHITE, and BLUE are often called for by a job, and WHITE is used most often, RED least often, enter:

```
CLP WHITE 1,3
```

```
CLP BLUE 4
```

```
CLP RED 2
```

After each command above the LPS displays the status of the modified cluster. If you enter CLP after all these commands, the system displays:

CLUSTER	ASIZ	PREFER	CURRENT	STATUS
MAIN	A	1	1	Has Trays
AUX	A	2	2	Has Trays
AUTO	A	1 2 3 4 5 6	1 2 3 4 5 6	Has Trays
WHITE		1 3		Is Defined
BLUE		4		Is Defined
RED		2		Is Defined



Note: If you enter CLU after all these commands, the system displays the first three lines above.

2. Set the autosize attribute for the cluster if the cluster has multiple paper sizes, and the LPS will assign the stock to each page after evaluating its size. /A in the CLP command sets the autosize attribute; /-A turns the autosize attribute off. For example:

```

CLP WHITE /A, 2
CLUSTER  ASIZ  | PREFER | CURRENT | STATUS
WHITE    A     | 2      |         | Is Defined
CLP WHITE +3
CLUSTER  ASIZ  | PREFER | CURRENT | STATUS
WHITE    A     | 2 3    |         | Is Defined
CLP WHITE /-A, -2
CLUSTER  ASIZ  | PREFER | CURRENT | STATUS
WHITE    A     | 3      |         | Is Defined

```

Cluster command examples

The following examples show the keyed in command in boldface.

Given that cluster White has preferred trays 1, 3 and 6:

```

CLP WHITE +4
CLUSTER  ASIZ  ----PREFER----  ----CURRENT----  ----STATUS----
WHITE    1    3                      Is Defined
CLP WHITE -3,4
CLUSTER  ASIZ  ----PREFER----  ----CURRENT----  ----STATUS----
WHITE    1                      Is Defined
CLP WHITE -ALL
0S0700 Cluster WHITE now deleted.
CLP WHITE
0S1291 Cluster name not defined.
CLP WHITE ALL
CLUSTER  ASIZ  ----PREFER----  ----CURRENT----  ----STATUS----
WHITE    1  2  3                      Is Defined
CLP WHITE 1,3
CLUSTER  ASIZ  ----PREFER----  ----CURRENT----  ----STATUS----
WHITE    1    3                      Is Defined

```

Keeping clusters in the database

A cluster is deleted from the cluster database when a CLU or CLP command deletes all trays (current and preferred) assigned to it.

Keep a cluster in the database by assigning it one or more preferred trays and using only the CLU command for operator tasks.

Operator tasks using clusters

The cluster definition identifies the preferred set of trays in which you normally load the stock. When you are requested to load paper, the system identifies the cluster's trays by their address (number). You can load the paper in the preferred trays (and confirm the preferred tray loading), or you can override the recommended loading by loading other trays and identifying them as the current trays for the cluster.

Some clusters have trays with one paper size and trays with other paper sizes. Refer to the "Defining clusters for printing" section above.

When you assign a current tray to a cluster (say cluster COVER), that tray is automatically deleted from the list of current trays of other clusters (say cluster LEGAL). If the cluster which loses the current tray (cluster LEGAL in this example) has no preferred trays assigned to it, then the cluster (cluster LEGAL) is left without trays and is automatically deleted from the cluster database.

To avoid deleting clusters in this way, maintain preferred trays for all clusters, or use the /NR parameter when you assign current trays.

Irregular screen displays

During system controller connection with the printer, cluster commands that produce more than one screen display may result in confusing displays. In order to avoid irregular displays, allow each cluster command to complete before entering the next cluster command.



Note: This situation usually occurs when the output task is not in memory, for example just after a boot, and you enter a CLU command. Because output must first connect to the printer before processing the cluster command, the system delays about 5 seconds before responding. If you enter a second cluster command during those 5 seconds, and the first CLU command requires you to press return before displaying a second response screen, the screen activity appears abnormal.

Changing stock in trays

If you change the stock in a tray, you must also change at least one cluster definition. For example, if tray 2 is in the list of currently loaded trays for the BLUE cluster, and you replace the blue paper in it with red paper, add tray 2 to the RED cluster with the CLU command before continuing the print job. The CLU command removes tray 2 from the current definition of the other clusters.

When you change the paper stock in response to a system generated request, enter the CLUSTER SET command as follows to confirm that all requested changes including new cluster definitions have been made:

Syntax CLUSTER [*cl-name*] SET

This command tells the system controller that the cluster is loaded exactly as specified by the cluster's defined preferred trays. The system responds by updating the cluster's status to show that the preferred trays are also the current trays. You can enter this command as CLUSTER SET, omitting a cluster name. This form of the command tells the LPS that all clusters in the active stockset are loaded as specified by their cluster definitions. The system controller responds by updating the status of all active clusters, showing that their preferred trays are the current trays.

Procedure Use the following steps to avoid feeding the stock for the following job to the current job:

- Step 1.** Lower the tray holding the stock to be changed. (The system now cannot select the tray until the stock change is completed.)
- Step 2.** Enter the CLU *cl-name n* (tray number) command. This command both adds the tray to the cluster for the following job, and deletes the tray from the cluster for the current job. For the above example the appropriate command is:

```
CLU RED +2
```
- Step 3.** Load the new stock in the tray.

Sample operator procedure

In the following example:

1. Enter CLU to see the current assignments.
 2. GREEN has no current trays, so enter CLU GREEN to see the GREEN cluster.
 3. Since tray 3 is the preferred tray, load tray 3 with green paper.
 4. Enter CLU GREEN 3 to tell the LPS it has been done. (The LPS displays the new status for the GREEN cluster.)
 5. Then enter CLU to verify that tray 3 was removed from the current set for the RED and AUX clusters.
1. **CLU**

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
MAIN	A	1	1	Has Trays
AUX	A	2	2	Has Trays
AUTO	A	1 2 3	1 2 3	Has Trays
RED		3	3	Has Trays
WHITE		1 3	1	Has Trays

2. **CLU GREEN**

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
GREEN		3		Is Defined

3. Load tray 3 with green paper.

4. CLU GREEN 3

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
GREEN		3	3	Has Trays

5. CLU

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
MAIN	A	1	1	Has Trays
AUX	A	2	2	Has Trays
AUTO	A	1 2 3	1 2 3	Has Trays
RED		3	3	Has Trays
BLUE		3	2	Has Trays
GREEN		3	3	Has Trays
WHITE		1	1	Has Trays

A current tray in two clusters

Sometimes a current tray belongs to two clusters. In the sample procedure above the MAIN cluster and the WHITE cluster have the same trays in the preferred and current sets. Since MAIN is not an actual stock name, the site has declared MAIN to be a logical for the WHITE stock.

Look at step 4 in the sample above. If the command includes the /NR (not remove) option, three clusters (AUTO, RED, and GREEN) will use tray 3 to feed the job, as follows:

4. CLU GREEN 3/NR

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
GREEN		3	3	Has Trays

5. CLU

CLUSTER	ASIZ	----PREFER----	----CURRENT----	----STATUS----
MAIN	A	1	1	Has Trays
AUX	A	2	2	Has Trays
AUTO	A	1 2 3	1 2 3	Has Trays
RED		3	3	Has Trays
BLUE		3	2	Has Trays
GREEN		3	3	Has Trays
WHITE		1	1	Has Trays

If you intend that tray 3 should feed only one of the clusters, this situation is an error. Sometimes the output task checks whether multiple clusters have the same current tray assigned to them; sometimes it does not check. This means the system may not flag this situation as an error. So, use the SET and NR parameters as shown below only during printing. Doing so helps you avoid assigning the same tray to more than one cluster.

```

CLU cl-name SET
CLU SET
CLU cl-name SET/NR
CLU SET/NR
    
```

Assigning the input trays with FEED

Use the FEED command to assign the input trays.

FEED (select input tray)

The operator can select the input paper trays or they can be selected automatically. The FEED command options specify how the input paper trays will feed stock to the job. FEED controls the primary source of paper.

Syntax FEED {MAIN | AUX | AUTO | *tray-number* | *cluster-name*}

Parameters Table 10-10 shows the parameters of the FEED command.

Table 10-10. **Parameters of the FEED command**

Parameter	Description
(no parameter)	Has no effect unless the current mode is FEED MAIN or FEED AUX. If so, switches to the other mode.
MAIN	Assigns to the input data the control over the source of the paper. The FEED MAIN command is the only command which ensures that all sheets which are designated with a stock name will be fed from the cluster associated with page.
AUX	Feeds all the following sheets from the AUX cluster. This command overrides any job-specified feed source.
AUTO	Is a special instance of FEED stock name. The AUTO cluster is predefined to be all trays physically present in the printer. You can redefine AUTO as any tray on the printer. AUTO overrides the job specified feed source.
<i>tray-number</i>	Feeds all the following sheets from the specified tray, regardless of the stock specifications in the FEED= JDL/DJDE options. This command overrides any job-specified feed source. Must be an integer between 1 and the highest configured feeder, a bulk feeder being number 7.

Table 10-10. Parameters of the FEED command

Parameter	Description
<i>cluster-name</i>	Feeds all the sheets from the specified cluster, regardless of the stock specifications in the FEED= JDL/DJDE options. This command overrides any job-specified feed source.

Important The cluster options now available change the execution of the local language commands for FEED MAIN, FEED AUX, and FEED AUTO.

- FEED AUX or FEED *cluster-name* affect paper feeding in jobs: only the English command “FEED MAIN” is overridden by the job’s .JSL file or DJDE records.
- You may set up clusters using the local language keywords for MAIN, AUX, and AUTO so that the system processes print jobs with the same results as produced by OSS version 2 and earlier versions.



Note: Keep in mind the following:

1. The MAIN cluster is predefined as tray 1, and the AUX cluster is predefined as tray 2. But either or both of these clusters may be redefined by the user.
2. FEED 1 and FEED MAIN are not the same, even if the MAIN cluster consists of only tray 1. FEED 2 and FEED AUX are synonymous if the AUX cluster consists of only tray 2.
3. Under certain circumstances, the automatic switching between two current FEED CLUSTER trays (for example FEED MAIN, where MAIN =1,2) may not take place. Specifically, if one tray feeds empty and lowers at the same time as the other is being readied to feed, the printer may cycle down and display the message: OS2091 NO AVAILABLE TRAYS ARE READY. To correct this condition, load one of the trays.
4. If your system does not use English, you must rename the MAIN, AUX, and AUTO clusters.

The FEED override sequence

Table 10-11 shows how keyed in FEED commands override each other.

Table 10-11. FEED override sequence

Command	Result
FEED MAIN	Gives the application control and feeds from the MAIN cluster unless the application overrides this command. The FEED= option of the OUTPUT statement, the FEED= DJDE, RFEED, COVERS, and RAUX pages feed from the clusters specified by the application. This is the normal operating mode.
FEED AUX	Overrides all application FEED instructions. All sheets feed from the trays in the AUX cluster. (If the AUX cluster definition is not modified, this command is equivalent to FEED 2.)
FEED <i>n</i>	Overrides all application FEED instructions.
FEED <i>cl-name</i>	Overrides all the application's FEED instructions except pages to be fed from the AUX cluster. These pages are fed from the AUX cluster specified by the application.
FEED AUTO	Same as FEED <i>cl-name</i> . AUTO is a predefined cluster.

Assigning an input tray for purge sheets

Use the PFEED command to assign an input tray for purge sheets.

PFEED (purge feed)

Use the PFEED command to specify the feeder trays from which purge sheets will be taken. If no parameter is specified, the system displays the trays currently assigned for purge feeding.

The feeder trays specified must contain paper of the largest size used by the current print job in order to ensure that the entire photoreceptor surface will be cleaned.

Syntax PFEED [*tray list*]

Parameter Table 10-12 shows the parameter of the PFEED command.

Table 10-12. **Parameters of the PFEED command**

Parameter	Description
<i>tray list</i>	<p>Specifies the tray or trays from which the system will take purge sheets. Each tray number must be a number within the range 1 to the highest configured feeder, a bulk feeder being number 7.</p> <p>If you enter a number outside the valid range, the system displays the message:</p> <p>OS1190 Invalid parameter, command ignored</p>

Restrictions

The following restrictions apply:

- Do not specify a tray containing ordered stock.
- To properly clean the fuser area, at least one of the trays specified must contain paper at least as large as the paper currently being used in the print job.

Using stocks of varying sizes

This subsection describes how to use stocks of varying sizes.

Printing system pages

Xerox DocuPrint 180 LPS paper sizes are associated with pitch modes. Pitch mode is the number of page images put on the LPS photoreceptor. For example: if the pitch mode is 5, the LPS prints 5 pages during each turn of the photoreceptor and if the pitch mode is 7, the LPS prints 7 pages during each turn of the photoreceptor.

If the stock for a job is 7 to 7.4 inches/170 to 188 mm in scan length, the LPS prints at 8-pitch mode. If the stock for a job is 8 to 9.02 inches/203 to 229 mm in scan length, the LPS prints at 7-pitch mode. If the stock is 9.02 to 10.21 inches/229 to 259 mm in scan length, the LPS prints 6 pages during each turn of the photoreceptor (6-pitch mode). If the stock is 10.22 to 12.13 inches/260 to 308 mm in scan length the LPS prints at 5-pitch mode. If the stock is 12.14 to 15.33 inches/309 to 389 mm in scan length, the LPS prints at 4-pitch mode. If the stock is greater than 15.33 inches/389 mm in scan length, only three page images fit on the photoreceptor, so the LPS prints at 3-pitch mode.

As the pitch mode decreases, so does the pages per minute. In addition, some performance degradation occurs when printing a system page that causes a pitch mode change. In order to change modes the LPS must stop printing, reset the pitch mode, and resume printing.

Table 10-13 shows the maximum and minimum paper size lengths used by each pitch mode.

Table 10-13. **Pitch mode boundary values**

Pitch Mode	Minimum scan value	Maximum scan value
8	7 inches/170 mm	7.4 inches/188 mm
7	7.4 inches/189 mm	9.02 inches/229 mm
6	9.02 inches/230 mm	10.21 inches/259 mm
5	10.22 inches/260 mm	12.13 inches/308 mm
4	12.14 inches/309 mm	15.33 inches/389 mm
3	15.34 inches/390 mm	17 inches/432 mm

Supported paper sizes

Xerox DocuPrint 180 LPS supports management of variable paper sizes in the range of 7 by 10.0 inches (170 by 254 mm) through 11.69 by 17 inches (297 by 432 mm). The standard paper sizes are listed in table 10-14.

8-pitch support is enabled by a hardware kit installed on the printer. Without this kit, the minimum size supported is 8 inches (203 mm).

Table 10-14 shows the standard paper sizes.

Table 10-14. **Standard paper sizes**

x - page width (inches)	y - page length (inches)	Generic name
7.17 (182 mm)	10.12 (257 mm)	B5
8.27 (210 mm)	11.69 (297 mm)	A4
8.50 (216 mm)	11.0 (279 mm)	USLETTER
8.50 (216 mm)	14.0 (356 mm)	USLEGAL
16.54 (419 mm)	11.69 (297 mm)	A3
10.12 (257 mm)	14.33 (364 mm)	B4



Note: The Xerox DocuPrint 180 LPS does not support Interpress or PostScript masters larger than 8.5 in. (216 mm) by 14 in. (356 mm).

SET CONTAINER (changing the size of a container)

Use the SET CONTAINER command to change maximum paper size allowed for a customized container. (This command is not used with Xerox containers.) The printer can sense the different paper sizes for each tray shown in table 10-15. If no parameter is specified, the system displays the current sizes for all seven containers.

Syntax SET CONTAINER [*n,w,l*]

Parameters Table 10-15 shows the parameters of the SET CONTAINER command.

Table 10-15. **Parameters of the SET CONTAINER command**

Parameter	Description
<i>n</i>	Specifies a valid container type by a number within the range 1 to 7.
<i>w</i>	Specifies the minimum container width within the range 178 mm through 432 mm (7 to 17 inches)
<i>l</i>	Specifies the container length within the range 254 mm through 364 mm (10 to 14.33 inches).

Restrictions In character mode, only values given in millimeters (mm) are valid.

SET TRAY (changing paper size for trays)

Use the SET TRAY command to identify to the printer nonstandard sizes of stock loaded in feeder trays. The printer uses this information to check print registration, to monitor paper status (size, not folded, not skewed) and to control paper flow through output delivery. The supported size ranges for stock loaded in trays 1 through 7 are listed in tables 10-17 through 10-19.

Syntax SET TRAY *n,w,l*

Parameters Table 10-16 shows the parameters of the SET TRAY command.

Table 10-16. **Parameters of the SET TRAY command**

Parameter	Description
<i>n</i>	Specifies the target tray number.
<i>w</i>	Specifies the paper width (in inches or millimeters). Must be within the range 178 mm through 364 mm (7 inches through 14.33 inches).
<i>l</i>	Specifies the paper length (in inches or millimeters).

Restriction In character mode, only values given in millimeters (mm) are valid. For paper sizes 7.4 in (188 mm) or less, 8 pitch is the default pitch mode. This mode increases the maximum system output speed to 154 ppm.

Table 10-17 lists the size ranges for feeder trays 1 and 2.

Table 10-17. **Size ranges for feeder trays 1,2**

Zone	Trays 1,2	
	Length (inches)	Width (inches)
0	11.18 to 11.85 (284 to 301 mm)	7 to 9.05 (178 to 230 mm)
1	11.88 to 13.5 in. (302 to 343 mm)	7 to 9.05* (178 to 230 mm)
2	10 to 11.14 (254 to 283 mm)	7 to 9.05* (178 to 230 mm)
3	13.54 to 14.02 (344 to 356 mm)	7 to 9.05* (178 to 230 mm)
4	Not present	Not present
5	Not present	Not present
6	Not present	Not present
7	Not present	Not present

* If the system does not support the 7- by 10-inch (178- by 254-mm) feature, the minimum paper size is 8- by 10-inch (203- by 254-mm).

Table 10-18 lists the size ranges for feeder trays 3, 4, 5 and 6.

Table 10-18. Size ranges for feeder trays 3,4,5,6

Zone	Trays 3,4,5,6	
	Length (inches)	Width (inches)
0	11.18 to 11.85 (284 to 301 mm)	7 to 9.05 (178 to 230 mm)
1	11.88 to 13.5 (302 to 343 mm)	7 to 9.05* (178 to 230 mm)
2	10 to 11.14 (254 to 283 mm)	7 to 9.05* (178 to 230 mm)
3	13.54 to 14.33 (344 to 364 mm)	7 to 9.05* (178 to 230 mm)
4	11.18 to 11.85 (284 to 301 mm)	9.09 to 17.0 (231 to 432 mm)
5	11.88 to 13.5 (302 to 343 mm)	9.09 to 17.0 (231 to 432 mm)
6	10 to 11.14 (254 to 283 mm)	9.09 to 17.0 (231 to 432 mm)
7	13.54 to 14.33 (344 to 364 mm)	9.09 to 17.0 (231 to 432 mm)

* If the system does not support the 7- by 10-inch (178- by 254-mm) feature, the minimum paper size is 8- by 10-inch (203- by 254-mm).

Table 10-19 lists the size ranges for feeder tray 7.

Table 10-19. Size ranges for feeder tray 7

Zone	Tray 7	
	Length (inches)	Width (inches)
0	10.0 to 14.33 (254 to 364 mm)	7.00 to 17.0* (178 to 432 mm)
1	Not present	Not present
2	Not present	Not present
3	Not present	Not present
4	Not present	Not present
5	Not present	Not present
6	Not present	Not present
7	Not present	Not present

* If the system does not support the 7- by 10-inch (178- by 254-mm) feature, the minimum paper size is 8- by 10-inch (203- by 254-mm).

Feeding paper from the short edge

Stock is normally fed into the printer with the longest edge of the sheet leading (long edge feed), and you assign each job a font with a landscape orientation (characters are aligned along the long edge of the page) or a font with a portrait orientation (characters are aligned along the short edge of the page). The Xerox DocuPrint 180 LPS supports paper sizes (17 by 11 inches and A3) for which the longest dimension exceeds the width of the photoreceptor belt. This means that this stock is fed into the printer with the shortest edge of the sheet leading. This is the *short edge feed* (SEF) option. Refer to the “Supported paper sizes” and “Printing system pages” sections above.

The reversal of the feed edge reverses the font orientation the job must employ. In this case, you implement SEF with PDL commands. First use a PDL command to specify the paper size values as (x,y) where x is the larger value. Then use PDL commands to map each landscape font used in a SEF job to its non-SEF portrait equivalent, and each portrait font used in a SEF job to its non-SEF landscape equivalent. Refer to the *Xerox DocuPrint 180 LPS PDL Reference*.

Remember that stock printed in 3-pitch mode, including all SEF jobs, prints at a lower throughput rate than does stock printed in 7-pitch mode.



Note: XPS/XPAF do not support SEF jobs.

Figure 10-3 displays a non-SEF job.

Figure 10-3. **Non-SEF job**

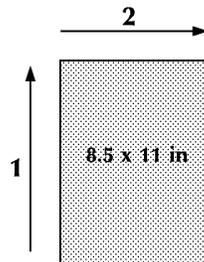
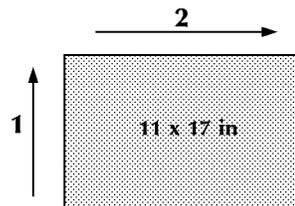


Figure 10-4 displays a SEF job.

Figure 10-4. **SEF job**



- 1 Direction of fast scan
- 2 Direction of feed

Feeding sequenced stocks

This subsection describes how to feed sequenced stocks.

