
Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Print Description Language Reference

**THE DOCUMENT COMPANY
XEROX**

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: 721P85640

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 96/DocuPrint 96MX Laser Printing System Print Description Language Reference* is part of the eight 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 96/DocuPrint 96MX Laser Printing System Operator Guide</i>	721P85590
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Operations Reference</i>	721P85610
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Message Guide</i>	721P85650
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System PDL Reference</i>	721P85640
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Forms Creation Guide</i>	721P85630
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System System Generation Guide</i>	721P85620
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Installation Planning Guide</i>	721P85600
<i>Xerox DocuPrint 96/DocuPrint 96MX Laser Printing Operator Command Summary Card</i>	721P85660
<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 96/DocuPrint 96MX Laser Printing System document set	xxii
About this manual	xxiv
PDL syntax conventions used in this manual	xxv
1. Overview	1-1
PDL features and functions	1-1
LPS component types	1-2
Hardware	1-2
Software	1-2
LPS hardware components	1-3
Advanced Image Subsystem (AIS)	1-3
LPS software components	1-4
PDL related programs and tasks	1-6
Operating system executive task (OSEXEC)	1-6
File control program (FCP)	1-6
Operator communication subsystem (OCS)	1-6
Editor task	1-6
Font editor task	1-6
Input processing task	1-6
Report task (RPT)	1-7
Dynamic job descriptor (DJD) task	1-7
Output processing task	1-7
Print description language (PDL) processor	1-7
Forms description language (FDL) processor	1-7
Preparing for a print job	1-8
Job flow process	1-9
OCS processing	1-9
Input processing	1-9
Output processing	1-10
Input data types	1-11
Unformatted data	1-11
Formatted data	1-12

2. Print Description Language (PDL)	2-1
Purpose of PDL	2-1
PDL command structure	2-2
Command line length	2-2
Command components	2-2
Identifier	2-3
Command keyword	2-3
Parameter	2-4
Comments	2-4
Syntax rules	2-5
Right-part constants	2-6
Value constants	2-6
String constants	2-6
Job source library (JSL) structure	2-10
Command levels	2-10
ID level	2-12
System or JDL level	2-12
Catalog level	2-12
Job or JDE level	2-13
Examples—JOB command	2-13
END command	2-13
Creating separate files for grouping PDL commands	2-16
Hierarchy of replacement	2-16
Hierarchy within a job descriptor library (JDL)	2-17
Non-JDL hierarchy — START command	2-18
START command	2-18
Hierarchy of replacement in an errored job descriptor library (JDL)	2-21
3. Creating a job source library (JSL)	3-1
What your JSL specified for LPS processing	3-2
Decisions to make before creating your JSL	3-3
Input data	3-3
Output specifications	3-4
Type of application to create	3-4
Special features	3-5
Interactions between JSLs, catalogs, and jobs	3-5
Review of PDL components and syntax	3-6
Command levels	3-6
Command components	3-6
Command identifiers	3-6

Command keywords	3-7
Command parameters	3-7
Comments	3-7
PDL syntax	3-8
Hints and tips	3-9
Steps in creating a JSL	3-11
Using the Editor	3-11
Name the JDL identifier	3-11
Specifying VFUs	3-11
Setting up input parameters	3-12
Specifying LINE command parameters	3-12
Specifying ACCT command parameters	3-12
Specifying use of DJDEs	3-13
Adding logical processing specifications	3-13
Specifying formats	3-13
Using copy modification entries	3-14
Defining paper requirements	3-14
Specifying output requirements	3-15
Ending a JSL	3-15
Finished JSL	3-16
Compiling the JSL	3-18
Printing the job	3-19
Page considerations	3-20
Paper sizes	3-20
System page	3-20
Physical page	3-20
Edgemarking	3-20
Non-imaged elements	3-20
Page orientation	3-21
Landscape orientation	3-22
Portrait orientation	3-23
Registration shift and skew	3-24
Fonts	3-26
Font and graphic memory	3-27
Xerox DocuPrint 96/DocuPrint 96MX compatibility with the 4850, 4135, 4635 and 4050/4090/4650 LPS	3-28
Xerox 4850 and 4890 HighLight Color LPS	3-29
Running 4850/4890 applications on your Xerox DocuPrint 96/DocuPrint 96MX LPS	3-29
Downloading 4850/4890 LPS applications to your Xerox DocuPrint 96/DocuPrint 96MX LPS	3-30

	4850/4890 HighLight Color LPS forms	3-30
	Points to note	3-31
	Xerox 4050/4090/4650/9700F LPS	3-31
	Creating Xerox DocuPrint 96/DocuPrint 96MX LPS applications on a 4050, 4090, 4650, or 9700F LPS	3-31
	Running 4050, 4090, 4650, and 9700F jobs on your Xerox DocuPrint 96/DocuPrint 96MX LPS	3-31
4. Specifying input parameters		4-1
	BLOCK command	4-2
	ADJUST	4-2
	CONSTANT	4-3
	FORMAT	4-3
	LENGTH	4-4
	LMULT	4-4
	LTHFLD	4-5
	OFFSET	4-5
	POSTAMBLE	4-6
	PREAMBLE	4-6
	ZERO	4-7
	Points to note	4-7
	Online versus offline JDLs	4-8
	Example	4-8
	CODE command	4-9
	ASSIGN	4-9
	DEFAULT	4-10
	Point to note	4-10
	Examples	4-11
	PCC command	4-12
	ADVTAPE	4-12
	ASSIGN	4-13
	DEFAULT	4-16
	INITIAL	4-17
	MASK	4-17
	Points to note	4-18
	Example	4-18
	RECORD command	4-19
	ADJUST	4-19
	CONSTANT	4-20
	FORMAT	4-20

LENGTH	4-21
LMULT	4-21
LTHFLD	4-22
OFFSET	4-22
POSTAMBLE	4-23
PREAMBLE	4-23
STRUCTURE	4-24
Points to note	4-24
Example	4-25
SEFFNT command	4-27
Mapping files	4-28
SEFMAP	4-29
MAP	4-29
Points to note	4-30
Examples	4-31
TCODE command	4-32
DEFAULT	4-33
TASSIGN	4-34
TRESET	4-35
Points to note	4-36
Examples	4-37
VOLUME command	4-38
BMULT	4-39
CODE	4-40
EOV	4-41
HOST	4-42
INTERPRESS	4-45
LABEL	4-46
LCODE	4-47
LPACK	4-48
MAXLAB	4-48
MINLAB	4-49
OPTIMIZE	4-50
OSCHN	4-51
OSHDP	4-51
OSTLP	4-52
PLABEL	4-52
RMULT	4-53
RSAT	4-53
TCODE	4-54

	UNPACK	4-55
	Points to note	4-56
	Examples	4-57
Input sources		4-58
Online printing systems		4-58
	Channel-attached LPS	4-58
	Online 3211/4245 mode	4-58
	Online-specific commands	4-59
	Creating a JDE or JDL	4-59
	DJDE processing	4-61
	Online optimization	4-61
	Copy-sensitive copy modification entries (CME)	4-61
	Report separation	4-61
	Universal character set buffers (UCSBs)	4-62
	UCSB processing	4-62
	Forms control buffer (FCB)	4-63
	Vertical format control processing	4-63
	Online record length	4-64
	Points to note	4-64
	Online recovery	4-65
	Online dump	4-66
	Starting and ending dump sessions	4-66
	Dump format	4-66
	Points to note	4-67
	Downloading files from the host to the LPS	4-68
	Valid download file types	4-68
	DJDE FILE command	4-69
Offline mode		4-70
	Host computer tape formats	4-70
	Tape codes	4-70
	Packed data formats	4-71
	Record formats	4-71
	Record structure	4-71
	Multivolume processing	4-72
5. Defining clusters		5-1
	Cluster features	5-1
	Cluster processing overview	5-1
	What clusters do for the programmer and operator	5-2
	Where clusters are stored	5-3
	How applications use clusters	5-4

Simple and OTEXT applications	5-4
Stockset applications	5-5
Mixing applications	5-5
Defining clusters and stocksets with PDL and DJDE	5-6
PDL commands	5-6
DJDEs	5-6
Points to note	5-7
Steps for creating clusters	5-8
Keeping stockset changes to a minimum	5-10
Using clusters with ordered stocks	5-10
6. Print format commands	6-1
ABNORMAL command	6-2
ERROR	6-3
IMISMATCH	6-4
ISUBSTITUTE	6-6
OTEXT	6-7
PAGES	6-7
REP	6-8
SECURITY	6-8
Recovery	6-9
Marker page	6-9
ACCT command	6-11
Accounting data on report basis	6-11
Overall usage accounting	6-12
Installation accounting report	6-13
DEPT	6-16
USER	6-16
Points to note	6-17
Example	6-17
ac:CME command	6-18
Short form CME specifications	6-18
Cataloged CMEs	6-19
CONSTANT	6-19
FONTS	6-20
LINE	6-21
POSITION	6-22
Points to note	6-22
Examples	6-24
Example 1	6-24
Example 2	6-24

EXPORT command	6-25
SEPARATORS	6-26
SNUMBER	6-27
SPLIT	6-28
SRECOVER	6-29
STIMING	6-29
Points to note	6-30
Examples	6-31
LINE command	6-32
DATA	6-33
FCB	6-33
FONTINDEX	6-34
MARGIN	6-35
OVERPRINT	6-36
PCC	6-37
PCCTYPE	6-38
UCSB	6-39
VFU	6-39
Points to note	6-40
Examples	6-42
Example 1	6-42
Example 2	6-42
Example 3	6-43
MESSAGE command	6-44
ITEXT	6-45
OTEXT	6-46
OUTPUT	6-47
BFORM	6-50
COLLATE	6-51
COPIES	6-51
COVER	6-52
CYCLEFORMS	6-53
DENSITY	6-54
DESTINATION	6-54
DUPLEX	6-55
FACEUP	6-55
FEED	6-56
FORMAT	6-57
FORMS	6-58
GRAPHICS	6-59

IMAGE	6-60
INVERT	6-61
MODIFY	6-62
NTO1	6-63
NUMBER	6-64
OFFSET	6-65
OSTK	6-66
PAPERSIZE	6-67
PURGE	6-68
RESOLUTION	6-68
SF1FUNCTION	6-69
SF2FUNCTION	6-69
SHIFT	6-70
SIZING	6-71
STOCKS	6-72
SYSPPR	6-73
TMODE	6-74
TRANS	6-75
UNITS	6-75
XSHIFT	6-76
XMP	6-77
Points to note — OUTPUT command	6-78
Examples	6-87
Example 1	6-87
Example 2	6-87
Example 3	6-87
Example 4	6-87
Example 5	6-88
PDE command	6-89
Multiple logical pages on a physical page	6-91
BEGIN	6-92
FONTS	6-93
PMODE	6-93
Points to note	6-94
Examples	6-95
Example 1	6-95
Example 2	6-95
Example 3	6-95
ROUTE command	6-96
Cataloged RTEXT files	6-96

	RFORM	6-97
	RTEXT	6-98
	Example	6-99
STOCKSET command		6-100
	ASSIGN	6-100
	INIFEED	6-101
	SYSPAGE	6-101
	Points to note	6-102
	Example	6-102
VFU		6-103
	ASSIGN	6-104
	BOF	6-104
	TOF	6-105
	Points to note	6-105
	Example	6-106
7. Using logical processing		7-1
Logical processing commands		7-1
	Logical processing command format	7-2
	Logical processing commands with TEST parameters	7-2
CRITERIA command		7-3
	CHANGE	7-4
	CONSTANT	7-5
	LINENUM	7-5
	Test expressions	7-6
	Specifying one CRITERIA command	7-6
	Specifying two CRITERIA commands	7-6
	Constant mode	7-7
	Change mode	7-7
	LINENUM parameter	7-7
	Combining change and constant modes	7-8
	Points to note	7-8
	Examples	7-9
	Example 1	7-9
	Example 2	7-9
String comparison concepts		7-10
	String comparisons	7-10
	Character types	7-10
	Masked comparisons using default type assignments	7-11
	Masked comparisons using non-default type assignments	7-12
BANNER command		7-13