OSTK (ordered stock)

Use the OSTK command to reset the logical paper stock counter in the feeder tray after you remove or add stock to a tray loaded with ordered stock.

You can sequence pre-numbered sheets or sheets with tabs repeatedly in a feeder tray. As a result, the job fed by that tray can use two or more different stocks consistently. Stock which you have sequenced in this manner is called *ordered* stock.

Syntax OSTK *m,n*

Parameters Table 10-20 shows the parameters of the OSTK command.

Table 10-20. **Parameters of the OSTK command**

Parameter	Description
<i>m</i>	Specifies the tray which holds the ordered stock.
<i>n</i>	Specifies the current position of the ordered stock. Must be a value between 1 and 254.

Example If:

- You are printing a job which is using pre-numbered sheets of paper, numbered from 1 to 10.
- The ordered stock is in tray 3.
- The printer jams on a “#6” page from tray 3.
- You decide the old stock in tray 3 caused the jam, so you replace it with new stock which begins with page “#1.”
- The LPS expects the top sheet to be “#7,” but it is actually “#1” because you replaced the stock.

You enter:

```
OSTK 3, 1
```

The command above tells the LPS that the current stock page is a “#1” page. In order to print the current job page, the LPS purges stock pages “#1” through “#5” and correctly resumes printing on stock page “#6” which was the page which jammed and was aborted.

Restrictions The following restrictions apply:

- Use the OSS OSTK command only when the print job is coded with the OSTK PDL command.
- The ordered stock (OSTK) feed tray cannot be the same as the PFEED tray. Ordered stock jam recovery does not work properly if the tray selected for PFEED contains ordered stock.
- The portrait/landscape parameter permits tab stock to have both portrait and landscape options. The supported tab stock is portrait which is the default condition

Setting up output from print jobs

This section describes how reports are sorted.

Stock size and stacker bins

The system sorts reports into the bins based on the stock size of the first sheet delivered to the bin. Reports are sent to the next available bin when the stock size differs from the first sheet delivered to the bin by more than 25 mm in length or width. Jobs which require stock differing in size more than 25 mm must be manually collated after being printed.

No size check is performed on pages being delivered to the Bypass Transport module.

Assigning the stacker bins

This subsection describes how to assign the stacker bins.

SELECT

Use the SELECT command to specify the active stacker bin or to allow the system to select it automatically. Manual bin selection allows the selected bin to be filled; when full, the operating system stops the printer. Automatic bin selection allows the active bin to be filled; when full, the operating system automatically switches to the alternate bin, thus allowing printing to continue and providing for generally improved throughput due to the overlapped operation. For both selection modes, a number of pages are delivered to the previously active bin after a new bin selection is made. Thereafter, pages are delivered to the new bin if it is ready.

Use the SELECT command to specify the bins in which the system will stack reports, or to put bin selection under automatic system control.

Syntax SELECT {A | B | C | D | E | TRAY | AUTO}

Parameters Table 10-21 shows the parameters of the SELECT command.

Table 10-21. **Parameters of the SELECT command**

Parameter	Description
A B C D	Specifies bin A, B, C, or D, respectively. If the specified bin is not ready, the printer is cycled down and an appropriate message is issued. Whenever the selected bin is filled, the printer is cycled down.
E	Specifies the Bypass Transport as the destination bin.
TRAY	Specifies the sample tray as the destination.

Table 10-21. **Parameters of the SELECT command**

Parameter	Description
AUTO	Selects another stacker bin automatically when the active bin becomes full. Selects active bin if ready or automatically switches bins at initiation of printing. AUTO places destination selection under the control of the data stream.

Restrictions The following restrictions apply:

- Transparencies must be directed to the sample tray.
- If you enter SELECT without an operand, the next stacker bin is selected but the mode does not change.
- If you enter SELECT TRAY, only the next job will be sent to the sample tray. The jobs that follow will be sent to the bin previously specified.

BLIMIT (bin limit)

Use the BLIMIT command to specify the maximum number of sheets that can be sent to a bin before it must be unloaded.

Syntax BLIMIT *bin number, sheet count*

Parameters Table 10-22 shows the parameters of the BLIMIT command.

Table 10-22. **Parameters of the BLIMIT command**

Parameter	Description
<i>bin number</i>	Must be A through D. Specifies the output bin for which a sheet limit is set.
<i>sheet count</i>	Must be a number between 100 and 2500 inclusive. Specifies how many sheets can be sent to the bin before output switches to another bin.

Restrictions If you enter a bin value which is not A through D or if you enter a number outside the valid range for sheet count, the system displays the message:

OS1190 Invalid parameter, command ignored

If you enter a bin value which is not in your LPS's stacker configuration, the system displays the message Feeder/stacker unavailable.

If you enter BLIMIT without the sheet count parameter, the printer displays the current setting for the bin.

If you enter BLIMIT without a parameter, the printer displays the message:

OS1190 Invalid parameter, command ignored

SEPARATE

The SEPARATE command causes the next output report processed to go to the next available stacker bin.

Syntax SEPARATE [OFF]

If you specify OFF, SEPARATE processing is disabled. The SEPARATE command remains in effect for the duration of the current report. At end of report, the bin change occurs, and SEPARATE is disabled until you request it again.

A bin change occurs only if the new bin is ready (such as not full or jammed). SEPARATE is disabled by one of the following:

- End of report
- SEPARATE OFF
- Current bin fills before end of report
- A SELECT command.

UNLOAD (lowering trays and bins)

Use the UNLOAD command to lower trays and bins. You can lower trays and bins only when they are up. The LPS does not allow you to select lowered or open trays or bins.

Syntax UNLOAD {*n* | *a*}

Parameters Table 10-23 shows the parameters of the SEPARATE command.

Table 10-23. **Parameters of the SEPARATE command**

Parameter	Description
<i>n</i>	Specifies the feeder tray number as 1 through the highest configured feeder.
<i>a</i>	Specifies the stacker bin letter (A through D).

This command has no default parameter. If you select a bin currently in use, then the output is sent to an alternate bin before the bin is lowered. If no alternate bins are available, then the printer cycles down before lowering the tray or bin.

BINFULL (setting maximum output to bin)

Use the BINFULL command to limit the quantity of output sent to a bin. You can define the bin as full when it holds a specified volume of output, or when the bin holds the first set, report, or job boundary after the specified volume of output is reached.

Syntax BINFULL [*threshold, boundary*]

Parameters Table 10-24 shows the parameters of the BINFULL command.

Table 10-24. **Parameters of the BINFULL command**

Parameter	Description
<i>threshold</i>	A number 0 to 5 inclusive, representing the number of reams.
<i>boundary*</i>	Must be SET, REPORT, JOB, or NONE.



Note: A bin may fill to its maximum physical capacity before the specified bin full boundary is reached. In this case the output is delivered to another bin, or printing cycles down if no bin is available. The system controller displays a message which states that a partial set, report, or job was delivered to the bin. This message indicates that the bin must be emptied.



Note: For BINFULL settings with a threshold = 0, the threshold will be reached as soon as the first page is delivered into the bin. Boundary separation will proceed as expected.

The defaults for the bin full criteria are determined by the stock's size, as follows:

2500 sheets for stock up to 8.5-by 14.0-inches

2000 sheets for B4

1500 sheets for A3

TRANS (running transparency jobs)

Transparencies must feed from trays 1 and 2, and stack to the sample tray. The cluster for the print job may specify tray 1 or 2. Even though other trays are part of the cluster, if TRANS is in effect, transparencies feed only from tray 1 or 2.

Use the TRANS command when the system is idle to override the TRANS statement in the next job's JDE/JDL data stream. The LPS automatically passes control of transparencies back to the job stream when the print job affected completes.

Syntax TRANS = {YES | NO}

If the selected feeder trays do not include 1 or 2, the printer cycles down. You must then specify tray 1 or 2 as the feeder tray and enter CONTINUE to resume printing.

Backward compatibility

Jobs which run labels must have the FACEUP command added to the .JSL file and be recompiled. Otherwise the labels may delaminate from the page when the page is delivered to the bin. Refer to the *Xerox DocuPrint 180 PDL Reference*.

This chapter explains how to run a print job on the Xerox DocuPrint 180 Laser Printing System.

Printing samples of system files

The operating system provides commands which enable you to sample print files and to control the flow of print jobs (starting, stopping, continuing, aborting) through the printing system.

As the operator you can request a sample print while a job is printing or when the system is idle. When a job is printing, the next page to be printed is printed twice. One copy is delivered to an output tray; the other is delivered to the sample print tray. If the system is idle when you request a sample, you must identify the file to be sample printed. The file can be a font, form, image, or logo file.

All sample jobs, except form samples, are formatted for printing using the default paper size specified at system generation time. Form samples are formatted for printing using the paper size specified by the form, if one is specified, or the default paper size specified at system generation if the paper size is not specified within the .FSL file. If the default paper size is smaller than a graphic image, the graphic image can be sampled only after a system generation which changes the default paper size to accommodate the graphic image.

SAMPLE (during a print job)

Use this command to request a sample print during job printing. If you have temporarily suspended a job being printed with a STOP command (or the STOP function key) or an error stop, then you can use the SAMPLE command (or the SAMPLE function key) to print the job's next sheet and send it to the sample print tray. SAMPLE runs when tape or disk print jobs are running.

Syntax SAMPLE (or press the SAMPLE key on the keyboard or on the printer)

Restrictions You cannot print a sample if:

- The document has fewer than nine pages.
- ABNORMAL SECURITY=YES is coded in the job descriptor library of the report being processed.
- An ordered stock job is running.

When the <SAMPLE> key is pressed, an extra copy of the next possible sheet must be scheduled and delivered to the sample tray. If the system is currently printing duplex and is in the middle of imaging the back or second sides, it must complete that task in order to maintain the integrity of the data stream. After completing that task, it can initiate the sample sheet. If that sample is a duplex sheet, the time to produce both sides adds to the total time, and thus the time from pressing the <SAMPLE> key to the delivery of the sampled sheet will be longer than a simplex sheet.

SAMPLE (form, font, or logo)

Use this command to request a sample print of a font, form or logo.

Syntax `SAMPLE [{file-name | mask}] .file-type [,D][,copies]`

Parameters Table 11-1 shows the parameters of the `SAMPLE (form, font, or logo)` command.

Table 11-1. **Parameters of the `SAMPLE (form, font, or logo)` command**

Parameter	Description
<i>file-name</i>	Specifies the name of a file to be sample printed. Masking is permitted. Using a mask sends samples to the stackers instead of to the sample tray.
<i>mask</i>	A generic file name which specifies a multiple number of files. Within a mask, one or more question marks (?) and/or asterisks (*) are substituted for valid alphanumeric characters. The name of each file on the disk is compared to the mask. When the file name equals the mask, the sample is printed. In file name comparisons, a question mark or an asterisk compares equal to any character. If the last character of the mask is an asterisk, any subsequent positions compare as equal. If the last character of the mask is a question mark, any subsequent positions must be blank (such as files with names longer than the mask are not sampled). These samples go to the stacker.
<i>file-type</i>	Specifies the type of file to be sample printed. The <i>file-type</i> can be FRM, FNT, IMG or LGO.
<i>D</i>	Specifies that the file should be printed on both sides of a single sheet. If a single <i>file-name</i> is requested for print, it is printed on both sides of a page. If all items of a <i>file-type</i> are printed, then each <i>file-name</i> is printed only once on alternate sides of duplex-printed sheets. The default is simplex.
<i>copies</i>	Specifies the number of copies to be printed.

If you sample all the forms on the Xerox DocuPrint 180 with `SAMPLE *.FRM`, the LPS prints the forms in two sets in order to change the pitch mode for printing only once. The first set consists of all forms which are smaller than 10.21 inches (259 mm); the second set consists of all forms which are larger than 10.21 inches (259 mm).

If all forms are printed with the command `SAMPLE *.FRM`, the form name is printed on the edge of each form.

If you invoke SAMPLE for fonts that are too large to be sampled (such as all the characters cannot be printed on a single page), the following message displays: OS6952 Imaging error. Possible data off page. Page integrity cannot be guaranteed, therefore page will not be printed. In order to sample large fonts, use the font editor utility to modify the original font to a manageable number of characters per line.

To abort a SAMPLE operation that is using a mask option, enter "ABORT job-id" to abort the job completely. Refer to the ABORT section in this chapter for more information.

SAMPLE (multiple copies of a form)

Use this command to request up to 32,767 copies of a form.

Syntax `SAMPLE file-name. TST [,D]`

Parameters Table 11-2 shows the parameters of the `SAMPLE (multiple copies of a form)` command.

Table 11-2. **Parameters of the `SAMPLE (multiple copies of a form)` command**

Parameter	Description
<i>file-name</i>	Indicates the name of a form file cataloged in the FRM library.
TST	Specifies that the file prints continuously.
D	Specifies that the file be printed on both sides of a single sheet (duplex).

The specified form prints continuously until you enter the `STOP` or `ABORT` command to halt printing.

SAMPLE (extended .IMG options)

The following instructions pertain to using a SAMPLE command to print samples of .IMG files.

Use a SAMPLE command with extended options to print samples of .IMG files. The extended options describe the horizontal/vertical positioning, orientation, and scaling of the image for printing. For each graphic specified, the SAMPLE command prints two pages: the image page and the descriptor page with metric data (refer to figure 11-2).

Syntax SAMPLE [{*file-name* | *mask*}] .IMG [,D] [,*copies*] [, ([*vpos* [*in* | *cm* | *dots*]) , [*hpos* [*in* | *cm* | *dots*]] [.L | ,P] [,*n* [/d]])]

Parameters Table 11-3 shows the parameters of the SAMPLE (extended .IMG options) command.

Table 11-3. **Parameters of the SAMPLE (extended .IMG options) command**

Parameter	Description
<i>file-name .IMG</i>	Identifies the graphic to be sampled. If only the IMG file type is specified, all .IMG files are sampled. Masking is permitted. The name of each .IMG file on the disk is compared to the mask, and a sample of the graphic is printed for each equal comparison.
D	Specifies that the graphic image is to be printed on the front of the sheet and the graphic's statistical information on the back. The default is simplex which will have the graphic image on the first page and the graphic's statistical information on the second page.
<i>copies</i>	Specifies the number of copies to be printed. The default is one copy.
<i>vpos</i>	An extended option that specifies the vertical position of the top edge of the graphic, relative to the physical page origin in <i>dots</i> , centimeters (<i>cm</i>), or inches (<i>in</i>), specified as a decimal number with up to three digits to the right of the decimal point. If no units are specified, inches are assumed. The default is the top left of the page or zero inches.
<i>hpos</i>	An extended option which specifies the horizontal position of the left edge of the graphic relative to the physical page origin. The form of this specification (<i>cm</i> , <i>in</i> , or <i>dots</i>) is the same as for <i>vpos</i> . The default is the top left of the page or zero inches.

Table 11-3. **Parameters of the SAMPLE (extended .IMG options) command** (continued)

Parameter	Description
L and P	An extended option which specifies a reference orientation (landscape and portrait) for positioning the graphic. The default orientation may be specified in the image file header. If none is specified there, the default is portrait.
<i>n/d</i>	An extended option which specifies the reference scale factor. Refer to the "Scaling" section in the "Graphics on the LPS" chapter for further information.



Note: Scaling and/or positioning factors which position the graphic off the page are modified to position the graphic on the page. The erroneous specification is flagged with an asterisk to its right on the sample descriptor page.

Examples

The following are some examples of SAMPLE commands that can be used to print .IMG files.

- SAM * .IMG
Sample print all .IMG files with all default options in effect.
- SAM XYZ .IMG
Sample print graphic file XYZ.
- SAM X* .IMG ,D
Sample print all .IMG files whose names begin with X. Print each sample in duplex.
- SAM X?Z .IMG
Sample print all .IMG files with three-character names beginning with X and ending with Z.
- SAM XYZ .IMG ,D , 3
Sample print three copies of graphic file XYZ and print each copy in duplex mode.
- SAM XYZ .IMG , (1 , 1)
Sample print graphic file XYZ with graphic image displaced one inch vertically and one inch horizontally from the page origin defined by default or from the orientation of the graphic.
- SAM XYZ .IMG ,D , (, , 2)
Sample print graphic file XYZ in duplex mode with a reference scale factor of 2.
- SAM XYZ .IMG , (2.54 CM , 10.16 CM , P)
Sample print graphic file XYZ with the graphic image displaced 2.54 centimeters vertically and 10.16 centimeters horizontally from the portrait page origin.
- SAM XYZ .IMG ,D , 4 , (375 DOTS , 1225 DOTS , L , 1 / 2)
Sample print four copies of graphic file XYZ in duplex mode with the graphic image displaced 375 dots vertically and 1225 dots horizontally from the landscape page origin and printed with a reference scale factor of 1/2.

Each graphic sample comprises two pages. The first contains the graphic image itself. The second contains a statistical description of the graphic, its file storage requirement, and LPS command parameters. A sample descriptor page is shown in figure 11-1.

Restrictions

The following restrictions apply to using the SAMPLE command:

- You cannot print samples of a job's pages when: the job is output to the sample tray; the job JDE/JDL includes the command SECURITY ABNORMAL; the job uses ordered stock.
- You can select a complete or an abbreviated descriptor page if the image file is not a secure file. (Only an abbreviated descriptor page can be printed for a secure file.) The following data is reported in the complete descriptor page only if it applies to the image:
 - Image generator
 - Image identification
 - Original scan parameters
 - Status.
- The complete descriptor page generated by the SAMPLE command does not include all the data in the .IMG file header block. The descriptor page generated by the HARDCOPY command includes all the .IMG file header block data.

Figure 11-1. **Sample page descriptor for image file**

```

SAMPLE PRINT OF IMAGE TC14CC

PRINTED ON: 31MAY98 09:08:00

IMAGE FILE HEADER INFORMATION:
  IMAGE NAME -
  IMAGE HEIGHT (PIXELS) - 352
  IMAGE WIDTH (PIXELS) - 467
  COMPRESSED IMAGE SIZE (BYTES) - 1804
  COMPRESSION FACTOR - 11
  IMAGE RESOLUTION (DOTS/INCH) - 300/2
STORAGE REQUIREMENTS:
  IMAGE MEMORY (K-BITS) - 14.06
  DISK ID - CENTRAL SYSTEM
  DISK BLOCKS - 5
SAMPLE SPECIFICATIONS:
  COMMAND LINE - SAMPLE TC14CC.IMG,4,D,(2.3,1.5,P,2)
  POSITIONING MODE - P
  ORIGIN - 750, 450
  SCALING - 4
    
```

Figure 11-2. **Sample page descriptor for image file**

```

SAMPLE PRINT OF FONT P0612A.FNT

PRINTED ON : 30APR98 14:59:29

FONT CHARACTERISTICS :
  ORIENTATION : PORTRAIT
  PITCH : FIXED
  CHARACTER CELL HEIGHT/LINE SPACING (DOTS) : 37
  NOMINAL CHARACTER HEIGHT (DOTS) : 37
  TOP OF CELL - BASELINE (DOTS) : 29
  BASELINE - BOTTOM OF CELL (DOTS) : 8
  AVERAGE CHARACTER WIDTH (DOTS) : 22
  HIGHEST CHARACTER CODE (HEX) : 60
  FONT MATRIX SIZE (K-BITS) : 26.72
  DISKID- EL SEGUNDO SYSTEM B126
  DISK STORAGE (BLOCKS) : 10
  FONT CATALOG NUMBER : P0612A REVISION : A
  FONT CONTROL NUMBER : 880699 A02

CHARACTER CODE :
X'10' TO X'1F'
X'20' TO X'2F'  !"#$%&'()*+,-./
X'30' TO X'3F'  0123456789:;^
X'40' TO X'4F'  ABCDEFGHIJKLMNO
X'50' TO X'5F'  PQRSTUVWXYZ _
X'60' TO X'6F'  ¢
    
```

HARDCOPY

The HARDCOPY command is similar to the SAMPLE command when printing graphic files (such as any sampled or hard-copied graphic file results in an image page and a descriptor page). The image page for a specific graphic file is printed in the same manner whether it is sampled or hard-copied, but the contents and formats of the descriptor pages differ. (Refer to and compare figures 11-1, 11-3 and 11-4.) The hardcopy descriptor page is more comprehensive and permits printing all the information that is available in a graphic file header block. The information printed on a sample descriptor page is a subset of the information that can appear on a hardcopy descriptor page. Descriptor pages are numbered evenly, and any errors entered on the Interpress exception page are referenced to the page number on the descriptor page. Figure 11-3 is a descriptor page for a graphic file received from a host computer using the Xerox XPPI-XDGI software. Figure 11-4 is a descriptor page for a graphic file received from a 7650 Pro Imager.

Hardcopy print is not allowed if the image is protected by a security command of the ABNORMAL statement (such as ABNORMAL SECURITY=YES).

Syntax HARDCOPY [{*file-name* | *mask*}] .IMG [,D] [,copies] [([vpos [*in* | *cm* | *dots*]] , [hpos [*in* | *cm* | *dots*]] [.L | ,P] [,n [/d]])]

Parameters Table 11-4 shows the parameters of the HARDCOPY command.

Table 11-4. **Parameters of the HARDCOPY command**

Parameter	Description
<i>file-name</i> .IMG	Identifies the graphic to be hard-copied. If only the .IMG file type is specified, all image files are hard-copied. If any other file type is specified, the command is executed as if it were a SAMPLE command with a corresponding descriptor page. The name of each .IMG file on the disk is compared to the mask and a hardcopy of the graphic is printed for each equal comparison. Each hard-copied graphic is positioned by using default rather than specified positioning parameters.
<i>D</i>	Specifies that the graphic's statistical information (the descriptive data page) is printed on the back of the sheet and the graphic image page on the front. The default is simplex. With simplex, two sheets are printed: the first sheet will have the image, the second sheet will be the descriptive data page.
<i>copies</i>	Specifies the number of copies desired. The default is one copy.
<i>vpos</i>	Specifies the vertical position of the top edge of the graphic, relative to the physical page origin in dots, centimeters (<i>cm</i>), or inches (<i>in</i>), specified as a decimal number with up to three digits to the right of the decimal point. If no units are specified, inches are assumed. The default is two-thirds up the page and centered in the given orientation.
<i>hpos</i>	Specifies the horizontal position of the left edge of the graphic relative to the physical page origin. The form of this specification (<i>cm</i> , <i>in</i> , or <i>dots</i>) is the same as for <i>vpos</i> . The default is two-thirds up the page and centered in the given orientation.
L and P	Specifies a reference orientation (landscape and portrait) for positioning the graphic. The default orientation may be specified in the image file image header. If none is specified there, the default is portrait.
<i>n/d</i>	Specifies the reference scale factor. Refer to the "Scaling" section in the "Graphics on the LPS" chapter for more information.

Table 11-5. Metrics page template for hardcopy/proof copy of graphic files

	Date: yyyy mm dd (YYYY MM DD)
	Time: hh:mm:ss
	Zone: <xx.xx
* Local Name: xxxxxx.IMG	*Disk I.D.: disk-name
* Path Name: <xxxxxx> Printer> IMG> xxxxxx>	
Print Command Line: HARDCOPY ... (or PROOF COPY)	
Sample Specifications:	
Printed Height xxxx units	Image Height xxxx pixels
Printed Width xxxx units	Image Width xxxx pixels
Image Orientation:	<i>N/A, Portrait, and so on.</i>
Upper Left Origin:	<i>yyyy pixels, xxxx pixels; Portrait/Landscape</i>
Graphics Origin:	<i>yyyy pixels, xxxx pixels; Portrait/Landscape</i>
Applied Scale:	<i>n/d (or n)</i>
Resolution (printer/generator):	<i>nnn/ddd (pixels per inch/pixels per inch)</i>
Effective Scale:	<i>n</i>
Local Storage Requirements:	
Disk Usage:	<i>xxx blocks of 512 bytes</i>
Image Memory Required:	<i>x.xxxxxx megabits of 4096 bit blocks</i>
Compressed Image Size:	<i>xxxxxx bytes</i>
Compression Factor:	<i>xx.xx</i>
*Image Generator:	
* Creation Time:	<i>date, time</i>
* Process:	
* Machine I.D.:	
*Image Identification:	
* Set Name:	
* Set Version:	
* Set Number:	
* Partition Number: alphanumeric strings	
* Max Partition Number:	
* User I.D.:	
* Description:	
* Graphics Generator:	
*Original Scan Parameters:	
* Right Left Top Bottom Width Height	
*Crops: xxx units xxx units xxx units xxx units xxx units xxx units	
*Sizing: Anamorphic/Proportional xxx%	
*Density: xxx	
*Contrast: xxx	
*Screen: x or none	
*Status: NNNNNNNNNNNN	
Note:	
* These lines are omitted if there is no data. <i>Italicized</i> items are replaced with variable data. <i>xxx units</i> replaced by: <i>xx.xx in, xx.xx pi, xxx mm, xxx pt</i> ; default is <i>xxxx pixels</i> .	

Table 11-6. Metrics page example for hardcopy of graphic file

	Date: 1998 02 01 (YYYY MM DD)
	Time: 09:01:23
	Zone: +08:00
@Local Name: CIRCLE.IMG Disk I.D.: V9GX SEPTEMBER 1, 1997	
Image Format: IMG 2.00 (Signature 4837)	
Print Command Line: HARDCOPY 405099.IMG,D...	
Sample Specifications:	
Printed Height 2400 points	Image Height 800 pixels
Printed Width 2256 points	Image Width 752 pixels
Image Orientation:	Portrait
Upper Left Origin:	100 pixels, 300 pixels;
Graphics Origin:	50 pixels, 300 pixels;
Applied Scale:	7/5
Resolution (printer/generator):	300/150 (pixels per inch/pixels per inch)
Effective Scale:	3
Local Storage Requirements:	
Disk Usage:	47 blocks of 512 bytes
Image Memory Required:	0.184 megabits of 4096 bit blocks
Compressed Image Size:	23174 bytes
Compression Factor:	3.25
Image Generator:	
Creation Time:	06-JUN-1997 11:05
Process:	XPPI
Machine I.D.:	123-245-89
*Image Identification:	
Set Name:	CIRCLE
Set Version:	1
Set Number:	1
Partition Number:alphanumeric strings	1
Max Partition Number:	1
User I.D.:	T. Meltonka
Description:	XPPI Sample for Resolution Test
Graphics Generator:	SAS-GRAPH Version 2
Status: O OK	

Table 11-7. Metrics page example for proof hardcopy of graphic file

	Date: 1998 02 01 (YYYY MM DD)					
	Time: 09:01:23					
	Zone: +08:00					
Path Name: <FOZZI> Printer>IMG>PRFIMG						
Image Format: IMG 1.00 (Signature 19661)						
Print Command Line: Proof Copy						
Sample Specifications:						
Printed Height	2.67 IN	Image Height	800 pixels			
Printed Width	2.67 IN	Image Width	800 pixels			
Image Orientation:	Portrait					
Upper Left Origin:	1250 pixels, 875 pixels; Portrait					
Graphics Origin:	1250 pixels, 875 pixels; Portrait					
Applied Scale:	1					
Resolution (printer/generator):	300/300 (pixels per inch/pixels per inch)					
Effective Scale:	1					
Local Storage Requirements:						
Disk Usage:	47 blocks of 512 bytes					
Image Memory Required:	0.184 megabits of 4096 bit blocks					
Compressed Image Size:	23174 bytes					
Compression Factor:	3.45					
Image Generator:						
Creation Time:	1998 06 06 11:05:00-0800					
Process:	7650 PROIMAGE					
Machine I.D.:	FO-2E-15-6C-77-9B					
Image Identification:						
Set Name:	PRFIMG					
Graphics Generator:	V3.00					
Status: O OK						
Original Scan Parameters:						
Crops:	Right	Left	Top	Bottom	Width	Height
	0.00 IN	2.00 IN	0.00 IN	2.00 IN	2.00 IN	2.00 IN
Sizing:	Proportional					
Density:	40					
Contrast:	100					
Screen:	8					

Controlling offline input for print jobs

The following section describes how to control offline input for print jobs.

SPACE (position tape or cartridge by report)

Use the SPACE command with the REPORTS option to position a tape or cartridge by logical reports within the limits of the start-of-job position and the end-of-data. You must enter the START command to initiate a job before you enter the SPACE command.

Syntax 1 SPACE TO REPORT *n*

The SPACE TO REPORT *n* syntax winds the tape up to the report specified by *n*. For example, SPACE TO REPORT 2 positions the tape (or cartridge) at the beginning of the second report on the tape.

Syntax 2 SPACE *n* REPORTS

The SPACE *n* REPORTS syntax winds the tape (or cartridge) from its current position *n* more reports. For example, if a tape is at the third report and you enter SPACE 2 REPORTS, the tape winds to the beginning of the fifth report.

Parameters Table 11-8 shows the parameters of the SPACE (position tape or cartridge by report) command.

Table 11-8. **Parameters of the SPACE (position tape or cartridge by report) command**

Parameter	Description
<i>n</i>	A decimal integer in the range -32768 to +32767. A positive number indicates forward report spacing, and a negative number, backward spacing. The tape or cartridge cannot be positioned before the start-of-job position or after end-of-data. If the start-of-tape position is on another volume, the tape cannot be positioned to it. The default is to forward space over one report.
REPORTS	Indicates positioning by logical report units.

Restrictions You can issue SPACE any time after initiating a job if:

- Input is stopped by the STOP INPUT command; it is confirmed by the message INPUT STOPPED, or when the system is suspended.
- The START command did not include the REPORTS option.

When report spacing is complete, the system prompts with the message: MOVE or SPACE function is complete. Enter 'CON I' or 'CON JDE, JDL' to start next report. You can then enter a CONTINUE command or another SPACE command.

SPACE (position report printing by page)

During report printing, use the PAGE command with the SPACE option to position pages forward and backward within the current report. Prior to selecting the SPACE PAGE command, you must stop output with the STOP or STOP O command. After the SPACE PAGE command, resume printing with the CONTINUE or CONTINUE O command.

Syntax SPACE [*n* PAGES]

Parameters Table 11-9 shows the parameters of the SPACE (position report printing by page) command.

Table 11-9. **Parameters of the SPACE (position report printing by page) command**

Parameter	Description
<i>n</i>	A decimal integer in the range -32768 to +32767. A positive <i>n</i> results in forward spacing over <i>n</i> pages unless end-of-report is encountered first. If end-of-report is encountered, no further forward spacing is allowed until printing is resumed. A negative <i>n</i> results in backward spacing over <i>n</i> pages unless the beginning of the report is encountered. If beginning-of-report is encountered, no further backward spacing is allowed until printing is resumed. The default value for <i>n</i> is 1. The integer <i>n</i> must include all pages, including the system generated pages. Output provides a summary of pages skipped upon the completion of the SPACE PAGE function.
PAGES	Indicates positioning by page units.

Output confirms that page spacing has begun by displaying one of the following messages:

OS0610 Page spacing forward

or

OS0620 Page spacing backward

During forward page spacing, output may catch up with input. Output displays a message indicating that this has occurred and resumes spacing when the catchup condition no longer exists. While output is caught up, output accepts and processes other commands with the exception of the SAMPLE command. A sample cannot be obtained while output is spacing. Once page spacing has begun, you cannot discontinue page spacing before the specified number of pages have been spaced without aborting the report.

An accounting page is produced at the end of the report, indicating that the page spacing was performed.

When page spacing is complete, output displays the following messages:

```
OS0630 Page spacing complete.  n Total pages spaced.
      N Simplex data pages.  n Simplex nondata pages.
      N Duplex data pages.  n Duplex nondata pages spaced.
```

Restrictions on page spacing

Page spacing is not executed when:

- ABNORMAL SECURITY = YES is coded in the JDE/JDL in effect. The OS1420 Page spacing not allowed by JDE message displays.
- The current report is N-1. The OS1421 Page spacing not allowed for N-1 report message displays:
- Output is not stopped. The OS2840 OUTPUT must be stopped before page spacing can be done message displays.
- Output is not currently working on a job. The OS2715 Command ignored...retry later message displays.
- Backward spacing is not allowed if the system is printing the last (or only) copy of a collated report.

Duplex page spacing

For duplex printing, page spacing must be done in pairs (such as front and back). Therefore, if you enter an odd page count in a space command, the total number of pages actually spaced is 1 greater. Page spacing logic counts physical sides of sheets; both sides in duplex, only one side in simplex. No consideration is given to logical pages on any physical side.

As an example: a report that requires 5 sheets in duplex, 10 sheets in simplex, and each side consists of four logical pages (4-up) for a total of 40 logical pages. The report has a space command of 3 pages. In simplex, the system will deliver 7 sheets, for a total of 28 logical pages. In duplex, the system increments the odd space count to 4, such that 3 sheets will be delivered (24 logical pages).

MOVE (position tape or cartridge by block or file)

Use the MOVE command to physically position an open-reel tape by a specified number of files or blocks either forward or backward. You can use MOVE during job processing, but input must be stopped first with STOP I or due to an error stop. If no job is active, you can enter the command directly.

Syntax MOVE [TD*n*]*n* [FILES | BLOCKS]

Parameters Table 11-10 shows the parameters of the MOVE (position tape or cartridge by block or file) command.

Table 11-10. **Parameters of the MOVE (position tape or cartridge by block or file) command**

Parameter	Description
<i>TDn</i>	Specifies the physical device unit (0 or 1 or 2). The default value is TD0:.
<i>n</i>	A decimal integer in the range -32768 to +32767. A positive number indicates forward movement and a negative number indicates backward movement. The default is 1.
<i>FILES</i>	Indicates positioning by files. A positive <i>n</i> moves the tape forward over <i>n</i> tape marks unless end-of-volume (EOV) terminates the tape (or cartridge) movement. A negative <i>n</i> moves the tape (or cartridge) backward over <i>n</i> tape marks unless BOT terminates the movement. File positioning is the default.
<i>BLOCKS</i>	Indicates positioning by blocks. A positive <i>n</i> moves the tape forward over <i>n</i> interblock tape gaps unless EOT or a tape mark terminates tape movement. A negative <i>n</i> moves the tape backward over <i>n</i> blocks unless BOT or a tape mark terminates tape movement.

If a job is in process (while MOVE is invoked), the system requests a CONTINUE I to resume. The CONTINUE I is not necessary if a job has not been initiated.

Whenever a tape (or cartridge) move is halted by reaching a tape mark, the tape (or cartridge) is always positioned at the block one greater than the absolute block number of the tape mark. This is done regardless of the direction of the move.

Restriction MOVE BLOCKS is not allowed if ABNORMAL SECURITY=YES is coded in the job descriptor library of the report being processed.

Rewind a tape or cartridge

Use either of these commands to rewind a tape or cartridge.

Syntax1 REWIND [TAPE|CARTRIDGE | TD*n*]

Syntax2 TAPE [CARTRIDGE | TD*n*] REWIND

Parameters Table 11-11 shows parameters of the REWIND command.

Table 11-11. **Parameters of the REWIND command**

Parameter	Description
TAPE	Specifies TD0.
CARTRIDGE	Specifies TD1.
TD <i>n</i>	Specifies a tape device number.

Unload a tape or cartridge

Use this command to unload a tape or cartridge.

Syntax TAPE [CARTRIDGE | TD*n*] UNLOAD

Parameters Table 11-12 shows parameters of the UNLOAD command.

Table 11-12. **Parameters of the UNLOAD command**

Parameter	Description
TAPE	Specifies TD0.
CARTRIDGE	Specifies TD1.
TD <i>n</i>	Specifies a tape device number.

Controlling online input for print jobs

Commands for online 3211/4245 mode are essentially the same as those for offline. The commands unique to online are described below. Refer to the "HIP processing on the LPS" chapter for HIP online operations information.

ONLINE

The ONLINE command moves the LPS from offline to online status. The system displays NOT READY (such as until a START command is entered, data input is stopped and the host does not transmit).

Syntax ONLINE

DRAIN (print completed pages)

Use the DRAIN command to print all completely formatted pages without ending the job in progress. If the print file has five or fewer pages, these pages may not be printed.

Syntax DRAIN

ENDJOB (end online job)

Use the ENDJOB command to free the system to execute a START command. ENDJOB marks the current LPS job as terminated and prints all the remaining pages in the print file. Use the ENDJOB command when you wish to:

- Print a trailer banner to mark the end of a report
- Print a second trailer banner when the first is incompletely sent
- Work around multicopy disk saturation
- Respond to a MOUNT FORM message from an IBM host terminal
- Print the job's accounting sheet.

Syntax ENDJOB

OFFLINE

The OFFLINE command moves the LPS from online to offline status. After this command is entered, online input data may not be transmitted to the LPS.

Syntax OFFLINE

Printing user files

The following section describes commands used during the printing process.

Controlling print processing

The commands described below are used to control print processing.

START (start a print job)

Use the START command to initiate printing for a user's print job. Use the various options depending upon the input source and the output destination.

Offline syntax `START [[jde][,][jdl] [,][S | M] [,][copies] [,][[REPORTS:r1, r2,...][,][TDn] | [TDn]] | [DISC:file.ext | FORM=form.frm]]]]]`

Online syntax `START [[jde][,][jdl][,][copies][,FORM = form]]]]]`

- Restrictions** Restrictions on using the START commands are as follows:
- The START command parameters are positional and must be separated by commas. You must enter a comma to replace a parameter that is not specified. The options you specify in the START command override those specified in the job descriptor library. Refer to figure 11-6 for further details on override parameters.
 - If a specified font or form file fails a validity check during input, the system aborts the job and displays the following messages:


```
OS8852 Invalid font file header
OS8855 Invalid form file header
```
 - START with parameters does not execute when HIP print jobs are being processed.
 - You cannot specify a tape device when you have specified DISC or FORM.

Examples of START command usage follow the parameter definitions below. START command options are as follows:

Parameters Table 11-13 shows the parameters of the START (start a print job) command.

Table 11-13. **Parameters of the START (start a print job) command**

Parameter	Description
<i>jde</i>	A 1- to 6-character identifier for the job descriptor entry to be used in processing the job. If the <i>jde</i> option is not specified in the START command, the user-specified default (DFLT) of the job descriptor library is used.
<i>jdl</i>	A 1- to 6-character identifier of the job descriptor library for the print job. It must be listed in the <i>jdl</i> file directory. If the <i>jdl</i> option is not specified, the default is used. The default <i>jdl</i> must have been created by the user with an identifier of DFAULT in the SYSTEM command.
S M	Parameters which control the processing mode of a job. The two modes are: single report (S) and multiple report (M). The default mode is multiple report. Single report mode halts the system after the processing of each report to allow the operator to select the job set-up parameters for the next report by entering a new START command. Whenever the S mode is invoked in the START command, the system processes only one report at a time, and the accounting sheet will call the report "REPORT 1." If the REPORTS option of the START command is invoked at the same time as the S mode, the REPORT option takes precedence over the S mode option. If multiple reports are listed in the REPORTS option, all the reports are processed in multiple report mode, except the last report, which is processed in single report mode. Multiple report mode allows all reports in all files to be processed continuously. Processing automatically sequences from report to report, file to file, volume to volume until all reports have been processed.
<i>copies</i>	Prints a specified number of copies of a report. It is an override of the value specified in the job descriptor entry, and also overrides a DJDE command if present in the input data stream.

Table 11-13. **Parameters of the START (start a print job) command** (continued)

Parameter	Description
REPORTS: r_1, r_2, \dots	Allows the user to specify the sequence and subset of reports to be processed. This option is available only when processing tape input. Only reports specified are printed. For r_1, r_2, \dots , the user specifies numeric values or ranges of values, representing the print order of reports. A range is specified as $n-m$, where n and m are the first and last reports in the range to process, respectively. For example, entering REPORTS: 6,1-3,5,4 causes the sixth report to print first, followed by the first through third, followed by the fifth and then the fourth. If the job contains more than 6 reports, they would not be processed. The REPORTS option is also useful for printing one report of a multiple report tape. This saves the step of spacing (refer to SPACE command) over reports not needed. A maximum of 14 values is allowed. A range of values uses 2 of the 14 maximum values allowed.
<i>TDn</i>	Specifies the tape device being used.
<i>DISC</i>	Indicates you want to print a file stored on the system disk
FORM	Allows the user to specify a form used for the job. This option overrides any form specification in the job descriptor entry (FORMS command of the OUTPUT statement and the RFORM command of the ROUTE statement). The specified form must be on disk in the FRM directory.

Examples The following are some examples of START command syntax.

- START J12,H2SYS

This command starts a print job using the H2SYS job descriptor library and the job descriptor entry J12. It runs in multiple report mode (by default) and prints the number of copies as specified in the J12 job descriptor entry. The job descriptor library, H2SYS, must reside in the JDL directory. After the START command is initiated, several messages display to inform the operator of the print jobs in progress. In most instances, only one more command entry is required. An example of an offline interaction as shown in figure 11-3.

Figure 11-3. Starting an offline print job

```
OS1000 READY FOR COMMANDS hh:mm:ss
START J12,H2SYS
OS1010 Starting job 00003
OS2010 Mount input tape; "CONTINUE I" when ready
CONTINUE I
OS0010 Resuming INPUT
OS0020 Resuming OUTPUT
OS1020 Job 00003 has completed input phase
OS1030 Job 00003 has completed printing
OS1000 READY FOR COMMANDS hh:mm:ss
```

- **START J12,H2SYS,,5**

This command is the same as in example 1 with the exception that five copies are requested. The value of 5 entered for copies overrides the value specified in the J12 job descriptor entry. Note that a comma replaces the unspecified mode option; therefore, the default mode, multiple report, takes effect.

- **START**

No options are specified, so the START command defaults take effect. The default for the job descriptor library is DFAULT, which must exist in the JDL directory. The job descriptor entry used is DFLT, which must exist in the DFAULT JDL. The command START, DFAULT has the same effect.

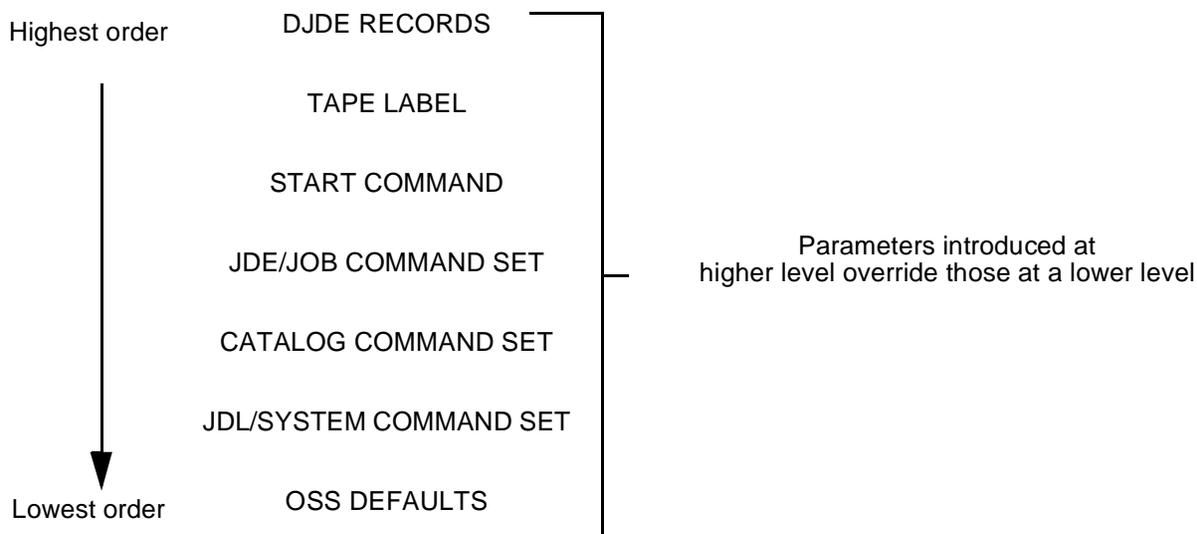
- **START J12,H2SYS,,2,REPORTS:3-4**

This command reprints 2 copies of the third and fourth reports on the data tape.