HCOUNT	7-14
HJOBNO	7-14
HRPTNA	7-15
TCOUNT	7-15
TEST	7-16
TYPE	7-16
Points to note	7-17
Examples	7-17
Example 1	7-17
Example 2	7-17
BSELECT and BDELETE commands	7-18
TEST—BSELECT and BDELETE commands	7-18
TEST	7-18
Points to note	7-18
Examples	7-19
Example 1	7-19
Example 2	7-19
RAUX command	7-20
TEST	7-20
Points to note	7-21
Example	7-21
RFEED command	7-22
TEST	7-22
Points to note	7-23
RSELECT and RDELETE commands	7-24
TEST	7-24
Points to note	7-24
Example	7-25
ROFFSET command	7-26
PASSES	7-26
TEST	7-27
Points to note	7-27
Example	7-28
RPAGE command	7-29
SIDE	7-30
TEST	7-31
WHEN	7-31
Points to note	7-32
RSTACK command	7-35
RSTACK delimiter modes	7-35

Delimiter display	7-36
Delimiter on accounting page	7-36
Status display	7-36
Online RSTACK usage	7-36
ACCTINFO	7-37
DELIMITER	7-38
HRPTNA	7-38
PRINT	7-39
TEST	7-39
Points to note	7-40
Example	7-40
RSUSPEND and RRESUME commands	7-41
BEGIN	7-42
TEST	7-42
Points to note	7-43
Example	7-44
TABLE command	7-45
CONSTANT	7-45
MASK	7-46
Examples	7-47
Example 1	7-47
Example 2	7-47
Example 3	7-48
Example 4	7-48
8. Specifying dynamic job descriptor entries (DJDEs)	8-1
Benefits of using DJDEs	8-2
Page- and record-oriented DJDEs	8-3
Page-oriented	8-3
Record-oriented	8-6
IDEN command	8-7
OFFSET	8-7
OPRINFO	8-8
PREFIX	8-8
SKIP	8-9
Points to note	8-9
DJDE record specification	8-10
Application of DJDEs	8-12
DJDE operator information pages	8-14
Job parameter modification restrictions	8-15
Duplex DJDE page printing	8-17

Effect of multiple logical pages	8-18
COPIES=processing	8-18
Online DJDE restrictions	8-19
DJDE processing optimization	8-19
DJDE parameter definitions	8-20
ALTER	8-20
ASSIGN	8-21
BATCH	8-21
BEGIN	8-22
BFORM	8-23
BOF	8-24
Parameter rules	8-24
C	8-25
CANCEL	8-25
COLLATE	8-26
COPIES	8-27
DATA	8-28
DESTINATION	8-28
DUPLEX	8-29
END	8-29
FEED	8-30
FILE	8-31
FONTINDEX	8-32
FONTS	8-33
Points to note	8-34
FORMAT	8-34
FORMS	8-35
Points to note	8-35
GRAPHICS	8-36
IMAGE	8-38
INVERT	8-39
ITEXT	8-40
JDE	8-40
JDL	8-41
MAP	8-41
MARGIN	8-42
MODIFY	8-43
NUMBER	8-44
OTEXT	8-45
OVERPRINT	8-46

PMODE	8-47
RFORM	8-47
RTEXT	8-48
SAVE	8-49
SEFMAP	8-50
SEPARATORS	8-51
SF1FUNCTION	8-52
SF2FUNCTION	8-52
SHIFT	8-53
SIDE	8-54
SNUMBER	8-55
SPLIT	8-56
SRECOVER	8-57
STIMING	8-58
STOCKS	8-58
TMODE	8-59
TOF	8-59
TRANS	8-60
XSHIFT	8-60
Points to note	8-61
Examples	8-63
Example 1	8-63
Example 2	8-63
DJDE FILE parameter	8-64
Points to note	8-65
File effectiveness	8-65
File replacement and deletion	8-65
Card-image file processing	8-66
LPS-labeled file processing	8-68
Delimited records for LPS-labeled files	8-69
9. Using graphics	9-1
Graphics considerations	9-1
Input for graphics	9-2
Processing modes	9-2
Random mode	9-2
Move mode	9-2
Block mode	9-3
Batch mode	9-3
Tape formats	9-3
Noninterleaved	9-3

	Document interleaved	9-4
	Page interleaved	9-4
	Batch mode	9-5
Online formats		9-6
	Noninterleaved	9-6
	Document and page interleaved	9-6
	Batch mode	9-7
	Document interleaved graphic file transfers	9-7
	Management of image files	9-7
PDL command and DJDE options for graphics		9-8
Performance		9-9
	Random mode	9-9
	Online	9-9
	Document interleaved file creation	9-9
Restrictions		9-10
	Graphic feature restrictions	9-10
A. PDL command and DJDE summary		A-1
	Conventions	A-1
B. PDL command quick reference		B-1
	Conventions	B-1
C. Character code assignment		C-1
	IBM BCD code set	C-1
	Honeywell 200/2000 BCD code set	C-2
	Honeywell 6000 BCD code set	C-3
	Fielddata translation	C-4
	UNIVAC ASCII character set	C-5
	Standard ASCII character set	C-6
	Standard EBCDIC character set	C-7
	Xerox EBCDIC to extended ASCII hexadecimal translation values	C-8
D. Offline specifications		D-1
	Input unpacking examples	D-1
	Valid host computer and label specifications	D-2
	Host system JDLs on system software tape	D-4
	Interpress data from magnetic tape	D-5
	Tape format	D-6
	Points to note	D-7
	Usage requirements	D-8
	Sample JSL	D-9
	Sample layout of data (Interpress)	D-10

E. Editor quick reference	E-1
Creating a new file on system disk	E-1
Obtaining files already stored on system disk	E-3
Displaying the text of a file	E-4
Modifying the text in a file	E-5
Modifying a portion of the work file line	E-6
Modifying entire lines	E-8
Saving your source code file	E-9
Terminating the editing session	E-9
Glossary	GLOSSARY-1
Index	INDEX-1

Introduction

This document is part of a reference set designed to help you receive maximum benefit from your Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System (LPS).

About the reference set

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 96/DocuPrint 96MX Laser Printing System document set

The Xerox DocuPrint 96/DocuPrint 96MX LPS document set includes the following:

**Xerox DocuPrint 96/DocuPrint 96MX
LPS Operator Guide**

This reference contains the following information:

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

**Xerox DocuPrint 96/DocuPrint 96MX
LPS PDL Reference**

This reference contains the following information:

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

**Xerox DocuPrint 96/DocuPrint 96MX
LPS System Generation Guide**

This reference contains the following information:

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

**Xerox DocuPrint 96/DocuPrint 96MX
LPS Operations Reference**

This reference contains 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 96/DocuPrint 96MX
LPS Forms Creation Guide**

This reference contains 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 96/DocuPrint 96MX
LPS Operator Command Summary
Card**

This reference provides a quick reference of commonly-used commands.

**Xerox DocuPrint 96/DocuPrint 96MX
LPS Message Guide**

This reference contains the following information:

- OSS and other messages
- Meaning and recovery procedures

**Xerox DocuPrint 96/DocuPrint 96MX
LPS Installation Planning Guide**

This reference contains the following information:

- LPS basic components and options
- Tasks that must be accomplished before installation
- Preinstallation requirements
- Installation process
- Postinstallation activities

Xerox LPS Tape Formats Manual

This reference contains the following information:

- Characteristics of different formats
- File organization
- Data formats
- Carriage control conventions

**Xerox LPS Standard Font Library Font
User Guide**

This reference contains the following information:

- Font naming conventions
- Listing of standard fonts
- Data Sheets
- Glossary of typography terminology

Helpful Facts About Paper

This reference contains the following information:

- Selection and guidelines
- Storage
- Specifications for different printers

About this manual

The *Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System PDL Reference* describes the print description language commands and dynamic job descriptor entries used to control the printing of jobs on the Xerox DocuPrint 96/DocuPrint 96MX LPS.

The *Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System PDL Reference* is divided into the following chapters and appendices:

Chapter 1. Overview: Overview of laser printing system (LPS) components and functions, print description language (PDL) features, key terms, job flow, input and output processing.

Chapter 2. Print description language (PDL): Job source library overview, the structure of PDL commands and JSLs, job source library command levels, hierarchy of replacement, compilation, and error processing.

Chapter 3. Creating a job source library (JSL): Steps and decision points in creating a JSL, syntax quick reference list, hints and tips, compiling JSLs, and starting print jobs.

Chapter 4. Specifying input parameters: Explanations, syntax, usage, parameters, and examples of the basic input processing commands: BLOCK, CODE, PCC, RECORD, SEFFNT, TCODE, and VOLUME, online and offline mode considerations, examples of input command usage, and points to note.

Chapter 5. Defining clusters: What clusters are and how to define them in your JSLs.

Chapter 6. Specifying print format commands: Explanations, syntax, usage, parameters, and examples of the various input and output processing commands, online and offline specific commands, and points to note.

Chapter 7. Using logical processing commands: Explanations, syntax, usage, parameters, restrictions, and examples of the various logical processing commands, test expressions, string comparisons, and online and offline usage.

Chapter 8. Specifying dynamic job descriptor entries (DJDEs): The purpose and benefits of using dynamic job descriptor entries, page and record oriented DJDEs, parameters, application of DJDEs, DJDE operator information pages, job parameter modification restrictions, duplex printing with DJDEs, online restrictions, and optimizing DJDE processing.

Chapter 9. Using graphics: PDL parameters for graphics, restrictions, and points to note regarding calling out graphics in your JSLs.

Appendix A: PDL commands and DJDE summary.

Appendix B: PDL command and DJDE syntax quick reference.

Appendix C: Character code assignment tables.

Appendix D: Offline specifications.

Appendix E: Editor command quick reference.

PDL syntax conventions used in this manual

Table 1 lists the syntax conventions and their usage.

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>

This chapter provides an overview of PDL related information you will need in order to effectively utilize the PDL capabilities.

PDL features and functions

Print Description Language (PDL) is used to describe printing jobs to a Xerox laser printing system (LPS). PDL accomplishes this by:

- Describing the input (type, format, characteristics)
- Describing the processing functions (logical processing)
- Describing the output (type, format, font selection, accounting options).

Diverse application needs can be met because PDL enables you to:

- Change and mix font types on a page-to-page, line-to-line, or character-to-character basis. Output can be customized for specific needs, for example, highlighting important headings by changing font styles and sizes.
- Change page orientation and positioning on a page-to-page basis. Characters may be printed horizontally or vertically with equal ease. The printing system switches instantly between horizontal and vertical page formats, combining the two styles within a single report.
- Print a number of previously separate logical pages on the same physical page of a document.
- Modify documents on a copy-to-copy basis by printing selected portions of data on a page-to-page basis. You can replace certain portions of text with other data, delete paragraphs from some copies, or label other copies "confidential."
- Merge variable print data with forms stored on the system disk. This eliminates the need for forms overlays and most preprinted forms, as well as assuring perfect registration.
- Add data, position it on the page, and print it on a variety of forms in one job. Multiple forms, stored in digital format, are changeable on a page-to-page or copy-to-copy basis.
- Print two different forms back-to-back (duplex) on one sheet of paper, therefore reducing paper costs. Additionally, this option offers potential savings in inventory, filing, storage, and mailing costs for computer-generated material.
- Feed paper either short-edge first or long-edge first to accommodate a wide variety of paper sizes.

Before discussing PDL commands in detail, a general understanding of LPS components and functions is helpful. The following sections provide such a general overview.

LPS component types

LPS components consist of hardware and software.

Hardware

Hardware refers to all the physical components of the LPS. Examples are the tape drives, the keyboard display, the high-capacity feeder-stacker, and the physical subsystems of the system controller and the printer. Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operator Guide* for detailed descriptions of the hardware components of your LPS.

Software

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

The hardware components consist of the Advanced Image Subsystem (AIS).

Advanced Image Subsystem (AIS)

The Advanced Image Subsystem (AIS) consists of three printed wire boards which contain character dispatcher, font and graphic memory. The three boards read the character and image data and convert it into video format for printing. With AIS, imaging a page is a two-stage process performed by the hardware/firmware combination:

- Step 1.** In the first step, referred to as "fill", the video format is stored into a 64-megabit memory called a "page buffer." The entire page is processed and stored into this memory resulting in a bitmapped image.
- Step 2.** In the second step, referred to as "dump", the page buffer contents are sent, one scan line at a time, to the laser.

Two page buffers enable one page's video to be "filled" into one buffer at the same time that the previous page's video is being "dumped" from the other page buffer.