- **START ,ONLINE,,,FORM=GBAR**

This command processes online data according to the DFLT JDE in the online .JDL file, using the GBAR form.

Figure 11-4. Hierarchy of replacement



* One exception to this hierarchy is that the copies option on the START command overrides copies DJDE from the input data stream.

START (print a tape or cartridge dump)

You can use the START command to print a dump of a tape or cartridge.

Syntax START *n*,DUMP

Parameters Table 11-14 shows the parameters of the START (print a tape or cartridge dump) command.

Table 11-14. Parameters of the START (print a tape or cartridge dump) command

Parameter	Description
<i>n</i>	A value from one to six representing the operating system which wrote the tape.
<i>n</i> =1	For a H6000 BCD tape (OCTAL dump)
<i>n</i> =2	For an EBCDIC tape (HEX dump)
<i>n</i> =3	For an ASCII tape (HEX dump)
<i>n</i> =4	For a H2000 BCD tape (OCTAL dump)
<i>n</i> =5	For an IBM BCD tape (OCTAL dump)
<i>n</i> =6	For a UNIVAC tape (OCTAL dump)
<i>n</i> =7	For an ICL1900 tape

Refer to the *Tape Formats Manual* for more information.



Note: You can also use the START command to print a file created by COMPRESS. Refer to the COMPRESS section in the "Managing LPS resources" chapter.

STOP (stop processing)

Use the STOP command to suspend input and/or output processing. The system can process input for one job and output for another job since input and output are controlled separately. Stopping input suspends the processing of input data but allows output to continue. Stopping output suspends printing after cycling out the paper path, but allows processing of input data to continue.

Syntax STOP [I | O]

Parameters Table 11-15 shows the parameters of the STOP (stop processing) command.

Table 11-15. **Parameters of the STOP (stop processing) command**

Parameter	Description
(no parameter)	Suspends both input and output activities.
I	Suspends input activities until resumed by entering a CONTINUE command. The job number and report number of the stopped job are displayed.
O	Suspends output activities until a CONTINUE command is entered.



Note: The printer control console <STOP> key, suspends output activities until a CONTINUE command is entered. The <STOP> key on the keyboard stops both input and output activities.

When the <STOP> key is pressed, the system will complete and deliver all of the sheets that are currently in the paper path, including all scheduled sheets that have not yet been fed.

CONTINUE (resume printing)

Use the CONTINUE command to resume input and/or printer output processing suspended by the STOP command or by the system report of an error condition. Since the system controls input and output activities separately, continuing input allows input data to process and spool to disk even if output is stopped. Continuing output prints or images existing pages even if input is stopped.

Syntax CONTINUE [I | O]

Parameters Table 11-16 shows the parameters of the CONTINUE (resume printing) command.

Table 11-16. **Parameters of the CONTINUE (resume printing) command**

Parameter	Description
(no parameter)	Resumes both input and output processing.
I	Resumes input processing.
O	Resumes output processing.



Note: The printer control console <CONTINUE> key resumes output processing. The <CONTINUE> key on the keyboard resumes both input and output activities.

ABORT (abort job)

The ABORT command removes a job from the system, aborts a report, or discontinues certain system processing: input, output, or both.

The system handles job aborts you initiate as follows:

- A job being input is stopped after the next data read and is removed from the queue, but data written to disk from previously processed jobs is printed.
- A job in the print queue is deleted.
- A job being printed is truncated, but remaining jobs in the queue are printed.
- A job is not deleted if it has one report in the input queue and other reports in the output queue: only the report in the input queue is deleted when the next print job starts.

Processing

If a tape job aborts, the system does not rewind the tape. Therefore, you should issue the REWIND command from the system controller PC UI or terminal before entering the next START command, since a label error can occur if the tape is not at the expected position.

When ABORT I stops input processing, output processing continues. Data processed by input is deleted; data processed by output is not affected. When input processing resumes, it begins with the next complete report.

When ABORT O stops output processing, input processing continues. Data processed by output is deleted; data processed only by input is not affected. When output processing resumes, it begins with the next complete report.

When ABORT stops both input and output processing, then both input and output data are deleted. Printing resumes only after you issue a START command.



Note: When no print processes are running, the ABORT command may stop other types of processes.

Other system activities

You can stop other system activities with the ABORT command. These include the language processors (PDL and FDL), tasks invoked by name, and other system level commands.

Syntax

ABORT {job-id | O | I}

Parameters

Table 11-17 shows the parameters of the ABORT (abort job) command.

Table 11-17. Parameters of the ABORT (abort job) command

Parameter	Description
<i>none</i>	Causes the currently active nonprint task (FDL, PDL, and so on) to abort. If either input or output tasks are active, the ABORT key-in must have a parameter (I, O, or <i>job-id</i>). Failure to specify a required parameter displays OS1190 Invalid parameter, command ignored.
<i>job-id</i>	Identifies the job identification number the system assigned to a job in response to the START command. To abort a SAMPLE operation that is using a mask option, enter ABORT <i>job-id</i> to abort the job completely.
O	Aborts the report currently being printed.
I	Aborts the report currently in input. It is typically used in response to an error message and solicited key-in sequence (for example NO SPACE FOR FILE XXXXXX.IMG. NO PURGE PENDING. INPUT STOPPED. ENTER CON I OR ABO I).

When an offline job is aborted by input (due to an error condition) and output has begun printing the job, input stops processing. Output prints whatever has been formatted up to the point of aborting. An offline job which has not begun to print will not start. An online job aborted by input is printed up to the point of aborting.

Examples The following are examples of the ABORT command:

- ABORT 5
OS0900 Job 5 aborted
- ABORT
OS0950 Task aborted

The system accounting file is updated for the aborted job. User accounting pages are printed if output has been initiated for the job. They are delivered for each report that input has read. If output has not been initiated, no accounting page is delivered.

The system prints an accounting sheet with a completion code and a message for each aborted job.

LOCKOUT

The Xerox DocuPrint 180 LPS Diagnostic Job Interrupt mode protects job integrity in two ways. The LPS enters this mode when the service technician issues the LOCKOUT START command, or when a service technician enters diagnostic routines on the printer. In the Diagnostic Job Interrupt mode, the LPS first stops the currently running job and stores it in a temporary area. Then the LPS accepts only the LOCKOUT END, LOGON, and CONTINUE commands from the UI. This ensures that the diagnostics run by the service technician cannot affect operator input or vice versa. The LPS then samples forms with printer diagnostic routines. Finally, when the service technician issues the LOCKOUT END command, the LPS automatically reactivates the interrupted job.

If LOCKOUT is issued when the LPS is running online jobs, input processing sends a message to the host indicating that the LPS must interrupt job transmission. Although not stated by this message, the message also indicates that the LPS will automatically receive data from the host once the LOCKOUT END command is issued.

Syntax LOCKOUT START

Restrictions LOCKOUT START and LOCKOUT END function only if the service technician enters the required password at the system prompt.

Processing The system controller checkpoints all print file data and identifies an alternative recovery area. All printer messages are sent to the Test Pattern Job (TPJ) with OCS. TPJ processes sample test pattern requests from the printer. Then TPJ displays the following messages as appropriate:

OS1011 Starting diagnostic print job.

OS1821 Cannot get LUN assignment. TPJ aborted.

OS1822 Cannot write to diagnostic print file. TPJ aborted.

OS1823 Problem with file __>filename. TPJ aborted.

OS1824 Duplex not allowed in this configuration. TPJ aborted.

OS1990 Preformat form-using system default paper size.

OS1991 No test pattern available for specified source size.

For more information on these messages, refer to the *Xerox DocuPrint 180 LPS Message Guide*.

After completing the diagnostics, the service technician enters:

LOCKOUT END



Note: The service technician is prompted for the required password before the system executes LOCKOUT END.

The system exits the diagnostic interrupt mode. Output restores all recovery data, and communication with the input task resumes.

RESET (stop all system processing)

Use the RESET command to stop all system processing activity. RESET removes all print jobs from the job queue and cycles any pages in the printer paper path to the stacker. If the input device is tape, the tape remains positioned as it was when the RESET command executed. The system remains in an idle state until you enter the next command.

Syntax RESET

Processing If the system is idle, the following displays on the screen:

OS1000 READY FOR COMMANDS

If the system is not idle, the following displays:

Reset will clear job queue. Confirm with 'Y' or 'N' «CR='Y'»

If RESET is invoked during job processing, accounting data is recorded in the system accounting file only for reports completely processed and printed.

RESET aborts a job being transmitted with HIP and gives the job an aborted status. HIP stops processing, but remains online. If the aborted HIP job is partially printed, the unprinted portion is deleted.

To terminate a specific job before its completion, enter the ABORT command, rather than the RESET command. Due to the powerful and abrupt nature of the RESET command, you may wish to restrict command usage (RCU) of this command to a higher logon level, thereby preventing use of the command at lower operating levels. (Refer to the "Limiting task execution privileges" section of the "Setting LPS parameters" chapter for more information.)

CLEAR

Use the CLEAR command to view a display of the current fault, hint, or information messages for the printer. Before displaying these messages, the LPS resets faults which do not require your physical intervention. A fault or hint message sometimes prompts you to enter CLEAR. You can also use CLEAR to take the printer out of the Energy Saver mode.

Syntax CLEAR

FHI (fault, hint, information)

Use this command to request display of the printer's fault, hint, and information arrays.

Syntax FHI

Checking print job status

The following section describes commands used to verify job status during printing.

Using the JOBS command

The various uses of the JOBS command are described below.

JOBS (job queue/report status)

Press the <JOBS STATUS> key or enter the JOBS command to request a display of system and job status information with detailed report accounting information. System status shows the state of the system (idle, online, offline, active tasks). Job status shows information about all jobs known to the system. Expect a delay of a few seconds when you request a job's status because responses may be delayed until end-of-report processing completes. You can also transmit the report information to other media for further processing (refer to the "ACCOUNT" section in the "Tracking system activity" chapter) and/or reinitialize it (refer to the SFS command).

Syntax JOBS [QUEUE | ALL | *n*]

The system displays the following:

- Date, time of day, and system status (online, offline, or idle)
- Any jobs queued for input
- Status of report currently in input
- Status of report currently in output.

The system displays additional data when you specify an option.

Parameters Table 11-18 shows the parameters of the JOBS (job queue/report status) command.

Table 11-18. **Parameters of the JOBS (job queue/report status) command**

Parameter	Description
QUEUE	All reports queued for output.
ALL	All reports queued for output plus the status of up to 100 completed reports. This display is terminated by entering an X (NULL) command. Refer to the "NULL" section in this chapter for more information.
<i>n</i>	All reports queued for output plus the status of up to <i>n</i> completed reports, where <i>n</i> cannot exceed 100.

In response to this command, the system outputs a formatted display of one or more of the appropriate messages:

Figure 11-5. **Jobs display.**

```

System status mm/dd/yy hh:mm:ss
Approximately XXX unused entries in the accounting file*
HOST INTERFACE PROCESSOR IS ACTIVE
System is idle**
System is online
System is offline
INPUT is stopped
OUTPUT is stopped
Job queue is empty
JOB ID xxxx JDE xxxxxx QUEUED FOR INPUT
JOB CUST RPT REPORT      JDE/JDL  CPY TOTAL PAGES  STATUS
NO. JOB# NO. NAME        CNT PAGES PRNTD MSG   MSG#
xxxx xxxxxx xxxx xxxxxxxxxxxxxxxxxxxx xxxxxx/xxxxxxx xxxx xxxxxx xxxxxx xxxxxxxx xxxx
    
```

* Indicates the number of unused entries in the status file, displayed in response to the JOBS command if the 2,000 report entry file was selected and facilitates management of the status file when the report accounting feature is being used (refer to the "Establishing accounts and copying account data to tape" section in the "Tracking system activity" chapter).

** Indicates input and output are not in memory. No jobs are in the job queue.

The report status information (the x's above) is repeated for each report. If one screen is not sufficient to display the requested information, the message ENTER RETURN TO CONTINUE appears. A new title line, followed by more report status information, is displayed if the <ENTER> key is pressed. If the <ENTER> key is not pressed, the job status display terminates.

Table 11-19 defines the information that is displayed for each report.

Table 11-19. **Definition of information displayed in JOBS status report**

Report Message	Definition
JOB NO.	The job identification number supplied by the system. This job ID number is referenced when using the ABORT command.
CUST JOB#	One of the following: <ul style="list-style-type: none"> • Host job number as defined by the user with the HJOBNO command of the BANNER statement if the system is online. • FILE ID (same as on accounting page) if the system is offline. • Blank if the information is not available.
RPT NO.	The report number as currently defined on the accounting page.
REPORT NAME	The report name as defined by the online user with the BANNER statement command, HRPTNA; or by the offline user with the RSTACK statement command, HRPTNA, or the DISC keyword in the START command. (Refer to the <i>LPS PDL Reference</i> for more information.)
JDE/JDL	The JDE/JDL name keyed in to start the report or the DJDE JDE/JDL encountered.
CPY CNT	The maximum number of copies if the report is in input, queued for output, or completed. If the report is in output, CPY CNT is the number of copies remaining to be printed. CPY CNT can differ from the copy count in the JSL due to the application of a DJDE.
TOTAL PAGES	The current number of pages written to disk if the report is in input. If the report is queued for output or being printed, TOTAL PAGES is the total number of pages in the report.
PAGES PRNTD	The number of pages completed if printing or completed. Otherwise, this value is zero.
STATUS MSG	The report status, which is one of the following:
INPUT	If the report is in the input phase and has not begun to print.

Table 11-19. **Definition of information displayed in JOBS status report** (continued)

Report Message	Definition
QUEUED	If the report completed the input phase and is waiting to print.
IN/PRNTG	If the report is in the input and print phases and has begun to print.
PRINTING	If the report has completed the input phase and is printing.
DONE	If the report is completed.
RESET	If the report was aborted by the RESET command.
RCVY N/S	If the operator did not request recovery on rollover.
ABORTED	If the job/report was aborted in input or output.
STATUS MSG#	One of the following:
	Completion code as on accounting page (is blank if completion is normal).
	Error message number if report aborted. This message number's form is xxxx and corresponds with an error message (OSXXXX) in the <i>Xerox DocuPrint 180 LPS Message Guide</i> .
	CS displays while the report is processing if the report is copy sensitive.



Note: Even though the TOTAL PAGES and PAGES PRNTD fields can display only a six-digit number, the status file maintains the correct page values.



Note: Only the job status header, containing no jobs status information, is displayed when a job is active. This occurs when both input and output processing are between reports, such as neither task is currently processing a report. Pressing the <JOB STATUS> key or entering the JOBS command after output processing has completed also produces a blank job status header, regardless of the current printer activity.

For additional report information and the structure of the status file, refer to the "Tracking system activity" chapter.

JOBS ALL options

The following section describes the JOBS ALL command.

Use the JOBS ALL options to request information for all jobs with the status you specify.

Syntax JOBS ALL [DONE | RESET | ABORT | RCVY | *n*]

Parameters Table 11-20 shows the parameters of the JOBS ALL command.

Table 11-20. **Parameters of the JOBS ALL command**

Parameter	Description
DONE	Jobs which are done
RESET	Jobs aborted by the RESET command
ABORT	Jobs aborted by the ABORT command
RCVY	Jobs with RCVY/NS status because the operator did not request recovery on rollover
<i>n</i>	The last one to 100 jobs.

The system displays the information described above in the Job queue/report status section of this chapter. If no jobs meet the JOBS ALL statement criteria, then the system displays OSXXXX No status entries to display for selected JOBS command.

NULL (X)

This command is intrinsically a null operation or NO-OP command. As such, it terminates a display without causing an INVALID COMMAND message. X can also be initiated from the editor. The command has no parameters.

Syntax X

Using the PSTATUS command

The following section describes how to monitor printer status.

PSTATUS

You can request printer status by:

- Pressing the <PRNTRSTATUS> key (on the terminal)
- Pressing the function key (on the PC UI)
- Or selecting Printer Status... from the Print Tasks window.

Expect a delay of a few seconds when you request the status during printing. If you have the system controller terminal (non-U.S. markets only) or are currently running jobs using command files, the following appears on the screen:

Figure 11-6. **PSTATUS display**

IOT State:		Clientlayer:		
Current feed mode:		feed mode		
Current select mode:		select mode		
Tray:	Status:	Paper level:	Paper size:	
1	Selected	1/2 full	11.5 by 8.0	
2	Ready	Full	11.5 by 8.0	
3	Ready	Full	11.5 by 8.0	
4	Ready	1/4 full	14.0 by 11.5	
7	Ready			
Bin:	Status:	Paper level:	Paper size:	Container type:
A	Selected	1/2 full	8.5 by 11.0	
B	Ready	Empty	8.5 by 11.0	
E	Ready	Unavailable	Unavailable	

If you invoke this command when the LPS is not cycling, tray status is displayed according to the current feed command. The next page to be printed is not considered. That is, if the current feed command is a cluster command with mixed tray (paper) sizes, the cluster tray that is displayed may not necessarily be the cluster tray with the correct paper for the currently active job.

With the terminal, the Xerox DocuPrint 180 system controller displays the following information for each tray and bin when you select Printer Status: client layer state, paper tray state, paper tray size, and paper tray levels. It also displays the following information for each bin: output bin state, output bin size, and output bin levels. In addition, the IOT (printer) state and current feed tray are identified.

Table 11-21 illustrates paper tray states.

Table 11-21. **Paper tray states.**

Selected	This tray is now feeding the current job. It is one of the In use trays. Only one of the In use trays has Selected status at any given moment.
Ready	These trays do not feed the current report, but can be selected to feed following reports.
Busy	Tray elevator is rising or descending. It cannot be selected until elevator completes trip.
Not Ready	Elevator is down, thus tray cannot be selected.

Table 11-22 illustrates paper tray sizes.

Table 11-22. **Paper tray sizes**

x by y	A supported paper size. Refer to SET TRAY section in the "Setting up print jobs" chapter.
--------	---

Table 11-23 illustrates paper tray levels.

Table 11-23. **Paper tray levels (by 20 lb. ream stack height equivalent)**

Tray 1	Empty, 1/4 full, 1/2 full, 3/4 full, 1, full
Tray 2	Empty, 1/4 full, 1/2 full, 3/4 full, full
Trays 3, 4, 5, 6	Empty, 1/4 full, 1/2 full, 3/4 full, 1 ream, 2 reams, 3 reams, 4 reams, full

Table 11-24 illustrates output bin size.

Table 11-24. **Output bin size**

Size	The LPS automatically assigns a report to bins set to receive the stock size used by the report.
------	--

Table 11-25 below lists output bin states.

Table 11-25. **Output bin states**

Selected	This bin is now receiving the current job. It is one of the In use bins. Only one of the In use bins has Selected status at any given moment.
Ready	These bins are available for use, but are not currently being used. You may use the SELECT command to activate the bin, or SELECT AUTO may activate the bin when the other bin fills.
Busy	Bin elevator is rising or descending. It cannot be selected until elevator completes trip.
Not Ready	Elevator is down, thus tray cannot be selected.
Unavailable	The bin cannot be selected.

Table 11-26 shows output bin sizes.

Table 11-26. **Output bin sizes**

None	Bin is empty and containers are not being used; thus any size output can be stacked in the bin.
x by y	The size of the sheets or container currently in the bin. Only reports printed on this paper size can currently be stacked in the bin.

Table 11-27 shows output bin levels.

Table 11-27. **Output bin levels (by 20 lb. ream stack height equivalent)**

Empty	No sheets are in the bin.
n	Approximately n reams in the bin.
Full	Bin is full by the current bin criteria. It is ready to be emptied.

The PC UI displays a printer mimic. The mimic provides the tray status, tray level, tray size, bin status, bin level, and bin size information in graphic form. Refer to the *Xerox DocuPrint 180 LPS Operator Guide* for additional information.

Responding to the Attention light or alarm

This section describes the function of the Attention light and alarm.

ATTENTION

The Attention light shines (or the alarm sounds) to warn you of print job conditions which require your attention either immediately or soon. The steady light (and the alarm) warn that the print process will cycle down soon due to printer conditions (such as low toner or paper supply) unless you take the appropriate preventive measures. The flashing light (and the alarm) signal that the printer is cycling down due to a printer condition, which may require operator intervention. You must check whether the light signals an information only message or signals the need for operator action.

By default the steady and flashing states are enabled for the Attention light on the system. The alarm is also enabled. When a system warning or error triggers the enabled light and alarm, they remain in their activated state until you turn them off.

Disable light and alarm syntax

CLEAR

The system tries to clear all printer faults and requests new fault and status information. The light and the alarm are both turned off, but may be turned on again by the system if the faults have not cleared.

Light disable syntax

You can disable one or both states for the Attention light.

ATTENTION LIGHT DISABLE STEADY

The light does not shine for printer information messages, but it flashes for printer fault messages.

ATTENTION LIGHT DISABLE FLASHING

The light shines for printer information messages, but it does not flash for printer fault messages.