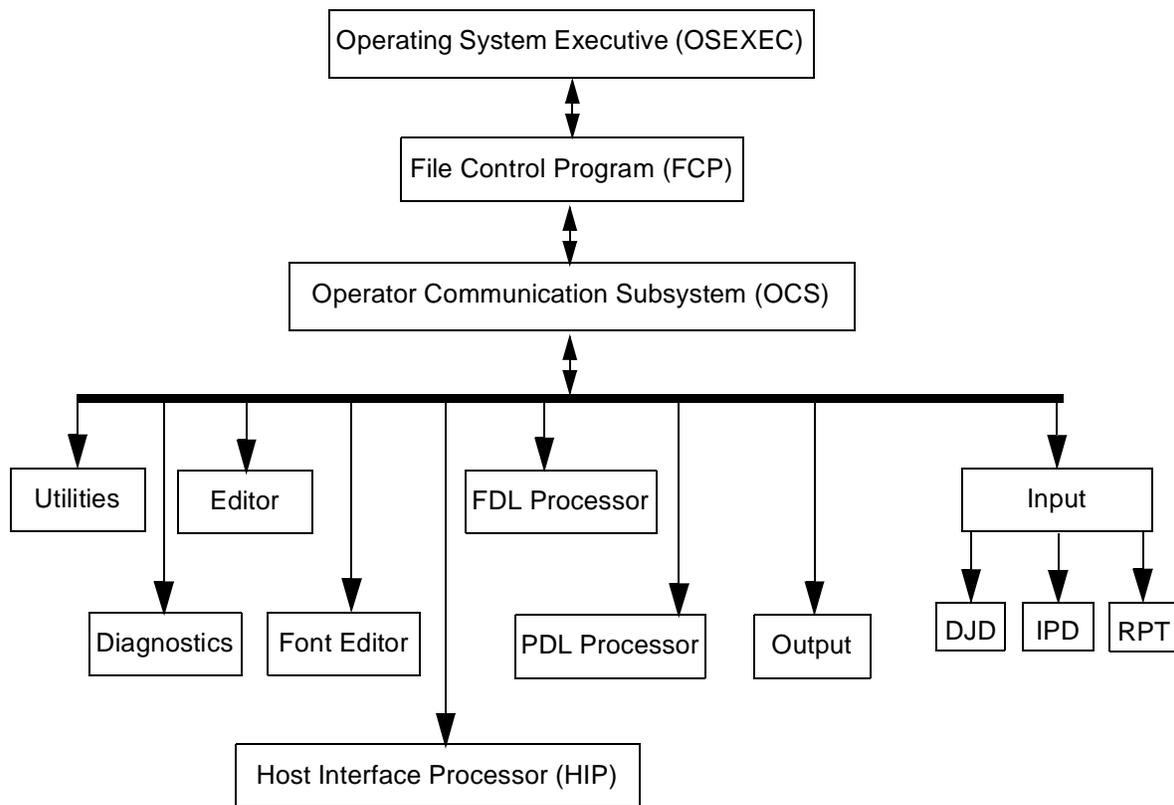
LPS software components

The LPS operating system software (OSS), housed in the system controller, runs both background and foreground processing. The major programs or “tasks” run by the system controller are shown in their hierarchical relationships in the list below. 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 processing task
- Ink catalog search (ICS)
- File transfer (XFR)
- Report processing (RPT)
- Memory control program (MCP)
- Terminal ancillary control program (TACP)
- Ancillary IOT monitor program (AIM)
- Dynamic job descriptor (DJD) task
- Interpress decomposer (IPD) task
- Output processing task
- Print description language (PDL) processor
- Forms description language (FDL) processor
- Host interface processor (HIP) task.

Figure 1-1 shows the relationships between these operating system software tasks which load or invoke LPS operations.

Figure 1-1. **Major LPS processing tasks**



PDL related programs and tasks

This section lists the programs and tasks most important to PDL.

Operating system executive task (OSEXEC)

The operating system executive (OSEXEC) task is always running. 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)

The file control program (FCP) manages disk resources. It manages and allocates all disk space, creates all disk files, and accesses disk files.

Operator communication subsystem (OCS)

The operator communication subsystem (OCS) task 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.

Editor task

The 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 and directs CMD files to execute.

Font editor task

The font editor task creates and modifies font files and allows you to tailor fonts within a font file to meet the needs of each print job.

Input processing task

The input processing task reads in job parameters (JDL file), creates a job control block, reads data, and handles job messages to the operator. Then the input processing task does the following:

- Unpacks and converts the data
- Selects and deletes blocks or records
- Records any special processing instructions (for page offsets, DJDE records, CMEs).

Report task (RPT)

The report task works in connection with the input task. It 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

The dynamic job descriptor (DJD) task compiles the dynamic job descriptor entries (DJDE) in the input data stream. The DJDEs give the LPS instructions for printing based on IDEN command parameter specifications in the input data stream.

Output processing task

The output processing task uses the page logs written by the input task to load fonts, graphics, and variable data in the Advanced Image Subsystem (AIS). It coordinates the activity of the AIS with the printer. In addition, output manages delivery of the printed pages to the correct bins, performs page recovery if necessary, and performs accounting functions.

Print description language (PDL) processor

The PDL compiler compiles an ASCII job source language (JSL) file into object code and saves it as a job descriptor library (JDL) file. It also creates ancillary files like PDE, CME, STK, LIB, etc.

Forms description language (FDL) processor

The FDL compiler compiles an ASCII form source language (FSL) file into executable object code and saves it as an (FRM) form file.

Preparing for a print job

Before a print job runs, the system administrator, programmer, operator, and LPS work together:

- System administrator loads fonts, logos, and signatures to the printer from tape, floppy, or the host. Refer to the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for detailed information.
- System administrator determines the fonts, forms, and graphics system parameters needed by the job.
- Programmer creates the JSL and FSL source files.
- Programmer invokes the FDL compiler to compile the FSL files into form (FRM) files for the print job.
- Programmer invokes the PDL compiler to compile the source files into CME, TST, PDE, JDL, LIB, and STK object files for the print job.
- Operator resets the system parameters to match the fonts, forms, and graphic requirements of the job.
- LPS completes any foreground processing and waits in IDLE status for the operator to enter the START command with the JDE and JDL for the print job.

Job flow process

The job flow process consists of OCS processing, Input processing, and Output processing.

OCS processing

The steps required during OCS processing are summarized below.

- 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 OSEEXEC task to load output, input, DJD, and IPD tasks.
- Step 2.** The OCS task then 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 name and the number of copies.

Input processing

The steps required during Input processing are summarized below.

- Step 1.** The RPT task first initializes the job queue buffer from the start JDE/JDL information by recording job characteristics such as block length, fonts needed, and required forms.
- Step 2.** The input task finds the disk addresses of the compiled JDE, form, font, and logo files to be used in printing the variable data.
- Step 3.** If the print job is in Interpress format, the IPD converts the data to ASCII format for the input task.
- Step 4.** The input task reads the variable data.
- Step 5.** The input task formats the variable data and writes it to the data buffer in the print file PRFIL1.SYS. For each page in the data buffer, the input task creates a page log entry in the page log buffer and writes the buffer to the PRFIL1.SYS file.
- Step 6.** When the input task processes all the data, or reaches page threshold (uses a certain portion of the area reserved for the PRFIL1.SYS file), it notifies the OCS task that the output task can begin.

Output processing

The steps required during Output processing are summarized below.

- Step 1.** 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.
- Step 2.** The output task reads the starting page log buffer and builds an entry in its own page tracking buffer. The page tracking buffer contains the information output processing uses for page creation (text, data, form, and font addresses, counts, and flags for options such as simplex or duplex).
- Step 3.** The output task copies portions of the font, graphic, and logo files from disk to the Advanced Image Subsystem (AIS). The AIS is instructed to create the page image from these specifications and store it in its private page buffer.
- Step 4.** The output task instructs the AIS to send the page image to the printer. As the AIS sends the image, the output task uses the page tracking buffer to monitor and record delivery of the printed page to the bin.
- Step 5.** The output task identifies each page successfully printed and delivered. The input task can then reuse that page space in the page buffer to process an incoming page.

The output task also identifies each page unsuccessfully printed or delivered. The output task uses the page buffer and log information stored in the PRFIL1.SYS file 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 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 displays the JOB XXXX HAS COMPLETED INPUT PHASE. message to the operator.
- Step 9.** When the output task finishes printing, it messages the OCS task that the job has completed printing.
- Step 10.** The OCS task displays the following message to the operator:

JOB XXX HAS COMPLETED PRINTING.

Input data types

The LPS can work effectively in many different environments, and it has the capability to handle input from a wide variety of sources.

Whether you are using the LPS in an offline capacity, over an Ethernet network, connected directly to a host computer, or remotely over phone lines, input data for printing is sent to the printer in one of two forms:

- Unformatted
- Formatted.

Unformatted data

Unformatted data—that is, raw data from a computer file—is selected, formatted, and combined with form data (boxes, lines, headings, totals, and so forth) and graphics files to produce a printed report.

- The raw data can be sent to the LPS offline from magnetic tape, online from a channel-attached host, or through an Ethernet network.
- The forms definition commands used to select records and format the printed output can be entered directly from the keyboard display, or from a host- or PC-based forms definition software package.

If you are using the LPS to create reports or other documents from unformatted data, several elements are required to complete the job:

- **Variable data.** Variable data is the part of the report that changes from page to page. In the example of an inventory report, the variable data would be the part numbers, descriptions, prices, costs, and so forth.

The variable data can be input from the magnetic tape system in the offline mode, from a host through a channel interface, through an Ethernet network interface, or from a remote host over phone lines.

- **Form data.** Form data can include headings, boxes, lines, and graphic image files, such as signatures or logos. Form data is entered in the form of compiled files, comprised of JDL and FDL statements entered through the LPS editor in the offline mode, or in the online mode through one of several host-resident forms design packages.

- **Processing data.** Processing data is optional and it allows the operator to control the output of selected reports, or selected copies of a multiple copy report, for cover-to-cover print processing on any job. For example, you may wish to specify that an inventory report has a blue card stock cover, 49 pages of the report, and a blue card stock back cover. You may also decide that four copies without cost information are needed for distribution to clients. The three command sets described below provide output control:
 - **JDE.** Gives the operator control over the mechanics of a particular print job. JDE commands specify the feed and output trays, simplex or duplex printing, stapling, collating, and so forth.
 - **DJDE.** Enables you to modify the printing environment dynamically. These commands are inserted into the input data stream to modify the command characteristics of the existing job descriptor entry (JDE). DJDEs can take effect on a report-to-report, page-to-page, and record-to-record basis.
 - **CME.** Enables you to replace certain parts of a report with predefined static data on selected copies or to specify font changes within the variable data.

Page description and report data from host-resident software is sent to the LPS in a form that it understands; therefore, no additional formatting commands are required.

Many such host-resident software packages communicate with the LPS in page description languages.

Formatted data

Formatted data is sent to the LPS from a host-based document composition software package, for example, XPPI, XDGI, or PC-based software through the front-end processor. These systems are often used for electronic publishing and can produce very sophisticated printed documents. Data from these sources come in a form that the Xerox DocuPrint 96/DocuPrint 96MX LPS can already understand; there is no need for the operator to create forms definition files or the like, as it is with the unformatted data described above.

2. Print Description Language (PDL)

This chapter discusses the following PDL topics used to create and control print jobs.

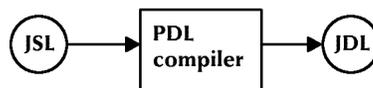
- Purpose of PDL
- PDL command structure
- JSL structure
- Creating separate command files
- Hierarchy of replacement

Purpose of PDL

For a job to be printed on an LPS, it is necessary to create a file of PDL commands to define the format of the input media, processing requirements, and the format of the printed output. The source or uncompiled file of PDL commands is referred to as a job source library (JSL) file. All JSL files must be compiled before they can be referenced to print a job. The object or compiled file of a job source library file is referred to as a job descriptor library (JDL).

This compilation process is depicted in figure 2-1.

Figure 2-1. **PDL compilation**



Each command has a set of parameters that can be used to define the characteristics of a print job (input, special processing, output). PDL commands used in creating a JSL may be entered at:

- Host, using a host-based editing facility to create 80 byte EBCDIC records.
- LPS system controller. Refer to the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*.

PDL command structure

The primary element of a JSL is a job. A job, which is one printing task, is referred to as a job descriptor entry (JDE). (In PDL, the terms "job" and "JDE" are used interchangeably.) It usually defines one input format, one set of processing instructions, and one set of output instructions. Each job has a user-defined name that you invoke to run the job.