ATTENTION ALL OFF

The light does not shine for printer information messages, and it does not flash for printer fault messages. If new faults are detected, the light turns on.

Light enable syntax

You can also enable the Attention light.

ATTENTION LIGHT ENABLE FLASHING or

ATTENTION LIGHT ENABLE STEADY



Note: You must disable the Attention light before you can enable it. The command forms are:

ATTENTION LIGHT DISABLE

ATTENTION LIGHT ENABLE

Alarm syntax You can enable or disable the alarm.

ATTENTION ALARM ENABLE {,0-10}

0-10 sets the loudness of the alarm (0 is quiet; 10 is maximum volume).

The alarm sounds for both printer information and fault messages.

ATTENTION ALARM DISABLE

The alarm does not sound for printer information nor for fault messages.

A. System command summary

Table A-1 provides a summary of system commands for the Xerox DocuPrint 180 Laser Printing System.

Table A-1. System command summary

Command	LPS	Function	Off**	On	Res***
ABORT	All	Aborts current job or abort report activity.	Y	Y	x
ACCOUNT	All	Creates and updates accounting file	I	N	1
ALIGN	All	Aligns laser image with paper	Y	Y	1
ATTENTION	4135/4635/ DP180	Enables and disables the Attention light.	Y	Y	
BINFULL	4135/4635/ DP 180	Limits quantity of output sent to a bin.	Y	Y	
CLEAR	4135/4635/ DP 180	Clears all faults, and redisplay outstanding faults, hints, info arrays.	Y	Y	
COMPRESS*	All	Compresses file directories and disk files.	I	N	2
CONTINUE	All	Resumes input/output activities.	Y	Y	x
COPY	All	Copies files from disk/disk to disk/tape.	I	N	1
CLP	All	Assigns preferred trays to a cluster.	Y	Y	
CLU	All	Assigns current trays to a cluster.	Y	Y	
DALIGN	4135/4635/ DP 180	Sets diagnostic job alignment values.	Y	Y	
DARKNESS	4135/4635/ DP 180	Makes the printed image lighter or darker.	Y	Y	
DELETE	All	Deletes files from system disk.	I	N	1
DRAIN	All	Prints all completed online pages.	N	Y	1
DSR*	All	Saves/restores disks.	I	N	2
EDGE	4135/4635/ DP 180	Decreases the density of print on page edge.	Y	Y	
EDI	All	Creates and modifies files.	I	N	1
ENDJOB	All	Prints all unimaged online pages.	N	Y	1
ENERGY	4135/4635/ DP 180	Sets default time for energy-saving mode.	Y	Y	
FCHECK	All	Displays of data about disk file allocations.	I	N	3
FDL	All	Compiles forms.	I	N	1
FEDIT	All	Creates and maintains fonts.	I	N	1

Table A-1. **System command summary** (continued)

Command	LPS	Function	Off**	On	Res***
FEED	All	Selects active paper tray	Y	Y	1
FHI	4135/4635/ DP 180	Displays the LPS fault, hint, and information arrays.	Y	Y	
FID	All	Creates a prototype command file.	I	N	1
FILE	All	Lists file directories on LPS display.	I	N	I
FIX	All	Modifies fonts and logos to LPS configuration.	I	N	1
FLOPPY	All	Initiates floppy commands.	I	N	1
FONTS	All	Sets maximum number of active fonts.	I	N	1
FORMS	All	Sets maximum number of active forms.	I	N	1
GRAPHIC	All	Sets maximum number of active graphics.	I	N	1
HARDCOPY	All	Prints hardcopy of identified graphics.	I	N	1
HIP	All	Initiates Host Interface Processor commands.	I	N	1
JOBS	All	Displays job status information.	Y	Y	x
LIST	All	Lists hardcopy of file directories.	I	N	1
LOCKOUT	4135/4635/ DP 180	Allows UI to process only LOGON, CONTINUE, and LOCKOUT END commands.	Y	Y	Y
LOGON	All	Determines access privileges.	Y	Y	x
MOVE	All	Positions tape by blocks or files.	Y	N	1
OCS	All	Displays key version numbers.	Y	Y	1
OFFLINE	All	Sets LPS to offline.	N	Y	1
ONLINE	All	Sets LPS to online.	Y	N	1
OSTK	4135/4635/ DP 180	Controls printing on sequenced sheets.	Y	Y	x
PDL	All	Compiles JSL files.	I	N	1
PFEED	4135/4635/ DP 180	Specifies input tray for purge sheets.	Y	Y	x
PSC	All	Downloads the CSI, RIP, and SCSI microcode to the appropriate PWB.	I	N	2
PURGE*	All	Clears data in unused disk sectors of print file.	I	N	2
RCU	All	Restricts command usage.	I	N	x
REA	All	Reallocates print file.	I	N	2
REPORT	All	Reports of system activity and accounting.	I	N	1
RESET	All	Forces all system activity to cease.	Y	Y	1

Table A-1. **System command summary** (continued)

Command	LPS	Function	Off**	On	Res***
REVIEW	All	Reviews file directories with optional delete.	I	N	1
REWIND	All	Rewinds tape.	Y	N	1
RNAME	All	Renames a file from one file name to another.	I	N	1
SAMPLE	All	Prints samples and test patterns.	Y	Y	1
SECURE	All	Secures a file against unauthorized access.	I	N	5
SELECT	All	Selects active output tray.	Y	Y	1
SET CONTAINER	4135/4635/ DP 180	Defines container size.	Y	Y	
SET TRAY	4135/4635/ DP 180	Defines the size for a feeder zone.	Y	Y	
SETTIME*	All	Sets current time.	I	N	2
SFS	All	Creates, clears, stores, displays, prints status file information.	I	N	1
SFT*	All	Performs single file transfer.	I	N	2
SPACE	All	Positions tape by logical reports or pages.	Y	N	1
START	All	Initiates print job.	Y	Y	x
STOP	All	Suspends input/output activities.	Y	Y	x
TAPE	All	Manipulates LPS-labeled tape.	I	N	1
TIME (date)	All	Displays time of day, date on display.	Y	Y	1
TIME (status)	All	Displays current time and status.	Y	Y	1
TRANS	All	Overrides job control of transparency feed.	Y	Y	
TYPE	All	Displays source file without having to use editor commands.	I	N	1
UNLOAD	4135/4635/ DP 180	Lowers and raises trays and bins.	Y	Y	
X	All	No operation.	Y	Y	1

* These commands may not be executed while HIP is active.

** In the Off (offline) column, I (for idle state) means that no printing is in progress and the system must be offline.

A Y in column On (online) or Off (offline) means the command is available when the printing system is in that state; an N in either column means the command is not available in that mode.

*** The Res column defines the system default operator restriction level associated with each command. An x in the Res column means that the command can be executed at level 1 but cannot be restricted.

B. Editor command summary

Table B-1 summarizes editor commands.

Table B-1. **Editor command summary**

File/tape	Function
CE	Same as CLEAR, END sequence.
CLEAR	Clears the contents of the working file.
CONVERT	Sets mode to convert.
COPY	Copies files from tape/disk to disk/tape.
DELETE	Deletes a file from the system.
EDIT	Begins an editing session.
END	Ends an editing session.
FID	Creates prototype command file.
FILE	Lists disk file catalog on the LPS display.
FIX	Modifies fonts and logos.
FLOPPY	Initiates floppy disk utility commands.
GET	Gets an existing file and copies in into working storage.
GPCE	Same as GET, PRINT, CLEAR and END sequence.
KEYS	Displays beginning and ending line numbers.
LIST	Lists disk file catalog on the printer.
MERGE	Merges an existing file into current working storage.
NOCONVERT	Sets mode to no convert.
PCE	Same as PRINT, CLEAR and END sequence.
PSCE	Same as PRINT, SAVE, CLEAR and END sequence.
REVIEW	Provides sequential listing of file names with optional delete.
RNAME	Changes the name and/or type of a file.
SAVE	Saves the contents of working storage as a permanent file.
SCE	Same as SAVE, CLEAR and END sequence.
TAPE	Initiates tape utility commands.
TYPE	Displays source files without having to use editor commands.
Record	Function
DISPLAY	Displays source lines on the operator's terminal.

Table B-1. **Editor command summary** (continued)

File/tape	Function
DUPLICATE	Duplicates lines from a file into another area of the file.
FIND	Finds and displays lines containing a specified text string.
INSERT	Inserts lines using a specified line number.
MODIFY	Specifies a range of lines for intra-record commands.
MOVE	Moves lines from one place to another.
PRINT	Produces hardcopy of the working file to sample or output tray.
REMOVE	Removes a specified group of lines.
RENUMBER	Renumbers the working file.
REPLACE	Specifies that lines are to be removed and others inserted.
SORT	Alphabetizes the records in the file currently being edited.
STEP	Specifies the modify range to be the next record.
Intrarecord	Function
D	Deletes string.
F	Inserts following string.
O	Overlays string.
P	Inserts prior to string.
S or C	Substitutes string.

C. Font editor command summary

Table C-1 summarizes font editor keywords and commands.

Table C-1. **Font editor keyword/command summary**

Keyword/command	Function
CHARACTER	One of three keywords sets the selection mode (CHARACTER, HEXADECIMAL, or OCTAL). CHARACTER signifies that subsequent characters included in the new font file will be expressed as character literals.
CLOSE	Explicitly terminates access to the current secondary source font file.
END	Explicitly terminates font editing session.
FEDIT	Invokes font editor.
HEXADECIMAL	One of three keywords sets the selection mode (CHARACTER, HEXADECIMAL, or OCTAL). HEXADECIMAL signifies that subsequent characters included in the new font file will be expressed as hexadecimal values.
INCLUDE	Precedes the list of characters to be copied from the currently accessible source font file to the new font file without character code reassignments (refer to RECODE).
INPUT	Specifies the primary source font file.
OCTAL	One of three keywords sets the selection mode (CHARACTER, HEXADECIMAL, or OCTAL). OCTAL signifies that subsequent characters included in the new font file will be expressed as octal values.
OPEN	Specifies secondary source font file.
OUTPUT	Stores the current contents of the working storage as a new and permanent font file.
RECODE	Specifies the list of characters which are copied from the source font file to the new font file with new character code assignments (refer to INCLUDE).
REORDER	Changes accessing sequence of characters within a specified font file as required to achieve compatibility with Interpress standards.
SUBSTITUTE	Allows the operator to replace any blank with another character.
#	In addition to being the symbol for the pound sign, this character is also used to signify shifting from uppercase to lowercase, or vice versa, alternately. This use is operative only in character mode.



Note: In addition to the above-listed keywords, the space character, hyphen, and comma are recognized as delimiters by the font editor.



Note: Whenever the font editor has been invoked and is ready to accept a keyword command, the font editor prompt `FEDIT>` is displayed.

D. Command file command summary

Table D-1 provides a summary of command file commands.

Table D-1. **Command file commands**

Command	Function
.ASK	Alters symbol settings.
.ASKS	Substitutes character string for variable name.
.ASKV	Assigns symbol integer values.
.BCP	Branches if correct processor.
.BEZ	Branches if symbol is zero.
.BF/.BR/.BT/.BX	Specifies branching instructions.
.BNZ	Branches if symbol is not zero.
.CHAIN	Chains editor command files.
.DEC	Decrements a symbol.
.DELAY	Delays for a specified time.
.EXIT	Terminates a command file.
.HOME	Clears LPS display.
.INC	Increments a symbol.
.NOSHOW	Discontinues displaying dot commands.
.PAUSE	Suspends execution of command file.
.SETC	Sets symbol to current logon class.
.SETF	Sets symbol false.
.SETT	Sets symbol true.
.SETV	Sets symbol to a value.
.SHOW	Displays dot commands.

E. Character code assignment tables

This appendix contains figures which depict various character code assignment tables.

IBM BCD table (6-bit representation)

Figure E-1. IBM BCD table (6-bit representation)

		Most Significant Bits								
		Octal (columns)	0	1	2	3	4	5	6	7
Least Significant Bits	(rows) Binary	000	001	010	011	100	101	110	111	
	0	000	8		Y	-	Q	&	H	
	1	001	1	9	/	Z	J	R	A	I
	2	010	2	0	S		K		B	&
	3	011	3	#	T	,	L	\$	C	.
	4	100	4	@	U	%	M	*	D	<
	5	101	5	'	V	_	N)	E	(
	6	110	6	=	W	>	O	;	F	+
7	111	7		X	?	P		G		

Note: 0'20' is the official blank character.
† Corresponds to BCD code set used by IBM users and defined by PDL command CODE = IBMBCD.

Honeywell 200/2000 BCD table (6-bit representation)

Figure E-2. Honeywell 200/2000 BCD table (6-bit representation)

		Most Significant Bits								
		Octal (columns)	0	1	2	3	4	5	6	7
Least Significant Bits	(rows)	Binary	000	001	010	011	100	101	110	111
	0	000	0	8	+	H	-	Q	<	Y
	1	001	1	9	A	I	J	R	/	Z
	2	010	2	'	B	;	K	#	S	@
	3	011	3	=	C	.	L	\$	T	,
	4	100	4	:	D)	M	*	U	(
	5	101	5		E	%	N	"	V	{
	6	110	6	>	F	[O]	W	}
	7	111	7	&	G	?	P	!	X	

Notes: 0'15' is the official blank character; 0'77' is the padding character.
^tCorresponds to BCD code set used by Honeywell 200/2000 users and defined by PDL command CODE = H2BCD.

Honeywell 6000 BCD table (6-bit representation)

Figure E-3. Honeywell 6000 BCD table (6-bit representation)

		Most Significant Bits								
Octal (columns)		0	1	2	3	4	5	6	7	
(rows)	Binary	000	001	010	011	100	101	110	111	
Least Significant Bits	0	000	0	8	space	H	$\overline{1} / \wedge 2$	Q	+	Y
	1	001	1	9	A	I	J	R	/	Z
	2	010	2	[B	&	K	-	S	_
	3	011	3	#	C	.	L	\$	T	,
	4	100	4	@	D]	M	*	U	%
	5	101	5	:	E	(N)	V	=
	6	110	6	>	F	<	O	;	W	"
	7	111	7	?	G	\	P	'	X	!

Notes: 1. Usual BCD Character
2. Printer Character

[†]Corresponds to BCD code set used by Honeywell 600/6000 series SSF tapes and defined by PDL command CODE = BCD or CODE = H6BCD.

Fielddata translation table

Figure E-4. Honeywell 6000 BCD table (6-bit representation)

		Most Significant Bits								
		0	1	2	3	4	5	6	7	
	Octal (columns)	0	1	2	3	4	5	6	7	
	(rows) Binary	000	001	010	011	100	101	110	111	
Least Significant Bits	0	000	@	C	K	S)	*	0	8
	1	001	[D	L	T	-	(1	9
	2	010]	E	M	U	+	%	2	'
	3	011	#	F	N	V	<	:	3	;
	4	100	^	G	O	W	=	?	4	/
	5	101	(blank)	H	P	X	>	!	5	.
	6	110	A	I	Q	Y	&	,	6	"
	7	111	B	J	R	Z	\$	\.	7	-

UNIVAC ASCII character set

Figure E-5. UNIVAC ASCII character set

Octal	Character	Octal	Character	Octal	Character
040	(blank)	100	@	140	\
041	!	101	A	141	a
042	"	102	B	142	b
043	#	103	C	143	c
044	\$	104	D	144	d
045	%	105	E	145	e
046	&	106	F	146	f
047	/	107	G	147	g
050	(110	H	150	h
051)	111	I	151	i
052	*	112	J	152	j
053	+	113	K	153	k
054	,	114	L	154	l
055	-	115	M	155	m
056	.	116	N	156	n
057	/	117	O	157	o
060	0	120	P	160	p
061	1	121	Q	161	q
062	2	122	R	162	r
063	3	123	S	163	s
064	4	124	T	164	t
065	5	125	U	165	u
066	6	126	V	166	v
067	7	127	W	167	w
070	8	130	X	170	x
071	9	131	Y	171	y
072	:	132	Z	172	z
073	;	133	[173	}
074	<	134	\	174	
075	=	135]	175	}
076	>	136	^	176	~
077	?	137	_	177	(null)

Standard ASCII character set

Figure E-6. Standard ASCII character set

		MOST SIGNIFICANT BITS															
Hexadecimal		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LEAST SIGNIFICANT BITS	0	0000			0	@	P	e	p								
	1	0001		!	1	A	Q	a	q								
	2	0010		"	2	B	R	b	r								
	3	0011		#	3	C	S	c	s								
	4	0100		\$	4	D	T	d	t								
	5	0101		%	5	E	U	e	u								
	6	0110		&	6	F	V	f	v								
	7	0111		'	7	G	W	g	w								
	8	1000		(8	H	X	h	x								
	9	1001)	9	I	Y	i	y								
	A	1010	{	*	:	J	Z	j	z								
	B	1011		+	;	K	[k									
	C	1100	}	,	<	L	\	l									
	D	1101	~	-	=	M]	m									
	E	1110	¼	.	>	N	^	n									
	F	1111	½	/	?	O	_	o									

Standard EBCDIC character set

Figure E-7. Standard EBCDIC character set

		MOST SIGNIFICANT BITS																	
Hexadecimal		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
Binary		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
LEAST SIGNIFICANT BITS	0	0000						&	-		¼	½			{	}	\	0	
	1	0001							/		a	j			A	J		1	
	2	0010									b	k	s		B	K	S	2	
	3	0011									c	l	t		C	L	T	3	
	4	0100									d	m	u		D	M	U	4	
	5	0101									e	n	v		E	N	V	5	
	6	0110									f	o	w		F	O	W	6	
	7	0111									g	p	x		G	P	X	7	
	8	1000									h	q	y		H	Q	Y	8	
	9	1001								e	i	r	z		I	R	Z	9	
	A	1010					c	!	^	:									
	B	1011					.	\$,	#	{	}							
	C	1100					<	*	%	@									
	D	1101					()	_	'			[]					
	E	1110					+	;	>	=									
	F	1111						~	?	"									

F.

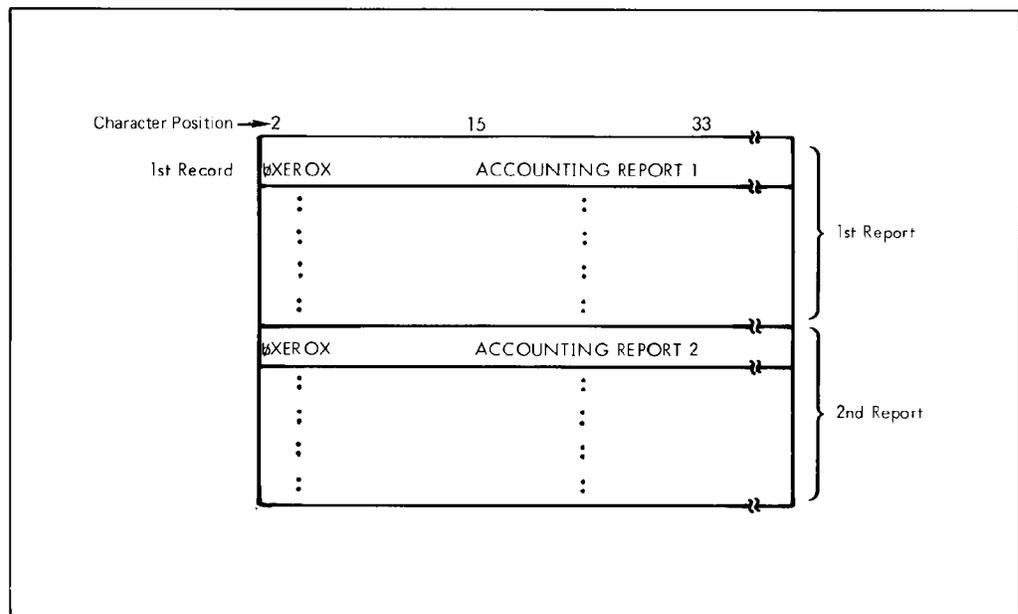
Page formatting

This appendix contains figures depicting various page orientations.

Page orientations

Figure F-1 shows positions and measurements for landscape and portrait modes.

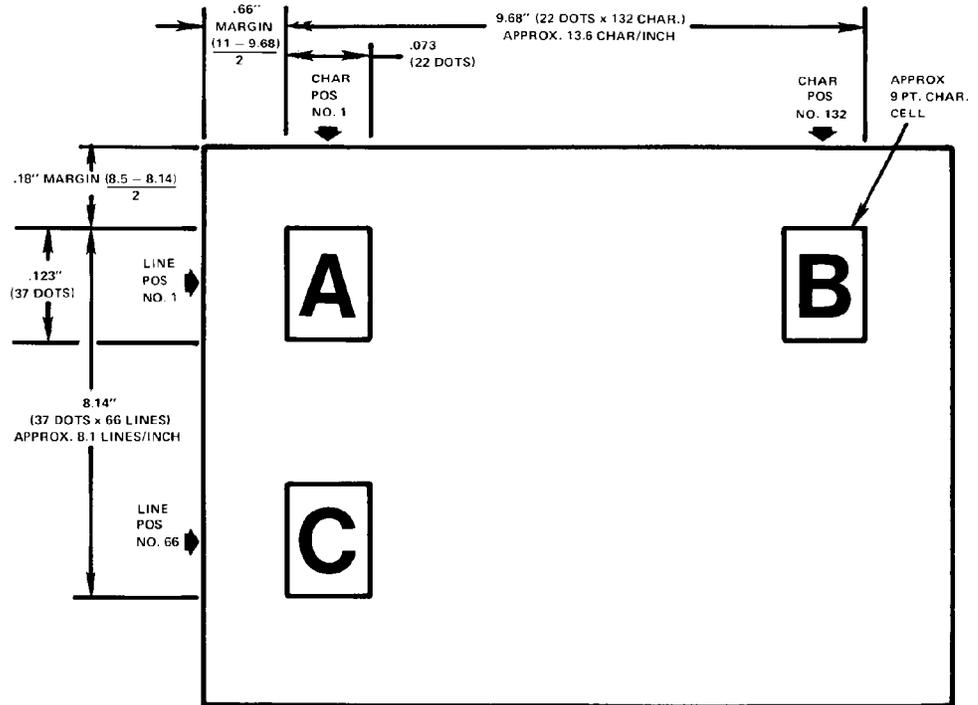
Figure F-1. **Vertical and horizontal positions in landscape and portrait modes**



Landscape orientation

Figure F-2 shows character positions and measurements for landscape orientation.

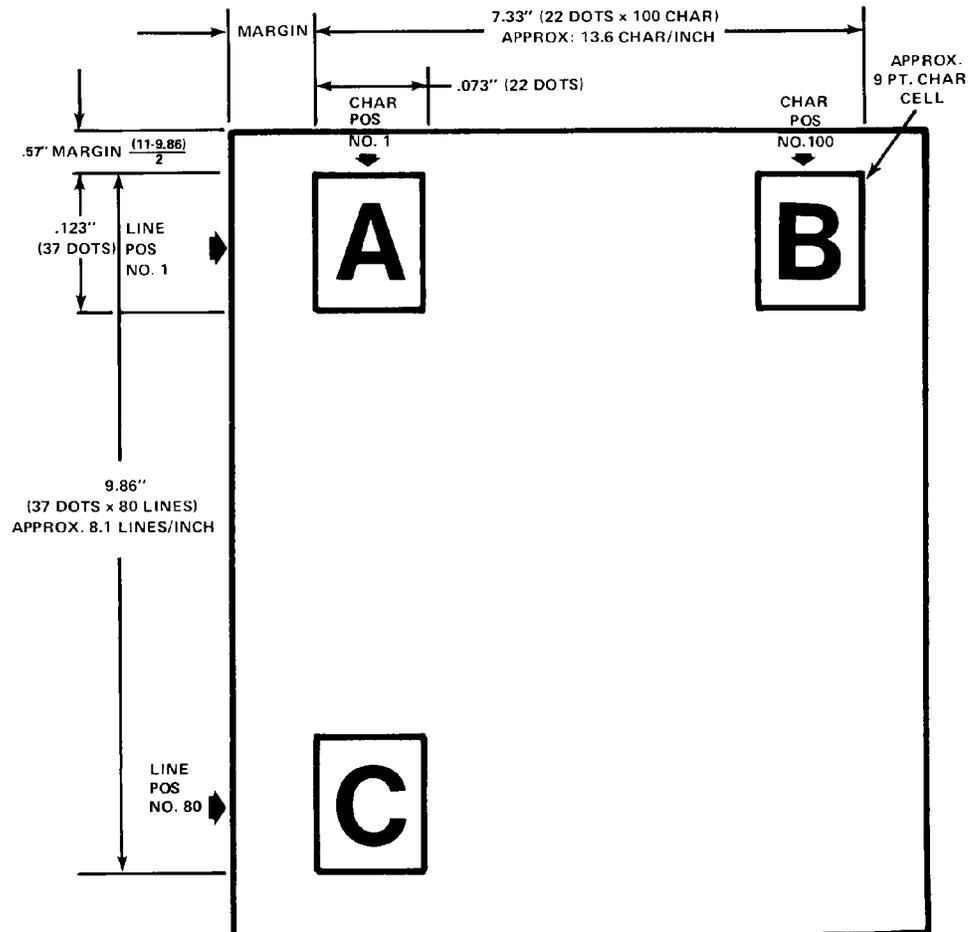
Figure F-2. **FMT1: equivalent impact printer format, 6 lines/inch**



Portrait orientation

Figure F-3 shows character positions and measurements for portrait orientation.

Figure F-3. **FMT6: equivalent impact printer format, 6 lines/ inch**



G. Crash recovery and dump

Severe problems sometimes make it impossible for the software to continue to function correctly. In these cases, the software will “crash” so that the conditions causing the problem can be documented and subsequently identified and resolved.

The following example illustrates such a crash and the operator actions which should be performed. Operator responses are **bolded** and underlined. If the system is running online, the host computer must cease communication with the system.

Messages which are conditional, depending on the configuration or the state of the system, are shaded followed by a brief remark enclosed in braces. For example, {this is a remark}.

The indented portion illustrates the occurrence of recovery messages. Since input and output processing are operating asynchronously, many combinations of actual messages may occur, such as input recovery to a page boundary combined with output recovery to a report boundary on a report which input has already completely processed.

The system attempts to reestablish the correct state of the system at the time of the crash to assure no loss of printed data. Accounting pages are produced for those reports which have completed input processing, even when no accounting has been requested. This provides a trace of unprinted reports not printed.

Note that more than one print job may be required to produce a complete Abort Analysis Dump. This depends on the type of system activity running at the time of the system crash.

When a sysgen rollover occurs following a yes response for both crash dump and job recovery, and a RESET key in, press **<CR>** or any other function key (such as CON, SAMPLE) to resume normal processing (such as print job is reset and character dispatcher continues).

Figure G-1. Crash recovery example

```

OS0510 INPUT stopped
  {if input processing was not already stopped at the time of the crash}
OS1215 INPUT stopped after orderly shutdown
  {if input processing was processing data at the time of the crash}
OS2575 System rollover, while online. Check host then enter 'C' C
  {displayed only if system was online at time of crash}
Loading Startup Task
XEROX XXXX
LASER PRINTING SYSTEM
Version n Revision m
Copyright (c) 1995 Xerox Corporation. All rights reserved.
Disk ID = thirtycharacterIDrecord
System recovery from <Q1>type<Q2> abort: TASK = task; CODE = code
  {refer to table 1 for possible values for type, task, and code}
Print abort data on printer for later analysis (Y/N)? Y
Print request will be queued
System mailbox initialized
Do you want to attempt automatic job recovery ('Y' OR 'N')? Y
  {is displayed only if a print job was in progress}
[Messages describing system and clientlayer initialization.]
OS1000 READY FOR COMMANDS.
OS1225 Automatic recovery initiated.
OS2960 Back up host to start of report or n pages, whichever is less.
OS4651 Output has recovered to page boundary.
OS2000 Enter 'CONTINUE O' to resume printing.
OS4650 INPUT has recovered to page boundary.
OS6651 OUTPUT has recovered to next report in job.
OS6650 INPUT has recovered to next report in job.
OS1000 Ready for commands
  {the system then completes processing the report(s) which were being
  processed at the time of the crash}
Performing abort analysis dump
OS1010 Starting job nnnnn
OS0020 Resuming OUTPUT
OS1020 Job nnnnn has completed input phase
OS1030 Job nnnnn has completed printing
ABORT Analysis dump complete
OS1000 Ready for commands

```

Table G-1 shows task and crash codes.

Table G-1. **Task and crash codes**

TYPE=	040 BMD DBG IDM MPR NAT OPR PWF SDN SDX SYS TSK	crash in location 0 exec bad memory deleted debug/diagnostic crash exec insufficient dynamic memory memory parity-executive no active task operator initiated crash (control-v) power failure executive stack depth negative stack depth exceeded disk access of bootstrap task initiated crash
TASK=	AIM BBU CPR DCD DJD DSR DUM EDI FCG FCP FDL FED HIP ICS IFU INPUT IPD MCP NULL OCS OSD OUTPUT PCH PDL PRF PSC REP RPT SCL SCU SDI SFS SFT SST STRTUP TACP TPJ XFR XLT	

Table G-1. **Task and crash codes** (continued)

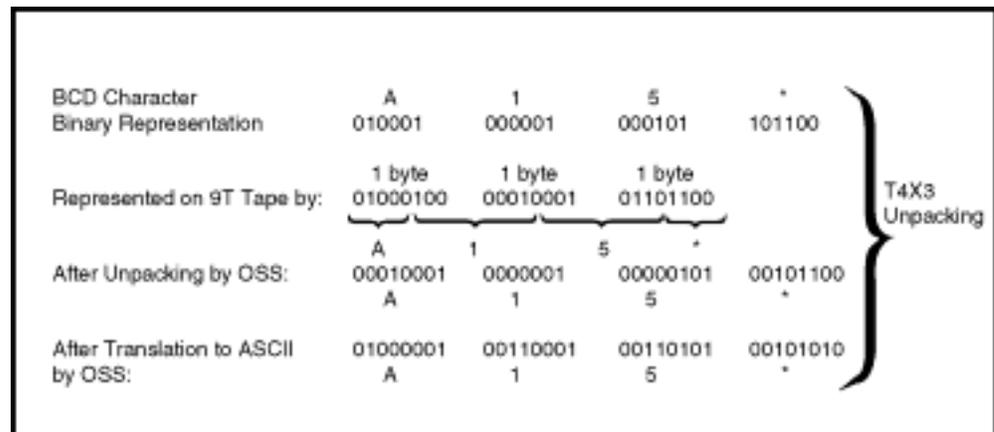
CODE=	ABO	task initiated abort
	AST	no room for AST trap
	BPT	break-point trap
	EMT	task EMT instruction
	ILG	illegal instruction
	IOP	I/O pending
	IOT	task IOT instruction
	LDF	loader failure
	MEM	non-mapped memory
	MPE	memory parity-task
	ODD	odd instruction location
	SST	no room for SST trap
	TRP	task TRP instruction

The following appendix contains notes on offline operation of the Xerox DocuPrint 180 Laser Printing System.

Input unpacking examples

Figure H-1 shows input unpacking examples.

Figure H-1. **Input unpacking examples**



Valid host computer/label specification table

Table H-1 shows valid host computer/label specifications.

Table H-1. **Valid host computer/label specification table**

Host type		Label specification					
		Unlabeled	ANSI	Standard	System print	COBOL	Undef
IBMOS		X	X	X			
IBMDOS		X	X	X			
GRASP POWER POWERVS		Label specification is ignored					
OSWTR			X	X			
US70		X	X	X			
XEROX		X	X	X			
B2500	Burroughs medium systems	X	X	X			
B2700		X	X	X			
B3500		X	X	X			
B3700		X	X	X			
B4700		X	X	X			
B6700	Burroughs large system	X	X	X			
H2000	Honeywell 200/2000 series			X	X	X	
H6000	Honeywell 600/6000 series			X			
DUMP		Label specification is ignored					
OCTDUMP		Label specification is ignored					
UNIVAC		X	X	X			
ANSI			X				
NCR		X		X			
RSX11		X					
ICL2900				X			
OLDUMP		Label specification is ignored					
Undef							X



Note: Each shaded area of this table shows the label that is substituted by PDL when an invalid host/label pair is specified. Each X indicates a valid label specification option for each type of host.

Host system JDLs on the system software tape

The job descriptor library source files supplied on the system software tape are summarized below.

Table H-2. **Job descriptor library source files**

JDL	JDEs provided for:	
BUR	Burroughs medium systems (B2500, B2700, B3500, B3700, and B4700)	
	Burroughs large systems <MDSH> labeled printer backup and standard ANSI-labeled tapes (B6700 and B7700)	
DUMP	Tape dumps of various tape formats where:	
	JDE	Characteristics
	1 -or- H6BCD 2 -or- EBCDIC 3 -or- ASCII 4 -or- H2BCD 5 -or- IBMBCD 6 -or- UNIVAC	H6000 BCD EBCDIC ASCII H2000 BCD IBM BCD UNIVAC Fieldata
H2000	Honeywell 2000 SPR-labeled tapes	
	Honeywell 2000 standard, COBOL and SPR BCD-formatted tapes	
H6000	Honeywell 6000 SSF BCD and ASCII-formatted tapes	
IBMRCA	IBM OS and DOS standard-labeled tapes	
	IBM ANSI-labeled and OS Writer tapes	
	US70 (RCA)-labeled tapes	
ICL	ICL 2900-labeled tapes	
NCR	NCR-labeled tapes	
OLDUMP	online dumps of host transmissions	
Online	online with/without banner and/or trailer pages	
POWGRP	IBM POWER VS, POWER 4.0	
	POWER 4.1/4.2; IBM DOS GRASP (including Interpress tapes)	
RSX	RSX-11 (FLX) ASCII tapes (including Interpress tapes)	
UNIVAC	UNIVAC SDF-formatted tapes	
XEROX	ANSI-labeled tapes	
	Unlabeled tapes	
	Xerox ANSI-labeled tapes	
	JDEs to print unknown tapes	
	Printing tapes without attempting to properly deblock the tape	

After an LPS software system is created by the user, the source for these JDLs are resident in the JSL directory. Use the editor PRINT command or the PDL compiler to obtain a printout of the JDL(s) desired.

The following appendix contains notes on online operation of the Xerox DocuPrint 180 Laser Printing System.

Processing 3211/4245 commands

While certain minor differences exist between the 3211/4245 and the LPS online system's emulation of this device, the responses are appropriate and require no programming changes to IBM software. The emulated command codes are listed on page I-5.

The 3211 commands are processed by LPS online as follows:

- All write commands perform the included carriage functions after the line is printed.
- Carriage control commands are processed exactly as the 3211 does, including the overprint function which results in overprinted characters. When processing a skip command, if the channel to be skipped to is assigned to the current line, no skip occurs unless the preceding command was printed without spacing.
- The 150 character per line option is supported. User specifies appropriate fonts in the PDE.
- TEST I/O is processed exactly as the 3211 does.
- NO-OP is processed exactly as the 3211 does.
- LOAD UCSB is processed as follows:

The train-image field is used in responding to a CHECK READ command that is not preceded by a DIAGNOSTIC GATE command.

The associative field is used to build the translate table based upon the FOLD or UNFOLD command in effect. Default is an UNFOLDED translate table.

3211 Dualing feature is not supported. The bits, if set, are ignored. Characters to be printed by the LPS are defined within the fonts.

- FOLD and UNFOLD are processed as the 3211 does.
- BLOCK (Disallow) DATA CHECK is processed as the 3211 does.
- ALLOW DATA CHECK is processed as the 3211 does.
- LOAD FCB is processed as follows:

The print position indexing (PPI) byte is processed as the 3211 does.

Line spacing control bit at FCB address 1 is ignored. Both 6 and 8 LPI are supported as a function of the fonts specified in the PDE statement.

The current channel assignment (VFU) table is released, and a new one is built.

- The diagnostic commands for the 3211 are emulated as follows:

SENSE is processed as the 3211 does.

READ PLB is processed as the 3211 does.

READ UCSB is processed as the 3211 does.

READ FCB is processed as the 3211 does but does not result in forms misalignment.

CHECK READ is processed as the 3211 does except that if a parity error occurs, all bytes in the PLB are flagged as having bad parity. This has no effect on operational software since this command is only used by IBM diagnostics.

DIAGNOSTIC WRITE is processed as in the 3211.

RAISE COVER is accepted but functionally ignored.

DIAGNOSTIC GATE command is functionally equivalent to the 3211. A flag is set for use during the processing of other commands (in particular, CHECK READ). This flag is cleared at the completion of the command following the DIAGNOSTIC GATE command.

The following 4245 specific commands are accepted.

- CLEAR PRINTER (ignored).
- SENSE ID (operates as does the 4245).

Universal Character Set Buffer (UCSB)

The Universal Character Set Buffer is used by the 3211 to define the order of characters on the print train and the complete set of printable characters available on the train. The buffer is 512 bytes long but the LPS uses primarily the associative field of 64 bytes in buffer locations 448 through 511. In this field, if a designated bit associated with a hexadecimal character value is set to one, the character is interpreted as printable. If the bit is zero, that character is interpreted as not printable.

The LPS substitutes a blank for any hexadecimal character value whose associative bit is set to 1 but for which there is no printable character in the font. It is recommended that an installation set the entire associative-bit field to binary 1's, defining all hexadecimal values as printable.

The following table defines the UCSB byte locations by function.

Table I-1. **UCSB byte locations**

Location	Function
1-432	Train image field -- used in responding to a CHECK READ command that is not preceded by a DIAGNOSTIC GATE command.
433-447	Reserved field -- ignored by the LPS.
448-511	Associative field -- bits 0-3 of each location are used to define printable characters.
512	Reserved field -- ignored by the LPS.

The correspondences between location in the associative field hexadecimal values, characters, and the associative bits that determine the character's printability are listed on page I-4.

Forms Control Buffer (FCB)

The Forms Control Buffer (FCB) defines channel positions and forms length. It is analogous to a carriage tape on an impact printer. Each byte in the buffer corresponds to one line on the form.

Table 1-1 describes the hexadecimal codes.

Table I-2. **Hexadecimal code definitions**

Hexadecimal code	Definition
01 to 0C	Channels 1 to 12
00	No channel assignment
1n	Last line, where n may be zero or a channel code

Indexing (shifting of print position one) may be initiated when the FCB is loaded. This is defined by a one-byte code, the print position indexing (PPI) byte, preceding the forms definition data. The PPI byte should be omitted from the FCB load data if no indexing is required. The maximum shift is 30 print positions.

Table 1-2 defines the binary codes.

Table I-3. **Binary code definitions**

Binary code	Definition
100N nnnn	Shift right n-1 positions
010N nnnn	Shift left n-1 positions (the first n-1 bytes of each record is not printed by the LPS).

Online input processing modifies the size and address of the print line based on the PPI byte, if present, before processing the DATA command of the LINE statement. If the resultant size of the print line is zero or negative (after PPI modification), only PCC action will take place.

Character correspondence in UCSB associative field

Figure I-1 shows character correspondence in UCSB associative field.

Figure I-1. Character correspondence in UCSB associative field

UCSB Location								
Byte	Bit 0		Bit 1		Bit 2		Bit 3	
	Hex	Char	Hex	Char	Hex	Char	Hex	Char
448	00		40	Sp	80	%	C0	{
449	01		41		81	a	C1	A
450	02		42		82	b	C2	B
451	03		43		83	c	C3	C
452	04		44		84	d	C4	D
453	05		45		85	e	C5	E
454	06		46		86	f	C6	F
455	07		47		87	g	C7	G
456	08		48		88	h	C8	H
457	09		49		89	i	C9	I
458	0A		4A	€	8A		CA	
459	0B		4B	.	8B	{	CB	
460	0C		4C	<	8C		CC	
461	0D		4D	(8D		CD	
462	0E		4E	+	8E		CE	
463	0F		4F		8F		CF	
464	10		50	&	90	½	D0	}
465	11		51		91	j	D1	J
466	12		52		92	k	D2	K
467	13		53		93	l	D3	L
468	14		54		94	m	D4	M
469	15		55		95	n	D5	N
470	16		56		96	o	D6	O
471	17		57		97	p	D7	P
472	18		58		98	q	D8	Q
473	19		59		99	r	D9	R
474	1A		5A	!	9A		DA	
475	1B		5B	\$	9B	}	DB	
476	1C		5C	*	9C		DC	
477	1D		5D)	9D		DD	
478	1E		5E	;	9E		DE	
479	1F		5F	~	9F		DF	

UCSB Location								
Byte	Bit 0		Bit 1		Bit 2		Bit 3	
	Hex	Char	Hex	Char	Hex	Char	Hex	Char
480	20		60	-	A0		E0	\
481	21		61	/	A1		E1	
482	22		62		A2	s	E2	S
483	23		63		A3	t	E3	T
484	24		64		A4	u	E4	U
485	25		65		A5	v	E5	V
486	26		66		A6	w	E6	W
487	27		67		A7	x	E7	X
488	28		68		A8	y	E8	Y
489	29		69		A9	z	E9	Z
490	2A		6A	^	AA		EA	
491	2B		6B	,	AB		EB	
492	2C		6C	%	AC		EC	
493	2D		6D	_	AD	[ED	
494	2E		6E	>	AE		EE	
495	2F		6F	?	AF		EF	
496	30		70		B0		F0	0
497	31		71		B1		F1	1
498	32		72		B2		F2	2
499	33		73		B3		F3	3
500	34		74		B4		F4	4
501	35		75		B5		F5	5
502	36		76		B6		F6	6
503	37		77		B7		F7	7
504	38		78		B8		F8	8
505	39		79	€	B9		F9	9
506	3A		7A	:	BA		FA	
507	3B		7B	#	BB		FB	
508	3C		7C	@	BC		FC	
509	3D		7D	,	BD]	FD	
510	3E		7E	=	BE		FE	
511	3F		7F	"	BF		FF	



Note: Characters in the "Char" column are shown for illustration. Actual characters printed are a function of the font specified in the PDE statement.

Emulated command codes

Figure I-2 shows emulated command codes.

Figure I-2. Emulated command codes

Hex	Function	Hex	Function	Hex	Function
01	Write without spacing	0B	Space 1 line immediate	00	Test I/O
09	Write and space 1 line	13	Space 2 lines immediate	02	Read Print Line Buffer (PLB)
11	Write and space 2 lines	1B	Space 3 lines immediate	03	No-op
19	Write and space 3 lines	83	Skip immediate to channel 0	04	Sense
89	Write and skip to channel 1	8B	Skip immediate to channel 1	05	Diagnostic Write
91	Write and skip to channel 2	93	Skip immediate to channel 2	06	Check Read
99	Write and skip to channel 3	9B	Skip immediate to channel 3	07	Diagnostic Gate
A1	Write and skip to channel 4	A3	Skip immediate to channel 4	0A	Read UCSB
A9	Write and skip to channel 5	AB	Skip immediate to channel 5	12	Read FCB
B1	Write and skip to channel 6	B3	Skip immediate to channel 6	23	Unfold
B9	Write and skip to channel 7	BB	Skip immediate to channel 7	43	Fold
C1	Write and skip to channel 8	C3	Skip immediate to channel 8	83	Load FCB
C9	Write and skip to channel 9	CB	Skip immediate to channel 9	6B	Raise cover (functionally ignored)
D1	Write and skip to channel 10	D3	Skip immediate to channel 10	73	Block data check
D9	Write and skip to channel 11	DB	Skip immediate to channel 11	7B	Allow data check
E1	Write and skip to channel 12	E3	Skip immediate to channel 12	FB	Load UCSB



Note: Also included:

SENSE ID E4_H

CLEAR PRINTER 87_H

Online dump example

Figure I-3 is an example of an online dump.