To produce a finished job or application, a JSL must be created and then compiled into a JDL file. To accomplish this, you must use PDL commands and be knowledgeable about PDL command structure, which includes the following topics:

- Command line length
- Command components
- Right-part constants

Command line length

Table 2-1 illustrates a set of typical PDL commands. PDL commands consist of command lines, also called records, whose length can be up to 133 characters for JSLs on tape, of which only characters 1-72 may be used for parameter information. Command lengths less than 72 characters are acceptable. Each PDL command consists of a command keyword and one or more parameters separated by commas or spaces. Spaces are also permitted around the equal sign (=).

Commands may be continued on successive records if parameters are separated by commas. Crossover is performed from one record to the next when column 73 is reached or when the end-of-record is reached for records of fewer than 72 characters. Multiple commands may appear on one record if separated by semicolons.

Command components

The components of a PDL command are:

- Command identifier, if required
- Command keyword
- One or more command parameters
- Parameter options
- Comments, if appropriate

These are shown in figure 2-2.

Parameter

Each command keyword is followed by parameters used to select its processing parameters. The parameters for a PDL command keyword consist of a left and right part separated by an equal sign (=).

Table 2-1 represents the typical components of a command statement and provides examples.

Table 2-1. **Set of typical PDL commands**

Identifier	Command keyword (required)	Parameter keyword	Parameter option	Additional parameter keywords
VFU1: CME4:	VFU CME	ASSIGN= LINE =	(1,1), (1,60),	TOF=1,BOF=55; POSITION=5, FONT=2;

Comments

Comments are statements you include in the source file to describe certain PDL commands and their functions. These comments can act as reminders if you, or someone else modifies the JSL at a later time.

Comments may appear anywhere within the JSL. They must be preceded by the character sequence slash and asterisk (/*), and terminated by the character sequence asterisk and slash (*). Examples are illustrated in figures 2-3, 2-4, and 2-5. Nested comments may be set within another comment. There is no practical limit to the level of nesting possible, as long as each nested comment is preceded by a slash and an asterisk (/*) and succeeded by an asterisk and a slash (*). An acceptable nested comment format is as follows:

```
/*comment
/*nested comment*/
*/
```

Syntax rules

When entering your JSL records on the system controller keyboard, make sure to follow these rules:

- Use commas or blanks to separate the individual left- and right-part parameters of a command.
- Use parentheses to enclose multiple right parts.
- List parameter options in the sequence shown in this manual. To specify a particular option but not the options preceding it, use commas or blanks as “place holders” for the options you do not specify. For example, the OUTPUT command BFORM parameter has three options:

```
BFORM=(form-id,[init][,copies])
```

To specify the form name (form-id) and number of copies, but not the initial copy (init) on which the backside form (BFORM) is printed, enter:

```
BFORM=(SMLFRM,,2)
```

The second comma (,) after SMLFRM tells the system that “2” specifies the number of copies on which the form is printed.

- Use blanks anywhere in the JSL except in keywords and constants.
- Abbreviate command and parameter keywords to the first three letters, for example, POSITION or POS, OUTPUT or OUT. The only exception is FOR, which the system interprets as the parameter FORMAT instead of FORM. Therefore, make sure to use the abbreviation FOR to represent FORMAT only, or avoid the abbreviation entirely to prevent errors.
- Use a semicolon (;) to indicate the end of an element of data for the system. It must be at the end of every PDL command.
- Enter command parameters such as FONT, FORM, and GRAPHIC in their singular form as shown, or with an optional plural “s,” such as FONTS, FORMS, and GRAPHICS.
- Enter the END command to signal the end of a JSL. You may then enter another JSL into the system if you wish. Use two END; commands to signal the end of all JSLs to be processed:

```
END; END;
```

Example

```
LINE VFU=VFU1, DATA=(1,10),  
      OVERPRINT=(PRINT,DISP);
```

This LINE command example contains three left-part command parameters, VFU, DATA, and OVERPRINT, a right-part reference to an identifier, VFU1, and parameter options (1,10) and (PRINT,DISP).

Right-part constants

Constants within the right part of a left/right-part parameter may be either value or string constants. The syntax of these constants is defined below.

Value constants

Value constants are constants that have arithmetic values. They should be expressed as decimal numbers. They may be expressed as hexadecimal values, octal values, or even character values, but these expressions are not recommended. Decimal constants may be signed and in some cases may have fractional digits, for example:

```
PDE BEGIN=(1.1, .37);  
BLOCK LENGTH=1320;  
RECORD LENGTH=132;  
IMAGE=(1.30 CM, 0.85 IN);
```

String constants

String constants are normally used to specify strings of characters or to reference identifier parameters. The length of string constants is important. String constants may be expressed as any of the following:

- Keyword
- Variable name
- Hexadecimal
- Character
- ASCII
- EBCDIC
- Octal
- H2 or H6

Keyword

Keywords are terms that direct the system to perform specific predetermined activities. Keywords always consist of the same characters and do not vary. For example:

```
BLOCK      ZERO=YES;  
ABNORMAL  ERROR=CONTINUE,  
           OTEXT=WAIT;  
ACCT      USER=BOTH;
```

Variable name

String constants may be used to specify names of forms, files, fonts, departments, and so on. In creating your JSLs, you assign names to the forms and files you want to specify. Each name you assign identifies the unique object you wish the system to act upon for your applications. For example:

```
OUTPUT  FORM=SMPLE,  
        BFORM=SMPBK,  
        FEED=BLUCVR,  
        MODIFY=CME12;
```

Hexadecimal Normally used as string constants, but they may also be used as value constants. Each pair of hexadecimal characters results in one byte. A hexadecimal constant must immediately be preceded by the characters X apostrophe (X) to indicate to the PDL compiler that the following expression is in hexadecimal. For example:

```
IDEN PREFIX=X'C1C2C3C4' ;
```

Character Normally used as string constants, but they may also be numeric value constants. Each character, including embedded blanks, results in one byte. A character constant must immediately be preceded and immediately followed by the apostrophe (') character. For example:

```
IDEN PREFIX='THIS IS A CHARACTER CONSTANT' ;
CONSTANT='ABCDE' ;
```

If the apostrophe character (') is required in a character constant, it must be defined in some other way, such as consecutive or double apostrophes ("), or the hexadecimal constant X'7D'. Character constants may be defined as EBCDIC and take their actual values from the standard EBCDIC table definition in appendix D.

ASCII Used as string constants. Each character results in one byte. The constants must be preceded by the characters A apostrophe (A) and followed by an apostrophe character. For example:

```
IDEN PREFIX=A'ABC' ;
```

The ASCII string type allows hexadecimal representation of characters to be embedded in a string. This is done by preceding the hexadecimal representation of the character with an ! character. For example:

```
IDEN PREFIX=A'ABC!44EF'
```

is equivalent to

```
IDEN PREFIX=X'414243444546'
```

The three-character sequence required for a hexadecimal representation of a character results in one byte.

Two successive ! characters (!!) are necessary to represent one actual ! character when printing. The two-character sequence (!!) results in one byte.

EBCDIC EBCDIC constants are used for value and string constants. They must be preceded by the characters E apostrophe (E) and followed by an apostrophe character ('). The EBCDIC string type allows hexadecimal representation of characters to be embedded in a character string. This is done by preceding the hexadecimal representation of the character with an ! character.

For example:

```
IDEN PREFIX=E'ABC!C4EFG'
```

is equivalent to the hexadecimal

```
IDEN PREFIX=X'C1C2C3C4C5C6C7'
```

Each character represented in EBCDIC results in one byte. Each three-character sequence representing a character hexadecimally results in one byte.

Note that EBCDIC is the default, therefore the E 'xxx' is usually not required.

Octal Octal constants should be used only as string constants because of the control program conversion process. Each octal character results in 3 bits. One word can store 3 characters. Their use as value constants, however, is not prohibited. Each 3-bit octal character is converted to an 8-bit octal character, internally, by prefixing two binary zeros. Thus, the arithmetic value of a multiple-character octal constant may be difficult to determine because each digit in the constant has been altered. An octal constant must be preceded immediately by the characters letter O apostrophe (O) and immediately followed by the apostrophe (') character. For example:

```
BLOCK CONSTANT=O'07070707'
```

H2 and H6 H2 and H6 constants generate H2000 BCD and H6000 BCD codes, respectively. Use of H2 and H6 is identical to use of E and A prefixes described above. For example:

```
BLOCK CONSTANT=H2'373737'
```

```
BLOCK CONSTANT=H6'373737'
```

Since H2000 and H6000 BCD are defined as 6-bit codes (refer to appendix C), no specification greater than X'3F' generates a legal character. If anything from X'40' to X'FF' is coded, PDL generates an error message and replaces the bad character with a blank.

String constants may be preceded by an optional repeat count. A repeat count is enclosed in parentheses and must be in the range of 1 to 255. For example, the command:

```
T1: TABLE CONSTANT=(3)'**';
```

is equivalent to:

```
T1: TABLE CONSTANT=('***');
```

Other examples of the use of a repeat count are:

```
T1: TABLE CONSTANT=(3)O'27';
```

```
T2: TABLE CONSTANT=(4)X'C1';
```

Table 2-2 represents a JSL which includes the components of PDL commands and shows a variety of right-part constants.

Table 2-2. **Sample PDL commands**

Identifier	Command keyword	Parameter keyword followed by parameter options and additional parameter keywords
VFU1:	VFU	ASSIGN=(1,1),TOF=1,BOF=55;
VFU2:	VFU	ASSIGN=(1,(4,59)),TOF=4,BOF=127;
T1:	TABLE	CONSTANT=('C#LIPS');
C1:	CRITERIA	CONSTANT=(14,5,EQ,T1),LINENUM=(6,3);
T2:	TABLE	CONSTANT=('1');
C2:	CRITERIA	CONSTANT=(0,1,NE,T2);
CME1:	CME	LINE=3,POS=59,CONSTANT='FIRST QUARTER';
CME4:	CME	LINE=(1,60),POSITION=5 FONT=2;
PDE2:	PDE	Pmode=PORTRAIT, BEGIN=(1.1,.37), FONTS=(P08TYA, P08SCA);
CODTAB:	CODE	DEFAULT=EBCDIC, ASSIGN=(X'4A',X'B4');
	VOLUME	LABEL=NONE, HOST=IBMOS, CODE=CODTAB;
	BLOCK	LENGTH=1320;
	RECORD	STRUCTURE=FB,LENGTH=132;
	LINE	VFU=VFU1,DATA=(1,10),OVERPRINT=(PRINT,DISP);
	ACCT	USER=BIN, DEPT='MEDCAR';
	OUTPUT	FORMS=XER111,FORMAT=PDE2,MODIFY=CME4;
	ROUTE	RTEXT=('ENGINEERING',2,56,109),RFORM=XERCOV;
	RDELETE	TEST=(C1,AND,C2);
	IDEN	PREFIX='\$DJDE',SKIP=10,OFFSET=1;

Job source library (JSL) structure

To simplify JSL coding, PDL commands are grouped into command levels. The use and syntax of command levels, along with the required END command, are defined in the following sections:

- ID level
- System level
- Catalog level
- Job or JDL level
- END command

The command levels are always preceded by the JDL coding, which provides the name of the compiled JDL.

Command levels

You can place PDL commands in any command level, depending on your particular application needs.

Table 2-3 outlines the command levels and some typical specifications that are included in these various levels.

Table 2-3. **Command levels and their general purpose**

Command level	General purpose
ID	Typically used to assign output channel numbers to printer carriage control channels through the VFU command, but any command which has an identifier may be used at the ID level, such as the CODE, PCC, and ROUTE commands.
System or JDL	Establishes installation defaults.
Catalog	Groups PDL commands for easy reference at the job level.
Job or JDE	Defines how individual print jobs are processed.

Figure 2-3 illustrates a sample JSL file format and provides examples.