Figure I-3. Online dump example

HOST COMMAND: 04	SENSE	END STATUS: 0C	SEQ.#: 1	LENGTH: 6	28-FEB-98	03:12:52.655
0000	00000000	0000		*		*
HOST COMMAND: 01	WRITE WITHOUT SPACING	END STATUS: 04	SEQ.#: 2	LENGTH: 1	28-FEB-98	03:12:52.671
0000	00			*		*
HOST COMMAND: 07	DIAGNOSTIC GATE	END STATUS: 04	SEQ.#: 3	LENGTH: 0	28-FEB-98	03:12:52.873
HOST COMMAND: 06	CHECK READ	END STATUS: 0C	SEQ.#: 4	LENGTH: 1	28-FEB-98	03:12:52.886
0000	AD			*		*
HOST COMMAND: 12	READ FCB	END STATUS: 0C	SEQ.#: 5	LENGTH: 72	28-FEB-98	03:12:52.906
0000	00000000	01000000	00000000	00000000	00000000	00000000
0020	00000000	00000000	00000000	00000000	00000009	00000000
0040	00000000	00000010				
HOST COMMAND: 63	LOAD FCB	END STATUS: 0C	SEQ.#: 6	LENGTH: 71	28-FEB-98	03:12:52.936
0000	00000000	00000000	00000000	00000000	00000000	00000000
0020	00000000	00000000	00000000	00000000	00000000	00000000
0040	00000000	00000110				
HOST COMMAND: 8B	SKIP IMMEDIATE TO CHANNEL 1	END STATUS: 04	SEQ.#: 7	LENGTH: 0	28-FEB-98	03:12:53.006
HOST COMMAND: 01	WRITE WITHOUT SPACING	END STATUS: 04	SEQ.#: 8	LENGTH: 100	28-FEB-98	03:12:53.016
0000	616140D1	D6C240D7	D3C1C9D5	40404040	40404040	40404040
0020	40404040	40404040	40404040	40404040	40404040	40404040
0004	40404040	40404040	C4C1E3C5	40F0F461	F1F361F8	F36BC3D3
0060	F561F2F5				D6C3D240	F1F561F3
						DATE 02/29/98
						CLOCK 15/3-
						*5/25
HOST COMMAND: 8B	SKIP IMMEDIATE TO CHANNEL 1	END STATUS: 04	SEQ.#: 9	LENGTH: 0	28-FEB-98	03:12:53.052
HOST COMMAND: 09	WRITE AND SPACE 1 LINE	END STATUS: 04	SEQ.#: 10	LENGTH: 84	28-FEB-98	03:12:53.262
0000	40079989	95A38584	40B2A840	A3888540	F7859996	A740F9F7
0020	958540C3	969497A4	A3859940	D7998995	A3899587	4032A8A2
0040	4007C1C7	C540F0F0	F1404040	40D3C9D5	A3859440	40404040
						* PAGE 001
						LINE 01
						* Printed by the Xerox On-Lin
						DocuPrint 96
						*ne Computer Printing System

NOTE: REFER TO THE SECTION "ON-LINE DUMP" IN CHAPTER 9 FOR FURTHER DETAILS.

Glossary

A3	Paper size measuring 297 by 420 mm.
A4	Paper size measuring 210 by 297 mm.
abort	To terminate the printing of a job or execution of a program before it completes.
algorithm	Computational procedure that can be repeated any number of times.
alignment	Positioning of an image on a page for printing.
alphanumeric	Set of characters including the letters A through Z, numerals 0 through 9, and all printable special symbols.
AIM	Ancillary IOT message processor. System task that initializes the client layer between the printer and the system controller. It also displays the Fault, Hint, and information messages.
ASCII	American Standard Code for Information Interchange. Standard 7-bit code that represents alphanumeric information. Each alphanumeric character and several nonprinting characters are assigned a binary number, covering 128 possible characters. It is used for information interchange among data processing systems, data communication systems, and associated equipment.
application	Use to which a computer program or system is put, for example, sorting employee records.
applications software	Host- or LPS-resident software that directs the computer to perform specific tasks or functions as opposed to the software used to operate the computer. Common business applications include payroll, accounting, and inventory.
ascender	Portion of alphabetic character that rises above the body of the character (its x-height portion). See also descender; x height.
asynchronous	Transmission in data communications controlled by start and stop elements at the beginning and end of each character. Thus, time intervals between transmitted characters may be unequal in length.
audit log	Captures the sheet delivery information for every page in an audited report, certain details about each sheet, the planned and actual report control totals, and waste management.

auxiliary menu	Menu that contains options not displayed in a window. The symbol for an auxiliary menu is a box containing three horizontal lines.
B4	Paper size measuring 250 by 353 mm.
background job	Low-priority job, usually batched, that is executed automatically as system resources become available.
backup file	File copied to a storage medium for safekeeping in case the original is damaged or lost.
band	Rectangular area in printer memory into which an image sent to the printer from a computer is divided.
batch processing	Allows for repetitive operations to be performed sequentially on batched data without much involvement of the computer operator.
baud	Measurement of data rate in bits per second. This term is used to describe information flow between two devices. Unit of data transmitting and receiving speed is roughly equal to a single bit per second. Common baud rates are 110, 300, 1200, 2400, 4800, and 9600.
binary	Numbering system based on 2 that uses only the symbols 0 and 1. Binary is used in computers and related devices since information can be represented with electric pulses (0=off, 1=on). Most computer calculations are binary.
binary digit (bit)	<p>In the binary numbering system, either of the characters 0 or 1. The "bit" is the base unit of information used by computers. It can take the form of a magnetized spot, an electric pulse, or a positive or negative charge. A sequentially stored set of bits represents a character on a computer.</p> <p>Multipliers are:</p> <p>1 or 0 byte = 8,192 bits</p> <p>kilobyte (KB) or 1,024 bytes = 8,388,608 bits.</p> <p>Computer space equivalents are:</p> <p>1.5 KB = about 1 single-spaced typed page</p> <p>30 KB = about 20 typed pages</p> <p>150 KB = about 100 typed pages</p>
binary synchronous transmission	Data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations.
bit	Abbreviation for binary digit, the smallest unit of information recognized by a computer. See also binary digit.

bit map	Visual representation of graphic images in which a bit defines a picture element (pixel); for example, if a bit is 1, the corresponding pixel is printed.
bit mapped	Display image generated bit by bit for each point or dot. A software-driven scanner is used to create characters or graphics.
blocking	Process of combining two or more records into a single block of data which can then be moved, operated upon, or stored, as a single unit by the computer.
block length	Number of characters or bytes contained in a block of data (the block is treated as a unit within the computer). Block length is usually invariable within a system and may be specified in units such as records, words, computer words, or characters.
boot	To load the initial instructions of a program into computer memory; these instructions then direct the loading of the rest of the program. Booting may require entry of a few commands at the keyboard or the flip of a switch to begin the process.
bps	bits per second. In serial communication, the instantaneous bit speed with which a device or channel transmits a character.
BSC	binary synchronous communication. 1. Communication using binary synchronous line discipline. 2. Uniform procedure using a standardized set of control characters and control character sequences for synchronous transmission of binary-coded data between stations.
buffer	Compartment of memory in which this data is stored during transfer from one device to another. This is useful for accumulating data into blocks before storage or processing and for adjusting differences of speed between devices, or between a device and a communicating facility.
Bypass Transport	Optional module that moves paper from the last stacker bin to a finishing device.
byte	Fixed number of bits (in data processing, usually 8) operated upon as a unit. A byte may represent a character, a machine instruction, or some other logical unit of information.
carriage return	Control character that causes the printing system to start printing at the left margin of the current line unless set to be interpreted as a line end.
channel	1. In data communications, a path or line that enables two or more devices to communicate (sometimes called a circuit, facility, or link). 2. In computers, a path for communication between the central processing unit (CPU) and input/output units, or between the CPU and peripheral devices.

character	Single printable letter (A-Z), numeral (0-9), symbol (& % #), or punctuation mark (, . ! ?) used to represent data. Characters can also be nonprinting, such as space, tab, or carriage return.
character cell	Digitized space containing a single character within a font set.
character code	Code representing alphanumeric information, for example, ASCII.
character code identifier	Code associated with the universal identifier "Xerox" to indicate the version of the Xerox character code standard used to code Interpress strings.
character set	Number of different characters used by a particular device, including alphabetic, numeric, and special characters such as symbols.
client layer	The software interface used by the AIM task and the Output task to communicate with the printer, allowing printing commands and fault and status information to be exchanged.
clocking	A method of synchronizing the sending and receiving of data communications devices. Clocking allows synchronous transmission at high speeds.
cluster	Group of paper feeder trays, usually containing the same size and type of paper (stock). Each cluster has a name, consisting of one to six alphanumeric characters. See also stock; stockset.
CME	copy modification entry. Entry modifying the output printing characteristics of a report on a copy-to-copy basis.
code	1. Set of symbols representing data or instructions to a computer. 2. To write a list of instructions (software) to cause the product/system to perform specified operations.
code conversion	Translation of one type of character or symbol code to another.
collate	To arrange or assemble into ordered sets.
column	Vertical arrangement of characters.
command	User instruction to a computer, using the system controller keyboard or the PC UI. Commands are words, mnemonics, or characters that cause a computer to perform predefined operations. Coded instruction to a computer or computer-based system.
command language	Set of commands that can be used for a system, such as how the system can be instructed to perform a task.
comment	Explanations written with program instructions. They are ignored by the computer.

communication line	Telecommunication line connecting devices at one location with devices at other locations in order to transmit and receive information.
communication link	Physical means, such as data link, connecting one location to another to transmit and receive information.
communications	Ability of two devices to transmit information to each other.
compatibility	Characteristic of computer equipment permitting one device to use the same information or programs as another device without conversion or code modification.
compiler	Software that translates instructions written in high-level language into machine language for execution by a system.
computer	Functional unit capable of performing substantial computations, including numerous arithmetic or logic operations without human intervention during a run.
computer language	Computer-oriented language consisting solely of computer instructions. See also machine language.
computer system	Central processing unit (CPU) with main storage, input/output channels and devices, control units, and external storage devices connected to it.
concatenate	To connect or link in a series, as when files are grouped together for faster processing. See also job concatenation mode.
console	Functional unit containing devices used by an operator to communicate with an operating system. It may consist of a display, keyboard, and certain switches or other controls.
consumable supplies	Supplies such as paper and dry ink that are depleted (used up) during the course of normal printer operation.
continuous printing	Refers to Interpress job integrity under any of the following conditions: excessive graphics, forms, or font use problems.
control program	Program that supports the operating system by monitoring the flow of jobs, tasks, processing, and so on, within the system; for example, a data communication program.
coordinate	Point on the x and y axis that determines a grid position.
copy	To duplicate data in a new location or on an additional storage medium, for example, to copy files from disk to tape.

copy-sensitive	Term used to indicate jobs in which multiple copies of a report will contain different data, as with paychecks and banking statements.
cpi	characters per inch. Designates the number of characters per inch for a particular typeface. See also pitch.
CPU	central processing unit. Interprets and executes instructions, performs all operations and calculations, and controls input and output units and auxiliary attachments.
data	1. In general, facts, numbers, letters, symbols, and so on, which can be processed or produced by a computer. 2. In data processing, the source data or raw data entered for processing, as opposed to the results obtained by processing.
database	Information to meet specific processing and retrieval needs. Generally applies to integrated file of data, arranged for access by many subsystems.
data communications	Transmission and reception of encoded information over telecommunication lines.
data file	Collection of related data records organized in a specific manner so that each record is similarly structured, for example, a payroll file set up with one record for each employee, last name first, indicating the rate of pay and all deductions.
data processing	Operations carried out on data by means of algorithms and programs to produce information or a specific result. The rearrangement of data into a suitable form for further use.
data rate	In data communications, the rate at which a channel carries data, measured in bits per second (bps).
data storage	Preservation of data on various media, for example, tape, disks, magnetic bubble memory, and so on.
data transmission	Transmission of coded data over telephone or other telecommunication lines.
debug	To detect and correct errors in a program.
decompose	To break down into component parts, such as when Interpress breaks down a Font Interchange Standard (FIS) master to compile font information.
default	Value assigned to a field by the system. Default fields may be used for such items as document formats, menu selections, input fields, font selection, and paper or image size. The default value of a field may be changed.

descender	Portion of alphabetic character that extends below the baseline. See also ascender, x height.
desktop	Basic display screen of the PC UI.
device	Any piece of hardware other than the central processing unit (CPU).
digitize	To express or represent data in digital (binary) form so that it can be processed electronically.
disk drive	Device that can read or write magnetic media.
display	Viewing device (monitor) that visually communicates system warnings, status, and error messages and reflects operator interaction with the system on a display.
DJDE	Dynamic Job Descriptor Entry. Command within an input data stream used to modify the printing environment dynamically.
DMR	Command parameter that designates a Digital Equipment Corporation host system.
document	1. Data medium and the data recorded on it, usually permanent, which can be read by you or a computer. 2. Collection of information pertaining to a specific subject or related subjects.
dot	Unit of measurement representing a fraction of an inch, for example, 300 dots per inch (dpi). It may also be referred to as a picture element (pixel) or spot.
download	To copy files using communication lines from the host onto LPS system disks.
dry ink	Minute dry particles of resin and carbon black used to create images. Dry ink can accept an electrical charge.
duplex printing	Printing on both sides (front and back) of a page. See also simplex printing.
EBCDIC	Extended Binary Coded Decimal Interchange Code. Coded character set consisting of 8-bit coded characters. It can accommodate 256 characters.
edgemarking	Use of graphic objects, usually lines or boxes, that bleed off the edge of the physical page. See also physical page.
electronic publishing	Integrated production of documents on demand, using digitally stored documents, computerized composition, and electronic printing systems.

elite	Smallest size standard typewriter type: 12 characters per horizontal inch. See also pica.
embedded blanks	Blank spaces within a command line.
Ethernet	Xerox local area network (LAN) that allows data to be transmitted by cable from one device to another, enabling it to share the network.
extended metrics	Measurements used in Interpress to alter the size of fonts, allowing more precision with character escapement. Used for rendered characters.
FCB	forms control buffer. Buffer for controlling the vertical format of printed output.
FDL	forms description language. LPS-resident source language used for designing electronic forms. See also FSL; form.
field	1. Part of a record that serves a similar function in all records of that group, such as name and address field. 2. Area or setting of practical activity or application.
file	Set of records or text that can be stored and retrieved. An organized, named collection of records treated as a unit. For offline, it is the data between the two tape marks. For online, it is the data between banner pages.
file protection	To prevent the contents on a disk or tape from being erased or written over by disabling the write head of a unit.
firmware	Permanent programs stored in read-only memory (ROM).
FIS	Font Interchange Standard. Standard that defines the digital representation of fonts and character metrics for the generation of an entire series of Interpress fonts.
fixed font	Font containing characters with fixed spacing. See also proportional font.
fixed pitch	Font set in which every character cell has the same width. In reference to character sets, this term describes typefaces in which all character cells are of equal width. Monospaced as opposed to proportional spaced.
fixed spacing	Arrangement of characters on a line so that all characters occupy the same amount of horizontal space.
floating accent	Nonspacing accent characters that can be combined with characters and printed as a composite.

font	Set of images, usually characters and symbols, having common characteristics such as style, width, height, and weight.
form	1. Compiled forms source library (.FSL) file. 2. Printed or typed document with blank spaces for inserting information. Specific arrangement of lines, text, and graphics stored in a computer under an identifying name. Page of data that, when preceded by proper commands, is stored on the system disk as a permanent file. It may be merged with variable data by a form start command. See also FDL; FSL.
format	1. Layout of a document, including margins, page length, line spacing, typeface, and so on. 2. In data storage, the way the surface of a disk is organized to store data. 3. To prepare the surface of a disk for acceptance of data.
form feed	Keyboard or printer control character that causes the printer to skip the top of the next page.
FSL	forms source library. Uncompiled collection of user-created files containing FDL commands. Refer to FDL; form.
function keys	Keyboard keys that produce no character but initiate a particular machine function, such as delete.
fuse	To affix dry ink to paper by heat or pressure or a combination of both.
GCR	group code recording mode. Refers to the specific density of data (such as 6250 bpi) as it is recorded on tape, which is measured in bits per inch (bpi).
graphics	Use of lines and figures to display data, as opposed to using text.
grid	Imaginary pattern of evenly spaced horizontal and vertical lines on a page.
grid unit	Smallest rectangle enclosed by horizontal and vertical lines on a grid. The size of a grid unit is expressed as the length of one side of a rectangle.
halftone screen	A tool used in offset printing, typesetting, and laser printing to convert a continuous tone (such as photographic) image to dots, which allows the image to be rendered accurately in these printing processes.
hardcopy	Machine output in permanent form, such as printed reports, listings, and so on. Output in a permanent form (usually on paper or paper tape) rather than in temporary form, as on a display. Contains readable printed copy of machine (for example, computer) output.
hard values	Nonoptimal adjustment of particular FIS fonts in terms of point size and orientation.

hardware	Physical components, such as mechanical, magnetic, and electronic elements of a system, as opposed to programs, procedures, rules, and associated documentation. Hardware is operated by software and firmware.
HCF	high-capacity feeder. Feeder tray capable of holding 2500 sheets of 20 pound/75 gsm paper. The high-capacity feeder trays are the primary paper supply for the DP180 LPS. They are located in the bottom half of the feeder/stacker modules.
HCS	high-capacity stacker. Stacker bin capable of holding 2500 sheets of 20 pound/75 gsm paper. In the LPS, the high-capacity stacker bins are located in the top half of the feeder/stacker modules.
hexadecimal	Numbering system with a base of 16. In this system, 10 through 15 are represented by A through F, respectively.
hierarchy	Relative priority assigned to arithmetic or logical operations that must be performed.
high-level language	Language consisting of words and symbols that are close to normal English and, therefore, readily understandable by the user. High-level languages are oriented to problems or commercial procedures and are the source languages for most programs.
host	Computer accessed by users which serves as a source of high-speed data processing for workstations with less computer power. See also mainframe.
host interface	Connection between network and host computer.
icon	Symbol appearing on the PC UI or printer control console that can be opened to display a window or screen options.
id	identifier. Character used to identify or name data and possibly to indicate certain properties of that data.
image area	Area on a physical page that may contain text or graphics.
information processing	Generic term encompassing both word and data processing, used to describe the entire scope of operations performed by a computer.
initialize	1. To prepare the magnetic surface of a blank diskette so that it can accept data. 2. To set all information in a computer system to its starting values (usually the first step is accomplished when a program is booted).
input	Data or text introduced into a computer-based system.
input/output	General term encompassing the flow of data into and out of a system.

input processing	Formatting control for the pages of a report.
insert	To add text or graphics to a document.
interface	Device by which two systems connect and communicate with each other.
interpolation	Series of logical rules implemented in the printer to convert a 300 spi input video stream to a 600 spi output video stream. Interpolation is functionally analogous to bit doubling (2x scaling), except the logical rules result in superior output.
Interpress	Xerox standard that defines digital representation of lines for printing. Interpress documents can be printed on any sufficiently powerful printer equipped with Interpress print software.
Interpress font utility (IFU) program	Program used to convert FIS fonts to LPS fonts.
Interpress master	File written according to the Interpress standard.
IPL	initial program load. For the optional open-reel tape drive, the internal initialization sequence whereby certain functions are loaded into random access memory (RAM).
JDE	job descriptor entry. Collection of job descriptions. See also job; JSL.
JDL	job descriptor library. Collection of compiled job descriptions. See also JSL.
job	1. Set of instructions (JDEs) defining a unit of work for the system. 2. In setting a separation boundary through the Bin Full Criteria task, job refers to everything printed as the result of a single start command. See also JDE.
job concatenation mode	In HIP, a mode in which multiple print jobs are processed as reports in one print job. See also concatenate.
job control	Program called into storage to prepare each job or job step to be run.
job management	Collective functions of job scheduling and command processing.
JSL	job source library. Collection of uncompiled job descriptions. See also job; JDE; and JDL.
keyboard	Group of alphabetic, numeric, and/or function keys used to enter information into a system.
keyword	Required part of a command. See also operator command.

label	1. In data storage, a reference to a file saved on tape or disk, a record indicating the file name or date created, or other control information. 2. In programming, a name assigned to a particular instruction or portion of a program as a locational reference (the computer translates the label into an address).
landscape page orientation	Orientation of print lines or top of an illustration parallel to the long edge of the paper if the sheet is within the standard size range. (Sheets larger than standard have the reverse print orientation.)
language	Defined set of characters and symbols combined together by specific rules. See also high-level language; low-level language.
laser printing	Technology that uses a laser to transfer character forms to a page by direct or indirect means.
latent image	Static charge present on the photoconductor before contact with dry ink particles.
leading	1. Vertical distance between lines (also called line space), measured from a baseline of one line to the baseline of the next. 2. Extra spacing between lines of type. 3. In typography, spacing between lines and paragraphs.
LED	light emitting diode. Solid substance that glows when a current is passed through it. Often used for indicator lights on disk drives or modems, as well as for displays on other electronic equipment.
LEF	long-edge feed. The movement of paper through the printer in the direction of the paper length (the longer side of a sheet of paper).
legal size	Sheet the standard size of legal briefs, 8.5 by 14 inches.
letter size	Paper sized 8.5 by 11 inches/216 by 279 mm.
library	In data storage, a collection of related files or programs.
line	One horizontal flow of characters.
line feed	Control character that, unless set to be interpreted as a line end, causes the printing system to begin printing in the current character position of the next line.
line tables	Internal data structures providing a record in memory of lines to be drawn on a page.
listing	Printout or display of the statements in a program, usually used as a convenience in examining or editing programs.