Figure 2-3. **Sample JSL file format**

```
JDLSMP:    JDL;
           /* THIS SAMPLE JSL SHOWS THE JSL COMMAND LEVELS*/
           /* ID LEVEL COMMANDS CODED HERE*/
VFU1:     VFU      ASSIGN = (1,1), TOF = 1, BOF = 66;
VFU9:     VFU      ASSIGN = (1,9), TOF = 9, BOF = 66;
           /* SYSTEM LEVEL COMMANDS CODED HERE*/
           VOLUME   HOST = IBMOS;
           BLOCK    LENGTH = 1330;
           RECORD   LENGTH = 133;
           LINE     VFU = VFU9, DATA = (1,132);
           OUTPUT   FORMS = CPPR1, DUPLEX = NO,
                   COPIES = 2;
           /* CATALOG LEVEL COMMANDS CODED HERE IF NEEDED*/
CAT1:     CATALOG;
           LINE     VFU = VFU1;
           OUTPUT   FORMS = FORM2;
           /* JOB LEVEL COMMANDS CODED HERE*/
JOB CPP:   JOB;
JOB2:     JOB      INCLUDE = CAT1;
JOB3:     JOB      INCLUDE = CAT1;
           OUTPUT   DUPLEX = YES;
           END;
```

ID level

The ID level has commands that require identifiers so that they can be referenced by other commands in lower command levels. For example, the ID level contains one or more VFU commands, as shown in table 2-2. As with the other command levels any PDL command can be specified at the ID level. The ID level must be preceded by JDL coding, which names the JSL. For example:

```
XSML: JDL;  
VFU1: VFU ASSIGN=(1,1),TOF=1,BOF=66;
```

In this example, XSML:JDL is the name of the complete JDL and the VFU command is in the ID level.

System or JDL level

A system or JDL command set establishes installation-dependent requirements and default values for job descriptor entries. At the system level, JDL may be used interchangeably with SYSTEM. At the system level, commands are specified which apply to all job descriptor entries (JDEs) identified within a job descriptor library (JSL). Each SYSTEM command results in the creation of a JDL when compiled.

The SYSTEM command has the form:

```
jdl-name {SYSTEM JDL;}
```

jdl-name is a 1 to 6 character alphanumeric identifier specifying the name of the JDL to be created. It must contain at least one alphabetic character.

For example:

```
SAMPL: SYSTEM;
```

This command identifies the start of a SYSTEM command and the beginning of a JDL. The *jdl-name* SAMPL corresponds to the name of the JDL to be used when printing a job. When DFAULT is coded for the *jdl-name*, the specification of a JDL parameter option in the START command is not necessary.

Catalog level

The catalog level allows the coding of commands common to several JDEs. A catalog can then be referenced in an INCLUDE command in each JDE. A catalog command level is identified by the CATALOG command and ends with the appearance of another CATALOG command or a JOB command. CATALOG commands may contain the same commands which appear in the JOB command.

The CATALOG command has the form:

```
cat-name: CATALOG;
```

cat-name is a 1 to 6 character alphanumeric identifier of which at least one character must be alphabetic. The *cat-name* is referenced by JDEs after the CATALOG command set has been defined.

For example:

```
POWER: CATALOG;
```

In this command, POWER is the catalog level identifier to be used in the INCLUDE parameter of a JOB command.

For example, to reference the catalog named POWER in a job, the job level command would be:

```
JOB1:JDE INCLUDE=POWER;
```

Job or JDE level

The job or JDE level allows the grouping of individual jobs together. PDL commands coded within the job command level override the system commands. PDL commands from a catalog command level can be incorporated as shown in the command syntax below. For each job, values not specified in any of the command sets are taken from the PDL defaults as defined in appendix A. The JOB or JDE command has the following form:

```
jdl-name: {JOB JDE} [INCLUDE = (cat-name1[,cat-name2][,...])];
```

jde-name is a 1 to 6 character alphanumeric identifier. It specifies the name of the JDE being defined.

cat-name or *cat-name-n* is a 1 to 6 character alphanumeric identifier of a previously defined catalog name.

Examples–JOB command

```
JOB2: JDE;
JOB3: JOB
      INCLUDE=POWER;
JOB4: JDE;
      LINE VFU=VFU2;
      OUTPUT FORMS=AY2F;
```

A JOB command continues until another JOB, JDE command or END command is encountered. The catalog identifier in a JOB or JDE command as with JOB3 above, JOB2 is used along with the identifier on the SYSTEM command set to initiate a print job. When DFLT is coded for the jde-name, the specification of a JDE parameter option on the START command is not necessary.

END command

A JDL terminates with the END command. If one JDL is to follow another, the next command after the END command should be another SYSTEM command.

The end of all JDLs to be processed is indicated by two consecutive END commands:

```
END; END;
```

Figures 2-4 and 2-5 provide examples of offline and online JDLs.

In figure 2-4, note that HOST=POWERVS indicates the source and structure of input data. The HOST command indicates whether the JSL is for an offline or online job. Some HOST parameters can apply to both online and offline hosts. Refer to appendix A for a quick reference.

Figure 2-4. Sample offline job descriptor library

```

IBMPDL: SYSTEM;
                                                    JDL coding
/*THIS JOB DESCRIPTOR LIBRARY CONTAINS JOB DESCRIPTOR ENTRIES FOR
PROCESSING GRASP, POWER AND POWER VS JOB TAPES

THE SYSTEM STATEMENT SET DEFINES CONSTANTS AND PROCESSING PROCEDURES
THAT WILL APPLY TO ALL JOBS PROCESSED USING THIS LIBRARY UNLESS
OVERRIDDEN BY THE CATALOG OR JOB COMMAND SETS */

VFU001: VFU      ASSIGN=(1,5),ASSIGN=(2,10),ASSIGN=(3,15)
                ASSIGN=(4,20),ASSIGN=(5,25),ASSIGN=(6,30),
                ASSIGN=(7,35),ASSIGN=(8,40),ASSIGN=(9,45),
                ASSIGN=(10,50),ASSIGN=(11,55),ASSIGN=(12,60),
                TOF=5,BOF=66;
                                                    ID level

VOLUME          HOST=POWERVS,PLABEL=YES,CODE=ASCII;
BLOCK          LENGTH=2048;
RECORD         LENGTH=136,STRUCTURE=VB,LTHFLD=2,
ADJUST=0,FORMAT=BIN,PREAMBLE=3;

LINE           DATA=(1,132),PCCTYPE=IBM1403,PCC=(0,NOTRAN),
OVERPRINT=(PRINT,NODISP),VFU=VFU001;
                                                    System
ACCT           USER=(BIN,TRAY);
                                                    level

/*          CATALOG COMMAND SET FOR POWER VERSIONS */

CATPOW: CATALOG;
VOLUME        HOST=POWER,CODE=EBCDIC;
BLOCK         LENGTH=2048,PREAMBLE=6,LTHFLD=2,FORMAT=BIN,
OFFSET=4;
RECORD        LENGTH=135,STRUCTURE=VB,PREAMBLE=2,LTHFLD=2,
FORMAT=BIN,OFFSET=0,ADJUST=3;

/*          CATALOG COMMAND SET FOR GRASP */

CATGRP: CATALOG;
                                                    Catalog
VOLUME        HOST=GRASP,CODE=EBCDIC;
                                                    level
BLOCK         LENGTH=4096,PREAMBLE=0,ZERO=YES;
RECORD        LENGTH=135,STRUCTURE=VB,PREAMBLE=1,
LTHFLD=1,FORMAT=BIN,OFFSET=0,ADJUST=2;

/* THE FOLLOWING JDES ARE FOR IBM POWER GRASP VS TAPES,
POWER VERSION 4.0 TAPES */

1:JOB;
VOLUME        HOST=POWERVS,CODE=PEBCDIC;
2:JOB;
INCLUDE=(CATPOW);
VOLUME        HOST=POWER;
RECORD        LTHFLD=1,PREAMBLE=1,ADJUST=2;
3:JOB;
INCLUDE=(CATPOW);
VOLUME=HOST=POWERVS,CODE=PEBCDIC
                                                    Job
4:JOB;
INCLUDE=(CATGRP);
VOLUME        HOST=POWER;
                                                    level

END;

```

In figure 2-5, note that online systems do not define the BLOCK command.

Figure 2-5. Sample online job descriptor library

```

ONLINE:  JDL;                                     JDL coding
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=4,BOF=66;
VOLUME  HOST=IBMONL,
          OPTIMIZE=(NCC,NDC,NPR);
LINE    PCCTYPE=IBM3211,VFU=VFU1,
          UCSB=IGNORE,FCB=IGNORE;
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 PAGES       */
/*                                           */
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,HCOUNT=1;
POWER:  JOB;
          BANNER  TEST=C5,HCOUNT=2;
/*                                           */
          END;

```

ID level

System level

Job level

Creating separate files for grouping PDL commands

If you have multiple commands of the same type, such as CMEs and PDEs, you may want to create separate files for them to group like specifications together and to make your JSLs shorter and more efficient. You can create these types of command files by simply listing them as you would in a JSL and complete the list with an END; command before specifying a JSL's JDL coding identifier instead of after it.

Files containing groups of PDE and CME commands may be created as system or editor files. Because you use the editor to create and modify your JSLs, it may be more efficient for you to create these command files in the editor.

Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for detailed information on using system and editor command files.

Hierarchy of replacement

The system default values shown in appendix A are the more commonly used values in job processing; they can be thought of as a basic job descriptor entry (JDE). PDL commands need coding for only those parameters that must be changed to process your unique print jobs. This coding process may be further specified by placing commands common to more than one job in the catalog command level. When these coding features are properly implemented, it is possible for the same command to be used in more than one job or JDE command level within a library. The PDL processor evaluates user coded commands and applies the highest order, error-free definition to the job for printing. This process, termed the hierarchy of replacement, is discussed in the subsequent paragraphs and illustrated in figure 2-6.

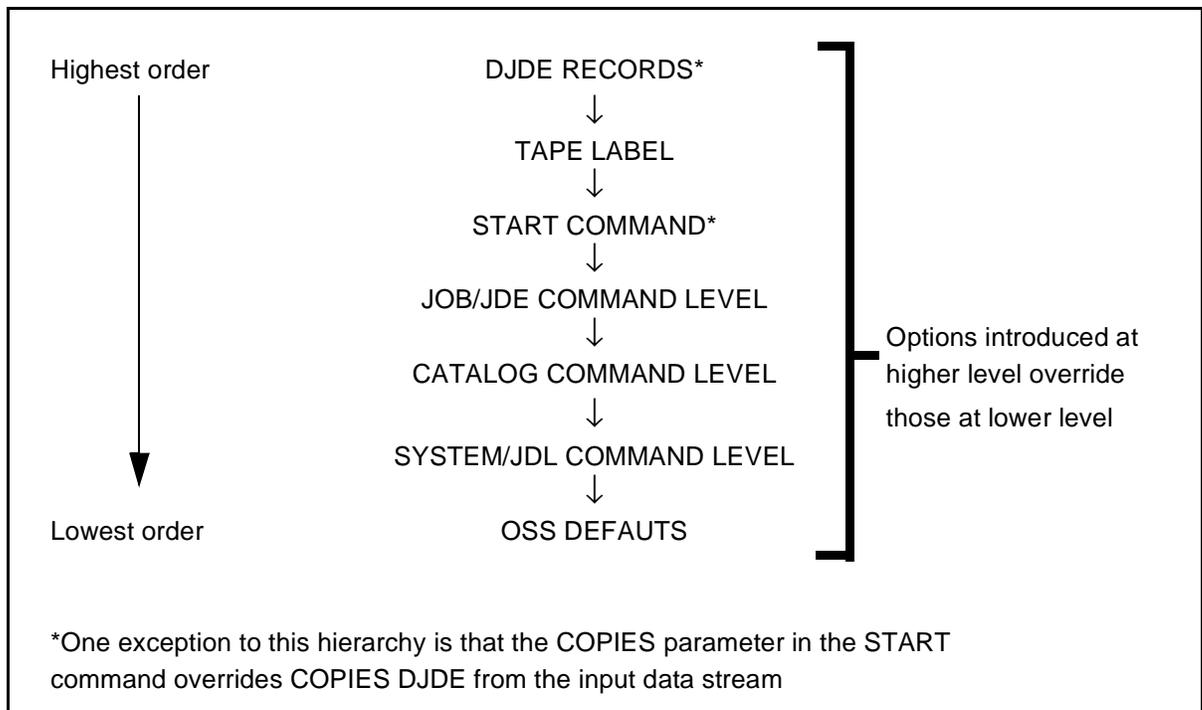
Hierarchy within a job descriptor library (JDL)

Figure 2-4 shows a coded JDL that contains four jobs or JDEs. A command to specify the recording code (CODE parameter of the VOLUME command) of the input data appears in three places:

- According to the system command level (or JDL) command set, the default recording code of the input data is ASCII (VOLUME CODE=ASCII).
- According to the catalog command level, the recording code of the input data is EBCDIC (VOLUME CODE=EBCDIC).
- According to the job or JDE command levels, for jobs one and three, the recording code of the input data is Printable EBCDIC (PEBCDIC). The PDL command: VOLUME CODE=PEBCDIC overrides both catalog and system (or JDL) command level definitions.