literal	Alphanumeric beginning with a letter, optionally including an asterisk, period, colon, or slash, and not enclosed in single quotes.
load	To enter data into storage or working registers.
location	Place in which data can be stored.
log	Collection of messages or message segments placed on an auxiliary storage device for accounting or data collection purposes.
logical page	In the Xerox printing systems environment, a formatted page that is smaller than the physical page. A logical page is defined by an origin, thus allowing more than one logical page to be placed on a physical page.
logo	Small illustration or design, usually simple, typically used to identify a company.
log off	Procedure by which a user ends a session.
log on	Procedure by which a user begins a session between an application program and a logical unit.
magnetic media	Term for all storage devices, such as disks, diskettes, and tape, on which data is stored in the form of magnetized spots on surface of the media.
magnetic storage	Use of magnetic media to store data, programs, and so on.
magnetic tape	Flexible plastic tape, with one side offering a magnetic surface suitable for storing computer data in the form of magnetized spots. Magnetic tape is often used for long-term storage since it can accommodate large volumes of information.
mainframe	Central processing unit (CPU) and memory of a large computer. More often used to denote any large computer of the type that might be used to control a group of smaller computers, terminals, or other devices. See also host.
margins	White space on each side of printed text.
mask	1. Selection of bits from a storage unit by using an instruction that eliminates the other bits in the unit. 2. In accessing files, a file name mask is used to reference one or more files with similar file-id (identifier) syntax. 3. In Interpress, a mask serves as a template, indicating the shape and position of an object on a page.
MB	megabyte. Unit of one million bytes.

media	Vehicles or devices by which information is stored or transmitted. Classifications include source, input, and output.
medium	Object or material on which data is stored, for example, magnetic tape or floppy disk.
memory	Space in a device where information is kept, or the ability of a device to keep information until needed.
menu	List of available functions, commands, and options.
message	Unit of information transmitted by one facility to another in a form that the receiving facility can understand and act upon. The standard message format consists of a header (containing identifying and control information), followed by the actual message content, followed by a trailer (indicating that the message is completed).
metacode	Same as "native mode." The method of speaking to and controlling the image generator. These controls are used by the character dispatcher to generate scan line information. This information is sent in the form of character specifications to the image generator, which uses it to compose the bit stream that modulates the laser.
MHz	megahertz. 1. Unit of cycling speed (one million cycles per second) for an electromagnetic wave (in particular, a radio wave). 2. Sending and receiving stations of a radio wave transmission must be tuned in to the same unit of megahertz.
mode	Manner in which an activity or process is carried out.
modem	Device that converts digital information into an analog signal suitable for sending over analog telecommunication lines. Also converts an analog signal from telecommunication lines into digital information.
module	Cohesive unit within a program. It is consistent in its level and identifiable in terms of loading or with other units.
mouse	Electronic device used with the PC UI to select options and enter data.
mouse pad	Base that provides friction and direction information to the electronic mouse.
mouse tray	Optional extension module attached to the system controller to accommodate the mouse pad and provide ample working space.
network	1. System of geographically separate computers, linked to one another over transmission lines. 2. Communication lines connecting a computer to its remote terminals.

nonimpact printer	Printer that forms characters without any strikes of a key or element against the paper.
object file	Source file converted into machine language (binary code).
offline	Devices not under the active control of a central processing unit. For example, a computer makes output to a magnetic tape. The tape is then used by an offline printing system to produce printed data. Offline operations are much slower than online operations. See also online.
offset	To place pages currently being printed in slightly different positions from previous pages.
offset printing	Widely-used method of commercial and corporate printing, where ink is picked up by a metal or paper plate, passed to an offset drum, and then passed to the paper.
online	Devices under the direct control of a central processing unit, for example a printing system in interactive communication with a mainframe. See also offline.
operating system	Basic host- or LPS-resident controlling program that governs the operations of a computer, such as job entry, input/output, and data management. The operating system is always running when the computer is active. Unlike other types of programs, it does not run to an end point and stop. The operating system of a Xerox LPS is referred to as the operating system software (OSS).
operation	Well-defined action that, when applied to any permissible combination of known entities, produces a new entity.
operator area	The 24-inch exclusive clearance that must be available directly in front of each component of an LPS for operator activities.
operator command	Statement to control a program, issued through a console device, PC UI, or terminal, causing a control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.
orientation	1. In reference to image area, orientation describes whether the printed lines are parallel to the long edge of the paper or the short edge of the paper. 2. Choice of printing portrait (vertically) or landscape (horizontally).
origin	In reference to image area, this is the upper left corner of a sheet.
output	1. Material produced by a peripheral device of a computer, such as a printout or a magnetic tape. 2. Result of completed operations.
overprinting	Printing more than one character at the same position.

overprint lines	Print lines whose carriage control specifies printing with no line spacing after the last printed line.
overprint ratio	Maximum number of variable data and form characters that may be intersected by a single scan line.
override	To take precedence or priority over, to overrule.
overstrike	To print characters over each other.
page	1. In computer programming, a block of instruction, data, or both that can be located in main or auxiliary storage. 2. In word processing, a defined section of a document.
page orientation	Direction in which data is printed on a report. See also landscape page orientation; portrait page orientation.
parameter	Part of a command, other than the keyword. See also keyword; operator command.
pass-through job	On systems with XPAF, a job that is sent directly from a host to a Xerox printer using XPAF, without undergoing XPAF processing.
password	Unique word or set of characters that an operator or user must supply to log on to a system.
patch	In programming, to modify a portion of the program at the machine language level, as opposed to modifying at the source program level.
PC UI	PC user interface. The PC hardware and Xerox-supplied software which allows the operator to control the LPS by means of a mouse, windows, and icons. See also object mode; TEM.
PDL	print description language. Language used to describe printing jobs to an LPS. PDL describes the input (type, format, characteristics), performs the processing functions (logical processing), and describes the output (type, format, font selection, accounting options).
physical page	Sheet of paper on which printing is done. See also edgemarking.
pitch	1. Horizontal character spacing; 10-pitch (10 characters per inch) spacing is called pica, and 12-pitch (12 characters per inch) spacing is called elite. 2. The number of page images placed on the xerographic belt during one revolution. The DocuPrint 180 LPS supports two pitch modes: 7 pitch when paper 9 inches/229 mm long or less is used to print a job, and 3 pitch when paper 9 to 17 inches/229 to 432 mm long is used to print a job.
pixel	Acronym for picture element. Smallest addressable point of a bit-mapped screen that can be independently assigned color and

intensity. Pixels are definable locations on a display used to form images. For graphic displays, more pixels generally provide higher resolution. Spots, dots, and pixels are used interchangeably.

point	Unit of measurement equal to 0.0139 inch. Points are always used to express type size and leading. There are 12 points to a pica and about 72 points to every inch. See also pica.
point size	Height of character set from the top of its ascenders to the bottom of its descenders in units (points). Point size does not always include leading.
portrait page orientation	Orientation of print lines or the top of an illustration parallel to the short edge of the paper if the sheet is within the standard size range. Sheets larger than standard have the reverse print orientation.
printer	Output device that produces hardcopy printouts. Also referred to as the IOT.
print file	Position of the system disk memory (up to 4 MB) received for temporary storage of formatted pages for printing. Pages are retained until they are delivered to the output tray.
printout	Informal expression referring to almost anything printed by a computer peripheral device.
process	1. To perform a systematic sequence of operations, such as add, edit, delete. 2. To produce a specific result by manipulating data.
program	Complete set of instructions in language compatible with the device to be used. A program directs a system to perform each operation at the right time in the proper sequence.
programmer	Person involved in designing, writing, and testing computer programs.
prompt	Message or symbol displayed on a system console requiring the operator to take action.
proportional font	Font containing characters that vary in width. See also fixed font.
proportional spacing	Text in which each alphanumeric character is given a weighted amount of space. Such output has print-like appearance. Proportional spacing allows more space for wide characters and less space for narrow characters
proportional type	Characters that vary in width.
protocol	Formal set of conventions governing the format of data and the control of information exchange between two communication devices.

purge	To delete data from a system.
queue	List of documents waiting to be processed.
RAM	random access memory. Storage that allows data, such as documents, to be stored and retrieved directly by address location without reading through any other data.
raster data	Binary data, usually consisting of dots arranged in scan lines, according to the print order.
rasterization	Creation of a page's bit map image for printing.
read/write head	Input/output device that reads and writes data in the form of magnetic dots on the surface of an external storage medium, such as a magnetic disk.
record	Collection of data or words treated as a unit.
recovery	Act of overcoming a problem or error during processing. Typically, a specialized software recovery routine gains control and attempts to resolve the error without crashing the system.
remote access	Access to a central computer by terminals or devices geographically separated from that computer.
replace	Process of exchanging one portion of text for another. This process encompasses two functions: deleting old text and inserting new.
report	In setting a separation boundary through the Bin Full Criteria task, report refers to a subset of a job (a job may consist of one or more reports).
resolution	Number of dots per unit. The LPS imaging system converts a character from digitized data into a printed image composed of these tiny dots. The greater the number of dots per inch, that is, the resolution, the clearer the image that is produced.
ROM	read-only memory. Solid-state memory for programs. It cannot be rewritten.
save	To store programs, data, or text for retrieval at a later time.
scale	To adjust font or image size according to given proportions.
scroll	Manipulation of a display to bring upper or lower portions of a document page into view when no space is available for the entire document at once.

scroll bar	Part of a window that allows you to view information extending beyond the window display.
SCSI	small computer system interface. Accepted standard for connecting peripheral devices to computers.
secondary storage	Form of storage external to a system, such as magnetic tapes or floppy disks.
security	1. Procedure for limiting access to the system resources, programs, or files, to authorized personnel. 2. Protecting programs and files from unintentional or undesired modification.
SEF	short-edge feed. The movement of paper through the printer in the direction of the paper width (the shorter side of a sheet of paper). For the DocuPrint 180 LPS, short-edge feed allows larger sizes of paper (up to 11 by 17 inches/279 by 432 mm) to be printed.
sequential	In numeric sequence, usually in ascending order. As applied to a file organization, describes files in which records are written one after another and cannot be randomly accessed. For example, the first 99 records in a sequential file-access file have to be read before the 100th record is accessible.
set	In setting a separation boundary through the Bin Full Criteria task, set refers to multiple copies of the same report.
simplex printing	Printing on one side of the page. See also duplex printing.
software	Programs, including operating systems, procedures, utilities, and applications programs, written for a system. Software can be supplied by the hardware manufacturer or other firms but does not include programs written by the user.
sort	To rearrange data records according to a particular item (field) which they all contain, using a predetermined ordering scheme.
source	Terminal node at which data enters a network. For example, a computer transmitting data through telecommunication lines to several other computers or receiving terminals.
source file	File containing source language statements or commands.
source language	Language, high-level or low-level, used by a programmer. A source language must be converted by a compiler to machine language for the instructions to be executed.
source program	Program written in source language
space	Blank area between words, recognized as a character by word and data processing systems.

special processing	Commands allowing the user to process special reports, such as printing certain records, or printing on special paper.
spooling	Process of releasing data from main memory and storing it temporarily until a peripheral device is ready to accept it, for example storing text before sending it to a printer.
spot	Unit of measurement representing a fraction of an inch, for example, 300 spots per inch (spi). May also be referred to as a picture element (pixel) or dot.
statement	Detailed written instructions in a program step. A statement is written according to specific rules called syntax.
static data	Information usually found on preprinted forms or overlays.
stock	User-defined name in the JSL that specifies a certain type of paper for printing a job. See also cluster.
stockset	Collection of stocks to be used on a print job. See also stock; cluster.
storage	Retention of information. Primary storage refers to internal storage where data and program instructions pertinent to current operations/jobs are held. Auxiliary storage refers to external media, such as disks or tapes, for use at a later time.
string	Connected sequence of alphanumeric characters treated as one unit of data by a program.
symbol	Character used in a computer language to specify a particular function.
synchronous	Efficient encoding of data suitable for high-speed, block-oriented data transmission by using equal time elements.
syntax	Rules governing the structure of expressions in a programming language.
syntax error	System response to a mistake in a command entry.
system	1. In data processing, a collection of parts and procedures organized to accomplish a set of specific functions. 2. Assembly of components united by some form of regulated interaction to form an organized whole. 3. Operations or procedures through which a business activity is accomplished.
system controller	Part of the LPS that provides interfacing capability, data handling, formatting, buffering, and operator control for the system. Also referred to as the "ESS".

system disk	Magnetic disk storage medium, usually of large capacity, that is not removable as opposed to floppy disk or disk packs.
system file	Master software program that keeps all components working together.
system generation	Process whereby the system is made ready to operate. Typically involves selecting the operative parameters and activating the relevant software.
system page	Maximum area in which text and graphics can be imaged on a printing system.
system software	Software programs that support and/or control system functions by governing hardware operation and input/output processes, interpreting source programs and breaking them down into machine language, distributing tasks among various processors, and so on.
tab	To move the cursor on a display or printer to a prespecified column on the display or paper, most often by using the <TAB> key on a keyboard.
tape	Recording media for data or computer programs. Tape can be in permanent form, such as perforated paper tape. Generally, tape is used as a mass storage medium in magnetic form and has a far higher storage capacity than disk storage, but it takes longer to write or recover data from tape than from disk.
tape density	The number of characters that can be stored on magnetic media, such as how close together data can be recorded. The Xerox LPS may use either 1600 bpi or 6250 bpi density magnetic media.
tape drive	Input/output device that controls the movement of magnetic storage tape past the read/write head while data is accessed or stored.
task	1. Any major job performed by a computer. 2. One of several programs being executed by a system.
telecommunications	Voice or data communications transmitted and received through telephone lines.
teleprocessing	Sending and receiving data through telecommunication lines for processing among various remote terminals and the central processing unit (CPU).
TEM	terminal emulation mode. The processing method of the PC UI which features a character display and the operator's use of the keyboard. The mouse, windows, and icons are inactive in this mode. The PC UI operates in either the TEM mode or in the object mode. See also object mode.

terminal	Device equipped with a keyboard and connected to a computer or a network
testing	1. Process of running a program for the express purpose of discovering any errors it may contain. 2. For computer-oriented systems, the process of verifying a system's ability to meet performance objectives in a simulated environment or validating its actual performance in a live environment.
text	In communications, the content portion of a transmitted message.
text string	Consecutive series of characters to be printed exactly as specified in a command.
throughput	In data processing systems, the amount of data that can be processed, transmitted, printed, and so on, per a specified unit of time.
toggle	To switch (alternate) from one tray to another. The system switches from an active feeder or stacker tray to an inactive one to allow continuous printing when the proper commands are invoked.
trailer	In data communications, the last portion of a message that signals the end.
transaction processing	Method of data processing in which files are updated and results are generated immediately after data entry.
translation	1. In data communications, the conversion of one code to another on a character-by-character basis. 2. In programming, the function of a language processor which converts a source program from one programming language to another.
transmission speed	In data communications, the rate at which data is passed through communication lines, usually measured in bits per inch (bpi).
transmit	To send data from one place to another.
truncated	Cut off before completion, as when data transfer from a host to a printer is cut off before all data has been transmitted.
two-up	Application that prints two logical pages on one side of a physical page.
typeface	1. All type of a single design. 2. Set of characters with design features that make them similar to one another.
type size	Height of a typeface, measured from the bottom of its descenders to the top of its ascenders, expressed in points.

type style	Italic, condensed, bold, and other variations of typeface that form a type family.
UCS	Universal Character Set. Printer feature that permits the use of a variety of character
upload	To copy files from a remote peripheral device to a host. LPS files are not copied to the host because one of the LPS functions is to store files for the host.
utility program	General-purpose program that performs activities, such as initializing a disk or sorting, which are not specific to any application.
validation	Process of testing a system's ability to meet performance objectives by measuring and monitoring its performance in a live environment.
variable	Information of a changeable nature which is merged with a standard or a repetitive document to create specialized or personalized versions of that document.
variable data	Variable data is not a part of a form design. It varies from page to page.
variable text	Text of changing nature, such as various names and addresses combined with a form letter to make a complete document.
virtual page	Page area selected by a forms designer for printing.
warning message	System-issued message indicating that an error has been encountered even though the program has not failed.
weight	Characteristic of type determined by how light or dark it appears.
wildcard	Part of a command (* symbol, / symbol, ? symbol) that represents a category for which the possible options are requested.
wildcarding	In a command, calling out a general category rather than a specific item within that category. The purpose of wildcarding is to generate the options within the given category.
window	In the PC UI, the graphic display object that allows you to select options and enter information.
write	To record data in memory or an external storage medium.
write protection	Data protection feature implemented on magnetic media (for example floppy disk, 9-track tape) to prevent stored data from being modified, written over, or erased.

x axis	Horizontal axis on a forms grid.
xdot	Unit of measurement representing a fraction of an inch. It may also be referred to as a picture element (pixel) or spot, for example, 1/600 spots per inch (spi).
xerographic engine	Component of a printer that develops an image, transfers it to paper, and fuses it for output as hardcopy.
x height	Height of lowercase letters without their ascenders or descenders (height of letter "x"). See also ascender; descender.
y axis	Vertical axis on a forms grid
ACT	Advanced Customer Training
AFP	Advanced Function Printing
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
BCD	binary coded decimal
BOF	bottom of form
BOT	beginning of tape
bpi	bits per inch
bps	bits per second
BSC	binary synchronous communications
CD	character dispatcher
CDC	control data communications
CD/IG	character dispatcher/image generator
CM	control module
CME	copy modification entry
cpi	characters per inch

CPU	central processing unit
CR	carriage return
DCE	data communications equipment
DDCMP	Digital Data Communication Message Protocol
DEC	Digital Equipment Corporation
DFA	Document Feeding and Finishing Architecture
DJDE	dynamic job descriptor entry
DMR	data mode ready (DEC interface for LPS)
DOS	disk operating system
dpi	dots per inch
DSDD	double sided double density
DSU	digital signal unit
DSR	disk save and restore
DSSD	double sided single density
DTE	data terminal equipment
EBCDIC	Extended Binary Coded Decimal Interchange Code
ENET	Ethernet network
EOT	end of tape
EP	electronic publishing
ESS	electronic subsystem, also referred to as the system controller
FCB	forms control buffer
FCG	finishing configuration utility
FCP	file control parameter

FDL	forms description language
FDR	file directory
FFM	font file management
FIS	Font Interchange Standard
FMS	file management subsystem
FPS	formatting print service
FSL	forms source library
FST	font specification table
GCR	group code recording
gsm	grams per square meter
HCF	high-capacity feeder
HCS	high-capacity stacker
HFDL	host forms description language
HIP	Host Interface Processor
hpos	horizontal positioning
IBM	International Business Machines Corporation
IFU	Interpress font utility
IG	image generator
IGM	image generator module
I/O	input/output
IOM	image output module
IOT	input output terminal, also referred to as "printer"
IPD	Interpress decomposer

IPFONTS	Interpress fonts
IPL	initial program load
IPM	Interpress mapping
ips	inches per second
JCB	job control block
JCL	job control language
JDE	job descriptor entry
JDL	job descriptor library
JID	job identifier
JSL	job source library
LAN	local area network
laser	light amplification by stimulated emission of radiation
LED	light-emitting diode
LEF	long-edge feed
LF	long-edge feed
lpi	lines per inch
LPS	Laser Printing System
MTU	magnetic tape unit (refers to the 9-track magnetic tape drive; also referred to as "magnetic tape drive")
OCR	optical character recognition
OCS	operator communications subsystem
OLI	online interface
OS	operating system

OSDS	operating system diagnostic software
OSEXEC	operating system executive
OSS	operating system software
PC	personal computer
PCC	printer carriage control
PC UI	personal computer user interface
PDL	print description language
PE	phase encoded
ppm	pages per minute
PQA	print quality adjustment
PROM	programmable read-only memory
PSC	printer subsystem controller
pt	point
PWBA	printed wiring board assembly
QIC	1/4 inch cartridge
RAM	random access memory
ROS	raster output scanner
SAFES	stand-alone field engineering software
SAN	software analysis number
sci	START command index
SCSI	small computer system interface
SDLC	synchronous data link control
SEF	short-edge feed

SFS	status file services
SIF	sequence insert file
SNA	system network architecture
spi	spots per inch
SST	system software tape
sysgen	system generation
TL/DL	text line/display list
TOF	top of form
tpi	tracks per inch
TPJ	test pattern job
TXC	total xerographic convergence
UCS	Universal Character Set
UCSB	Universal Character Set Buffer
UI	user interface
VM	virtual memory
vpos	vertical positioning
VS	virtual storage
WAN	wide area network
XCSC	Xerox Customer Support Center
XDGI	Xerox DCF and GDDM Interface
XDSS	Xerox Documentation and Software Services
XICS	Xerox Integrated Composition System
XJCF	Xerox Job Control Facility

XMP	xerographic mode persistence
XMS	xerographic mode switching
XPAF, XPF	Xerox Printer Access Facility
XPMF-VMS	Xerox Print Management Facility - VMS Version
XPPI	Xerox Pen Plotter Interface
XPS	Xerox Publishing System