For job descriptor entry 2:JOB;, the recording code of the input data is EBCDIC, as specified in the CATPOW catalog command. In 4:JOB;, the recording code of the input data is EBCDIC because the JOB command's INCLUDE parameter specifies the CATGRP catalog which, in turn, specifies EBCDIC in the VOLUME command CODE parameter.

Figure 2-6. **Hierarchy of replacement**



Non-JDL hierarchy — START command

The next level of command replacement above the JOB or JDE command (as illustrated in figure 2-6) is the START command. Values specified in the START parameter override those in the job command level. When the COPIES parameter is specified in the START command, it overrides a DJDE value for COPIES from the input data stream.

START command

The START command is used to initiate printing of the application you have created. The options shown below vary depending on the input source and the output destination. The offline and online START command formats are as follows:

Offline `START [[jde] [,jdl] [,S | M] [,copies] [,REPORTS:r1,r2,...] [,TDn] | [DISC:file.ext]]]]]`

Online `START [[jde] [,jdl] [,copies] [,FORM=form]]]]`

Restrictions Note the following restrictions:

- 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.
- 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.

Table 2-4 lists the START command parameter option(s) and definition(s).

Table 2-4. **START parameter option(s) and definition(s)**

Option(s)	Definition(s)
<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	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.
REPORTS:r 1,r2,...	Allows you 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 r1,r2..., 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 six reports, they are not be processed. The REPORTS option is also useful for printing one report of a multiple report tape. This saves the step of spacing over reports not needed. A maximum of 14 values is allowed. A range of values uses 2 of the 14 maximum values allowed.
TDn	Specifies the tape device being used.
DISC	Indicates you want to print a file stored on the system disk.
FORM	Allows you to specify a form used for the job. This option overrides any form specification in the job descriptor entry (FORMS command of OUTPUT statement and the RFORM command of the ROUTE statement). The specified form must be on disk in the FRM directory.

Example 1 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 is shown in Figure 2-7 below.

Figure 2-7. 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
```

Example 2 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.

Example 3 START

Since no options are specified, 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.

Example 4 START J12,H2SY&S,,2,REPORTS:3-4

This command reprints two copies of the third and fourth reports on the data tape.

Example 5 START ,ONLINE,,,FORM=GBAR

This command processes online data according to the DFLT JDE in the online JDL file, using the GBAR form.

Final modification can be made to job descriptor parameters through DJDE parameters in the job input data stream. These DJDE parameters are discussed in the “Specifying dynamic job descriptor entries (DJDEs)” chapter.



Note: “INTERPRESS” is not a valid keyword.



Note: FORM option is only valid online

Hierarchy of replacement in an errored job descriptor library (JDL)

The following discussion illustrates the effect of errors when the PDL processor is evaluating a JDL source file. If an error occurs during compilation, it is mandatory to fix the error and recompile.

Example:

The VOLUME command CODE parameter in the catalog level below contains a syntax error (VOLUME CODE=EBDIC). If you use the incorrect JDL, the code default, ASCII, specified in the system command level is applied to the job level (job descriptor entry JOB1).

```
SAM2 : SYSTEM ;
VOLUME CODE=ASCII ;
AA : CATALOG ;
VOLUME CODE=EBDIC ;
OUTPUT COPIES=100 ;
JOB1 : JOB INCLUDE=(AA) ;
OUTPUT COPIES=50 ;
END ;
```

3. Creating a job source library (JSL)

There are many commands available to include in your job source libraries (JSLs) and many ways of organizing them. There are relatively few absolute rules and virtually infinite combinations you can use to create applications through the use of Print Description Language (PDL).

This chapter presents the following topics to use as a tool in guiding you through JSL creation:

- What your JSL specifies for LPS processing
- Decisions to make before creating your JSL
- Review of PDL components and syntax
- Helpful hints
- Steps in creating a JSL
- Compiling your JSL
- Printing the job
- Short explanations of some application-related issues:
 - Paper types
 - Page orientations
 - Registration shift and skew
 - Fonts
 - Compatibility with the 4050, 4090, 4650, 4850/4890, 9700F, and 4635 laser printing systems.

Keep in mind that many of the contents of this chapter are samples and suggestions of what can be done with PDL to create your desired applications, that there is a multitude of other possibilities and options available, and that detailed information on each topic is provided in the other chapters of this manual, and in the *Xerox DocuPrint 96/DocuPrint 96MX LPS Forms Creation Guide*.

What your JSL specified for LPS processing

If your document requires forms, you code programs in form source language (FSL).

You then code jobs in job source language (JSL). Your JSL files may call FRM files.

If your document uses color, the PDL applications you code must include the appropriate Xerox 4850 or 4890 HighLight Color LPS ink specifications. On your Xerox DocuPrint 96/DocuPrint 96MX LPS, a color JSL is compiled without errors, but the color ink is ignored and the application prints in black, the system default.

Refer to your *Xerox 4850/4890 HighLight Color Laser Printing System PDL/DJDE Reference* for details on using inks in your color applications.

Then you invoke the PDL compiler to compile your JSL file into a JDL file to tell the printer such information as the following:

- Use of black or color ink for text and images
- What variable and fixed data to use
- Placement, font, and point size for the variable and fixed data
- Which fonts, forms, images, signatures, and logos to use
- Paper stocks to use for the job
- How to feed stock for the print job, in other words, which paper trays to use
- Page layout
- How to build the print image for each page
- How many copies to print.

Decisions to make before creating your JSL

Before starting to develop the JSL for your application, there are some key decisions you must make, based on your site-specific needs and the design of the application.

Input data

In general, you should know this information about the input data before creating the JSL:

- The input source, such as the following:
 - Host supporting the 3211 or 4245 host interface
 - Reel-to-reel tape or cartridges, which are offline devices
 - Host interface processor (HIP) connection, which is a high-speed channel for data transmission to the LPS from any of several other interfaces, including XNS, XPMF-VMS, and XPF
 - System disk
 - Combination of two or more of these types.
- Computer on which the data was created
- Block and record lengths and structure
- Code in which the data is encoded, such as ASCII and EBCDIC
- How printer carriage control (PCC) information should be processed
- For tapes or cartridges only, the label format used.

Output specifications

You must also decide on some basic questions about the output:

- Will the page orientation be landscape or portrait?
- What fonts will you use?
- What forms, if any, will you use?
- Will Segment Management be used?
- What paper sizes will be used?
- Should CMEs be used?
- How will the output be delivered, such as face up or collated?
- Will the data be printed on one side of the page (simplex) or on both sides (duplex)?
- Will graphics be used?
- What types of applications your site normally prints, for example:
 - Forms
 - Reports
 - Letters
 - Customer statements using variable data, that is, information that varies from customer to customer.
- What are your site's conventions, if any, for naming forms, JSLs, files, and jobs?

This information assists you in planning your applications in terms of the type of input data to specify, the type of application to design, and how much you will need to customize the application to meet its intended purpose rather than using system defaults.

Type of application to create

When planning the specifications you will indicate in your JSL, you must first decide the type of application you want by answering the following questions:

- Will you modify an existing application or create a new one?
- Will you use a form? If so, you need to either call out an existing form in your JSL or create a new form first and then call it out in the JSL.
- Will the application be landscape (horizontal) or portrait (vertical) orientation? (Orientation is discussed later in this chapter.)
- Will you use more than one color or type of stock in the print job?

Special features

You also want to determine the following:

- Will you include operator information, such as routing sheets or messages displayed on the system controller?
- Will you select paper trays?
- Will you allow operators to override specifications in the JSL?
- What structure will appear on the printed page? For example, a letter or a large form may require an entire page but, if smaller forms are usable, you may want to print two or even four on a page.
- Will you use Dynamic Job Descriptor Entries (DJDEs) to change the application on a page or report basis?

Interactions between JSLs, catalogs, and jobs

You will also want to consider the interactions, similarities, and differences between various JSLs, catalogs, and jobs:

- What characteristics are used globally, if any, for all of the applications at your site? For example, do all jobs use the same host, format, paper size, page orientation, block or record length, test criteria, DJDEs, error responses, accounting requirements, fonts, or forms?
- What names will you call the JSL and catalogs or individual jobs within the JSL? Use names that will be meaningful to you and others who may use the application.

Review of PDL components and syntax

Before beginning the formatting of your JSL, a quick review of PDL components may be helpful. These are discussed in more detail in the "Print description language components and processes" chapter.

Command levels

There are four command levels:

- ID level, which has commands with identifiers which must be coded in the library before they can be referenced by other commands in the other levels.
- System or JDL, which establish installation-dependent requirements and default values for job descriptor entries (JDEs). These are best used for specifications that apply to the majority of your applications.
- Catalog, which consists of groups of commands
- Job or JDE, which define how individual print jobs are processed.

A job descriptor library (JDL) command must precede all of the levels. It provides the name of the JDL, which is the compiled or "object" form of the JSL.

ID, system, and catalog level commands can be referenced by commands following them. For this reason, these commands must be numbered or named.

Command components

These are the components of a PDL command:

- Command identifier
- Command keyword
- One or more parameters
- Comments, if desired

Command identifiers

Command identifiers, if required, are usually coded at the system level so that they can be referenced by commands at other levels in the JSL. The following is an example of a VFU being identified at the system level:

```
VFU1: VFU ASSIGN=(1,1),ASSIGN=(2,5),  
      TOF= 1, BOF = 66;
```

VFU= vfu-id as a parameter of the LINE command can call out the SYSTEM level VFU identifier 1 at the CATALOG or JOB levels, for example, here in catalog B:

```
CATAB: CATALOG;   LINE VFU = VFU1;
```

Command keywords

Command keywords identify the command for which various parameters may be selected. For example, the command keyword "OUTPUT" may be followed by any number of its parameters, such as the following:

```
OUTPUT    GRAPHICS=YES , DUPLEX=NO ,
          FORMAT=PDFLT , FORM=BAR ;
```

Command parameters

Each command keyword is followed by one or more parameters which give precise specifications for the print job or application. Each parameter has one or more options from which to select. For example, the OFFSET parameter of the OUTPUT command has three options: ALL, FIRST, or NONE.

Each parameter consists of a left and a right part separated by an equal sign (=).

Commas or blank spaces separate parameters.

Parentheses are used to enclose multiple right parts of a parameter, for example:

```
LINE  VFU = VFU1 , DATA=(1,132) ,
          MARGIN=(1,POS) , PCC=(0,NOTRAN) ;
```

Comments

Comments may appear anywhere, before or after a command, within a JSL and are useful for providing information for future use or to others who may be using the JSL. They must be preceded by a slash and an asterisk (/*) and terminated by an asterisk and a slash (*). A comment can be nested within another comment. Here are some examples of comments within JSLs:

Example 1 /* CATALOG FOR ONLINE ONLY JOBS*/
SMPL1: CATALOG;

Example 2 /*CHECK PAPER STOCK FOR EACH JOB*/
LETTER: JDE INCLUDE=SMPL1 ;
 OUTPUT DUPLEX=YES ,
 COPIES=10 ,
 FORMS=BKUP ,
 BFORM=NONE ;

LABELS: JDE INCLUDE=SMPL1 ;
 OUTPUT DUPLEX=NO ,
 GRAPHICS=YES ;

PDL syntax

Keep the following rules in mind when entering your PDL commands and DJDEs:

- You may abbreviate the first three letters of commands and parameters, for example, POSITION or POS, and CATALOG or CAT. The only exceptions are the OUTPUT command parameters FORMS and FORMAT. Spell these out completely.
- Use a semicolon (;) at the end of a command. A comma (,) or blank space may also be used at the end of a line.
- Use the END command plus a semicolon (END;) to signal the end of a JSL.
- Use all UPPERCASE letters in PDL. Comments, however, need not be in uppercase.

Hints and tips

The following tips may help you as you create your JSL:

- The only required elements in a JSL are the following:
 - A JDL name, which is the name of the file created by compiling the JSL
 - Job names
 - END; command at the end of the JSL.

System defaults could be used for all other specifications, although typically each application has its own specific characteristics.

- Use tab spacing to create columns for each element of the JSL: command identifiers, commands, and parameters. While not required by the system, this organization makes it much easier to identify command sets, their commands, and each command's parameters quickly. Here is a short example:

```

SMPLST:      JDL;
V1:          VFU          ASSIGN=(1,5), ASSIGN=(2,10),
              ASSIGN=(3,15), TOF=5, BOF=66;
              VOLUME     HOST=XEROX, LABEL=NONE,
              CODE=EBCDIC, PLABEL=NO;
              BLOCK      LENGTH=3990;
              RECORD     LENGTH=133, STRUCTURE=FB;
              LINE       DATA=(1,132),
              PCCTYPE=XEROX,
              PCC=(0,NOTRAN), VFU=V1;
T1:          TABLE     CONSTANT=(132) ' * ' ;
C1:          CRITERIA   CONSTANT=(2,132,EQ,T1),
              LINENUM=(1,5);
CME1:       CME        LINE=(9,-), POS=1, FONT=1;
JOB1:       JDE;
              OUTPUT     DUPLEX=YES,
              FORMS=BARS, MODIFY=CME1;
JOB2:       JDE;
              OUTPUT     FORMS=NONE,
              DUPLEX=NO, COPIES=2;
JOB3:       JDE;
              BANNER     TEST=C1,
              HCOUNT=1, TCOUNT=0;
END;

```

- If you are not sure which specifications to select, try running a job using the system defaults and then adjust the JSL to meet your requirements. You can modify an existing JSL in the same manner.
- Keep the hierarchy of replacement, described in chapter 2, "Print description language (PDL) components and processes," in mind. It is much easier to specify generic or global characteristics at the system level, for example, than to call the same specifications out over and over again for each job or catalog.

- You do not need to use all command levels in a JSL. Many JSLs have only ID, system, and job-level commands.
- Keep in mind that the specifications to select in your JSL can be changed easily. By using the IDEN command (discussed in chapter 9, "Using graphics") you can allow DJDEs to override PDL commands on a page-by-page or record-by-record basis. Also, certain operator commands can alter the print job in such areas as the number of copies to be printed, the sequencing of reports, and paper feed specifications.

Steps in creating a JSL

There are many steps in creating a JSL, and many ways in which a JSL can specify your application's requirements. The following sequence is simply one example of the format and content of a JSL to help you get ideas on how to set up your own applications.

Using the Editor

To refresh your memory on using the Editor to create or modify PDL files, refer to the appendix, "Editor quick reference."

Name the JDL identifier

The first step in creating a JSL is to give the JDL a name, which can be no longer than six alphanumeric characters, for example:

```
XRXSPL: JDL;
```

Specifying VFUs

The VFU command specifies the vertical tabbing for the print job. There can be more than one VFU identified. All are typically specified at the ID level, for example:

```
/* ID level commands*/  
VFU1:      VFU   ASSIGN=(1,11),  
           ASSIGN=(12,66),  
           TOF=11,BOF=66;  
VFU2UP:    VFU   ASSIGN=(1,(11,77)),  
           ASSIGN=(12,(66,132)),  
           TOF=11, BOF=132;  
VFU3:      VFU   ASSIGN=(1,1),  
           TOF=1, BOF=132;
```

Setting up input parameters

Next, you may want to specify the input data characteristics for the application. The basic input processing commands are BLOCK, CODE, PCC, RECORD, TCODE, and VOLUME. Input processing characteristics vary depending on the data source. For example, if your JSL is for an online application, the BLOCK command is not applicable. Also, parameters within a command may apply to offline only, online only, or both. For example, with the VOLUME command, the parameters CODE and HOST can apply to both online and offline, EOVS applies only to offline applications, and OPTIMIZE applies only to online applications.

The following is a sample of typical offline commands:

```
/*System level commands*/  
  
      VOLUME      HOST=IBMOS ,  
                  LABEL=STANDARD ,  
                  CODE=EBCDIC ,  
                  PLABEL=YES ;  
      BLOCK      LENGTH=2660 ,  
                  PREAMBLE=4 , LTHFLD=2 ,  
                  FORMAT=BIN ;  
      RECORD      LENGTH=136 ,  
                  PREAMBLE=4 ,  
                  STRUCTURE=VB ,  
                  LTHFLD=2 , OFFSET=0 ,  
                  FORMAT=BIN ;
```

Specifying LINE command parameters

The LINE command references VFUs from the ID level and allows you to instruct the system on which parts of the data in each record are to be printed. For this reason, it typically follows the RECORD command. For example:

```
LINE   DATA=( 1 , 132 ) ,  
        PCCTYPE=ANSI ,  
        PCC=( 0 , NOTRAN ) ,  
        VFU=VFU1 ;
```

Specifying ACCT command parameters

The ACCT command often follows the LINE command and, as with the other commands mentioned above, the ACCT command can be coded into any command level. Here, it accompanies the other system level commands:

```
ACCT   USER=TRAY ;
```

Specifying use of DJDEs

Most PDL commands can also appear in the form of DJDEs, which allow page-by-page or record-by-record modifications to your applications. In order to use DJDEs, an IDEN command must be specified in the JSL to advise the system where to look for them in the input data stream. For example:

```
IDEN    PREFIX='C9700', SKIP=7,
        OFFSET=1;
```

Refer to the appendix "PDL/DJDE command summary," for information on which PDL commands have DJDE counterparts.

Adding logical processing specifications

Logical processing commands are invoked when the system locates satisfactory test criteria. These test criteria are set up for record or block fields and, if met, allow special processing to take place for such things as banner pages, block selection or deletion, page selection from auxiliary paper trays, and page offsetting. An example of logical processing tests and criteria is provided in this catalog level command set:

```
/* Catalog level commands*/
TCCS:  CATALOG;
      T1: TABLE CONSTANT=( 'PAGE          1' );
      T2: TABLE CONSTANT=( 'INPUT RECORDS' );
      T3: TABLE CONSTANT=( 'JOB' );
      C1: CRITERIA CONSTANT=(122,11,EQ,T1),
          LINENUM=(1,10);
      C2: CRITERIA CONSTANT=(14,13,EQ,T2);
      C3: CRITERIA CONSTANT=(14,3,EQ,T3);
```

Specifying formats

There are many standard formats available for you to select for your JSL. These formats are listed in the PDE command section of chapter 7, "Specifying output parameters." PDEs, like VFUs and IDRs, require identification, for example:

```
PDE1:  PDE      BEGIN=(0.18 IN,0.66 IN),
          FONT=(LO112B,LO1BOA),
          PMODE=LANDSCAPE;
PDE2:  PDE      BEGIN=(1.03, .51),
          BEGIN=(6.30, .51),
          FONTS=P1O12A,
          PMODE=PORTRAIT;
```

Using copy modification entries

Copy modification entries (CMEs) allow you to change report output on static data and to change fonts within variable data. Sample CMEs follow:

```
CME1:  CME          LINE=( 9 , - ) , POS=1 , FONT=1 ;
CME2:  CME          LINE=3 , POS=59 ,
          CONSTANT=' FIRST QUARTER ' ;
```

Defining paper requirements

There are several ways of specifying paper stock in PDL. One method is the STOCKSET command, which can be referenced in lower-level commands. Each formatted page is then associated with the active STOCKSET command and the active FEED parameter of the OUTPUT command. If no FEED parameter is specified, the INIFEED parameter of the STOCKSET command takes effect. Here is a sample STOCKSET command:

```
SSML:  STOCKSET  ASSIGN=( ' WHITE ' , CVR ) ,
          ASSIGN=( ' YELLOW ' , BLL ) ,
          ASSIGN=( ' GREEN ' , SUM ) ,
          INIFEED=CVR ,
          SYSPAGE=SUM ;
```

The STOCKSET command calls out an INIFEED option and lets it override an OUTPUT command FEED parameter option.

The FEED parameter of the OUTPUT command can either specify a stock assigned in the STOCKSET command or bypass this referencing and specify that a stock be called out in the current STOCKSET.

Use the OUTPUT command PAPERSIZE and SYSPPR parameters to specify physical and system paper size and paper stock types. The OSTK and TRANS parameters allow you to specify the type of stock a job uses, and the NTO1 parameter enables you to instruct the printer to print the last page of a report first.

Specifying output requirements

There are many specifications you can select to define the manner and look of your printed application. You can also have messages displayed to operators to advise or remind them of special circumstances. This is done with the MESSAGE command:

```
MESSAGE  OTEXT=( 'ALL FORMS DUPLEX
                ONLY!!!',1,WAIT),
          ITEXT=( 'COPY 2 WILL NEED BLUE PAPER' );
```

Similarly, the ROUTE command sends printed information preceding the report to operators. Most output specifications are selected from parameters of the OUTPUT command. As with other commands, these can be specified at any command level, but are most often specified at the job level because of the many variations possible. Here are some examples:

```
/* Job level commands*/
```

```
JOB1:  JDE;
        OUTPUT  DUPLEX=YES,
                SHIFT=YES,
                FORM=SMPLE,
                MODIFY=CME1,
                FORMAT=PDE1,
                COPIES=10,
                BFORM=SMLBK,
                OFFSET=FIRST,
                PURGE=NO;
```

```
JOB2:  JDE;
        LINE    VFU=VFU2UP;
        OUTPUT  FORMAT=PDE4,
                FORMS=SPL2,
                COPIES=7;
```

The SEFFNT command specifies font mapping.

The EXPORT command specifies how reports will be segmented.

The DESTINATION parameter of the OUTPUT command specifies where printed output is delivered.

Ending a JSL

When you are finished constructing your JSL, you must let the system know you are finished by entering the END command plus a semicolon:

```
END;
```

Finished JSL

The following shows the JSL constructed in this section in its entirety:

```

XRXSPL:  JDL;

      /* ID level commands*/

VFU1:   VFU           ASSIGN=(1,11),
                        ASSIGN=(12,66),
                        TOF=11,BOF=66;

VFU2UP: VFU           ASSIGN=(1,(11,77)),
                        ASSIGN=(12,(66,132)),
                        TOF=11, BOF=132;

VFU3:   VFU           ASSIGN=(1,1),
                        TOF=1, BOF=132;

      /*System level commands*/
      VOLUME           HOST=IBMOS,
                        LABEL=STANDARD,
                        CODE=EBCDIC,
                        PLABEL=YES;

      BLOCK            LENGTH=2660,
                        PREAMBLE=4, LTHFLD=2,
                        FORMAT=BIN;

      RECORD           LENGTH=136,
                        PREAMBLE=4,
                        STRUCTURE=VB,
                        LTHFLD=2, OFFSET=0,
                        FORMAT=BIN;

      LINE             DATA=(1,132),
                        PCCTYPE=ANSI,
                        PCC=(0,NOTRAN),
                        VFU=VFU1;

      ACCT             USER=TRAY;

IDEN    IDEN           PREFIX='C9700', SKIP=7,
                        OFFSET=1;

PDE1:   PDE           BEGIN=(0.18 IN,0.66 IN),
                        FONT=(LO112B,LO1BOA),
                        PMODE=LANDSCAPE;

PDE2:   PDE           BEGIN=(1.03, .51),
                        BEGIN=(6.30, .51),
                        FONTS=P1O12A,
                        PMODE=PORTRAIT;

CME1:   CME           LINE=(9,-), POS=1, FONT=1;
CME2:   CME           LINE=3, POS=59,
                        CONSTANT='FIRST
                        QUARTER';

SSML:   STOCKSET     ASSIGN=('WHITE', CVR),
                        ASSIGN=('YELLOW', BLL),
                        ASSIGN=('GREEN', SUM),
                        INIFEED=CVR,
                        SYSPAGE=SUM;

MESSAGE OTEXT=('ALL FORMS DUPLEX
ONLY!!!', 1, WAIT),
ITEXT=('COPY 2 WILL NEED BLUE
PAPER');

      /* Job level commands*/
JOB1:   JDE;
      OUTPUT          DUPLEX=YES,

```

```
SHIFT=YES ,  
FORM=SMPLE ,  
MODIFY=CME1 ,  
FORMAT=PDE1 ,  
COPIES=10 ,  
BFORM=SMLBK ,  
OFFSET=FIRST ,  
PURGE=NO ;  
  
JOB2:  JDE ;  
        LINE  VFU=VFU2UP ;  
        OUTPUT FORMAT=PDE2 ,  
          FORMS=SPL2 ,  
          COPIES=7 ;  
  
END ;
```

Compiling the JSL

Compiling converts the information from a job source file (JSL) into an object file (JDL). This is the command to enter at the keyboard display user interface to compile the JSL:

PDL [*filename* [.JDL]] [,*parameters*]

The *file-id* is the source file that contains the PDL command. The *file-id* can be the file-name alone, or, more typically, the file-name and the file directory or extension, in the form *file-name.file-type*.

Table 3-1 lists the *filename* option(s) and definition(s).

Table 3-1. **PDL filename option(s) and definition(s)**

Option(s)	Definition(s)
<i>file-name</i>	Specifies the 1 to 6 character name of the JSL file to be input to the PDL task. If no file name is specified, source input for PDL is read from tape, and a JDL file is created on disk for each SYSTEM (or JDL) system command encountered.
<i>file-name.JSL</i>	The file-id which allows you to optionally include the file extension, or file-type, to the file name in the PDL command. If the file extension is used, it must be as shown as JSL. Any deviation aborts the command.

Parameters may be any combination of the parameters listed in table 3-2.

Table 3-2. **PDL filename parameter option(s) and definition(s)**

Option(s)	Definition(s)
NOPRINT	Specifies that only source records that contain errors, the diagnostics that apply to those lines, and the PDL summary reports are printed during compilation. If there are no errors, there is no printout. The default is to print all of the above plus the PDL source records.
NOSOURCE	Specifies that no source files (JSLs) be created when the input is from tape. This option has no effect when input is not from tape.
REPLACE	Specifies that an existing JDL file may be replaced by a new output file of the same name. This is the default.
NOREPLACE	Specifies not to create a new JDL file if it has the same name as an existing file. REPLACE is the default.
TRAY	Sends the compiled listing of your code and diagnostics to the sample tray. BIN is the default, which delivers the printout to the output tray.
NRWD	Specifies that no rewind occurs after a tape file is processed. The default is to rewind.
DISPLAY	Displays all PDL messages. The default prints the messages only on the PDL listing.
TD n	Tape device (assigned at SYSGEN time) that contains the JSL.

Examples:

```
PDL XEROX ,NOSOURCE ,NOPRINT
```

```
PDL H2SYS ,TRAY
```

XEROX and H2SYS are JSL names. Note that parameter keywords are not order dependent.

Printing the job

Once your JDL is compiled successfully, you can have the LPS print the actual application using the START command:

```
START jde-id,jdl-id
```

The START command specifies the job descriptor library (JDL) and the job descriptor entry (JDE), also called the JOB for printing. Refer to the “Non-JDL hierarchy—START command” section in the “Print description language components and processes” chapter in this manual or to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operator Guide* or *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*, for additional information on the START command.

Page considerations

Now that you have gone through many of the steps and considerations involved in creating a JSL, the following sections give you information on these page-related topics:

- Paper sizes
- Page orientations
- Registration shift and skew

Paper sizes

With the laser printing system, paper sizes are considered in terms of page frames, which are boundaries associated with a page as a unit of printing or imaging. Three page frames are defined in the system:

- Physical page
- System page
- Virtual page

In addition to page size, edgemarking and non-imaged elements must be considered when you design the pages of your applications.

System page

This refers to the maximum image area of the printer. The system page size varies, depending on the size of the paper you are running for your job.

Physical page

The physical page is the size of the paper itself. You may select any page dimensions within a 8- by 10-inch minimum and a 14.33- by 17-inch maximum.

Edgemarking

Edgemarking is the placement of marks along the edge of the page. These marks consist of graphic elements that bleed off the paper, tabs for section reference, or marks that denote changes made in redline drafts. Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for detailed information on edgemarking.

Non-imaged elements

Elements of a page, that is, text and graphics, may begin at the edge of the physical page and may even extend off the page. However, if any part of a printed element begins off the system page, then no part of the element is imaged.

- If a line of variable data begins off the system page, no part of the line is printed.
- If a ruled line begins off the system page, no part of the ruled line is printed.

- A ruled line near the edge of the system page must be positioned at least one half the line's thickness inside the system page to be printed. If positioned less than one half of the line's thickness, you will get an error message but the system will print the job. Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for additional information.

One common cause of print elements accidentally beginning off the system page is the improper use of the OUTPUT command SHIFT parameter. This command is used to shift the entire page's contents relative to the boundaries of the system page. When a negative shift value is entered (as is often the case for the back side of duplex pages), and that value exceeds the left margin, no text elements print. When using a negative value for the SHIFT parameter, be sure that it is less than the value of the left margin.

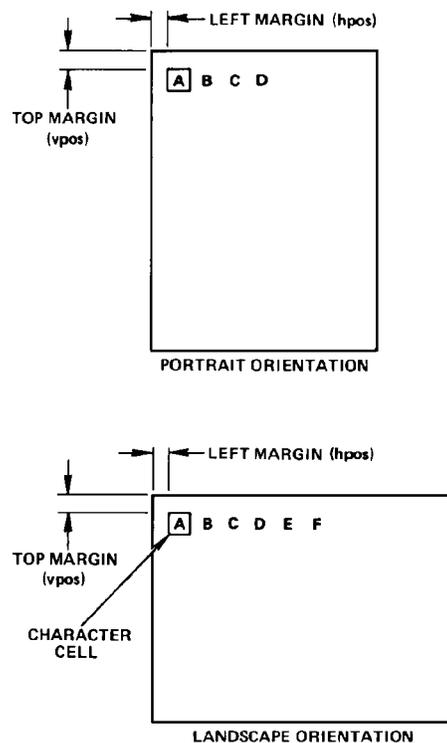
Page orientation

There are two types of page orientation:

- Landscape
- Portrait

The vertical and horizontal positions for each of these orientations is shown in figure 3-1.

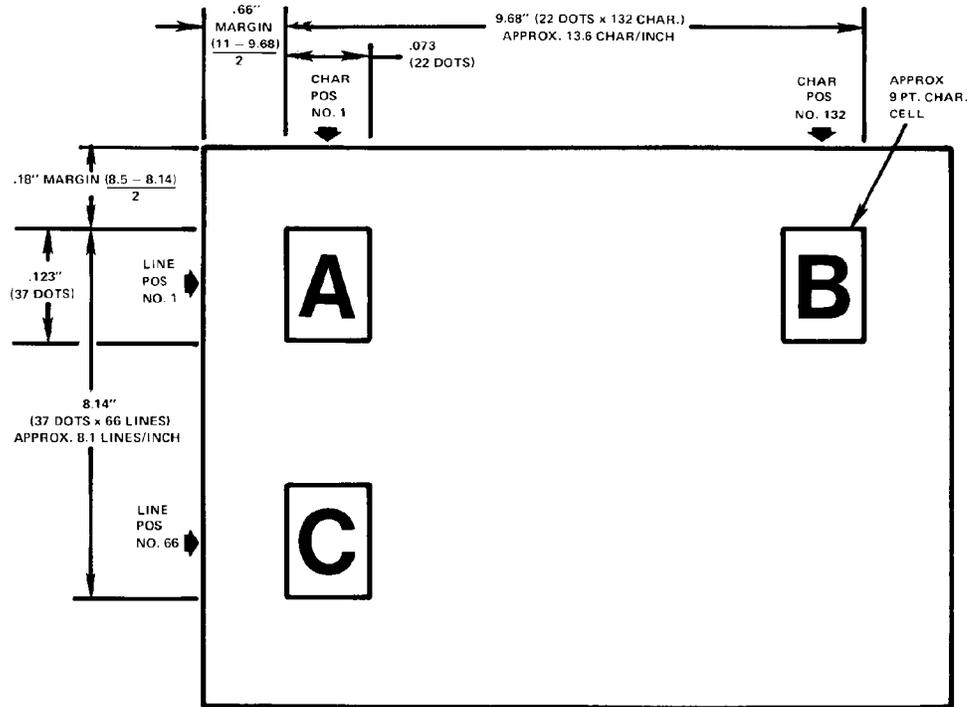
Figure 3-1. **Vertical and horizontal positions in landscape and portrait modes**



Landscape orientation

Figure 3-2 shows landscape orientation dimensions.

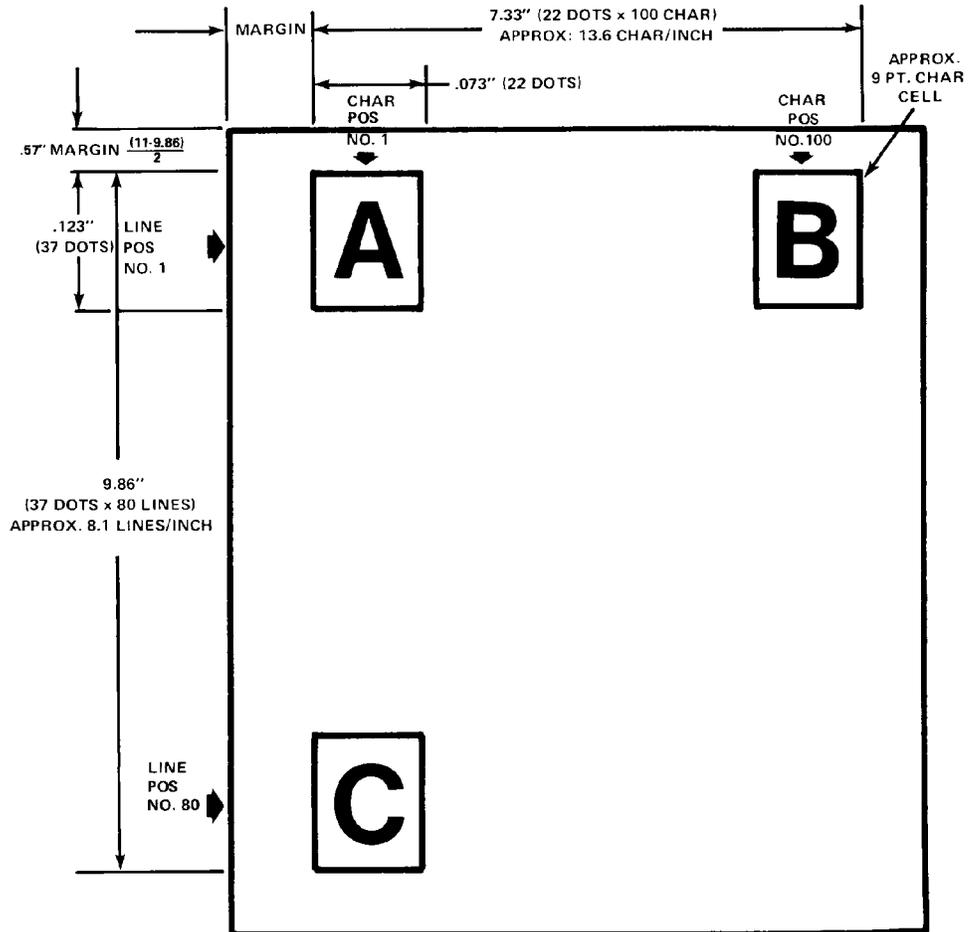
Figure 3-2. **FMT1: equivalent impact printer format, 6 lines per inch**



Portrait orientation

Figure 3-3 shows portrait orientation dimensions.

Figure 3-3. **FMT6: equivalent impact printer format, 6 lines per inch**



Registration shift and skew

The registration of a printed image can appear shifted or skewed on a page if the sheet of paper is misaligned as it enters the printer. Because of the design of the LPS feeder, the image registration on each page can vary slightly both horizontally and vertically by up to .05 inches/1 mm. The image can also be slanted or skewed slightly by up to .05 inches/1 mm in opposite directions, for a maximum skew of 0.1 inches/2 mm. Refer to figures 3-4 and 3-5.



Note: The following figures are the same specifications merely rotated to show portrait and landscape orientations. The shift and skew variances described here are within allowable specifications but, as this can affect the registration of variable data in preprinted forms and the placement of images close to the edge of the page, it is important to make allowance for this condition.

Figure 3-4. Landscape orientation shift and skew

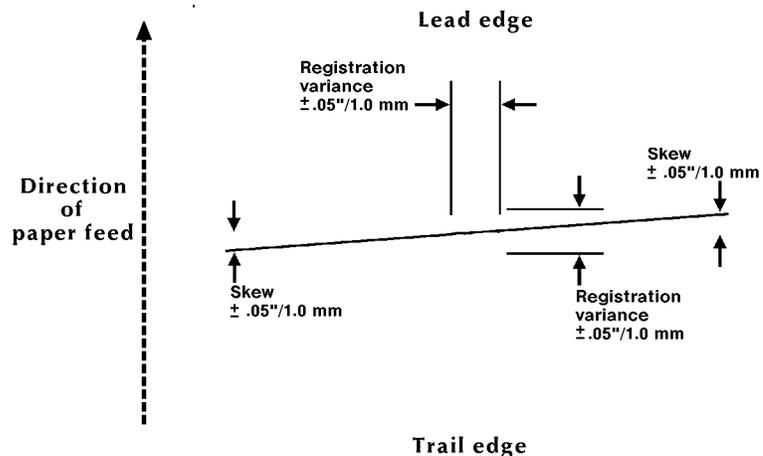
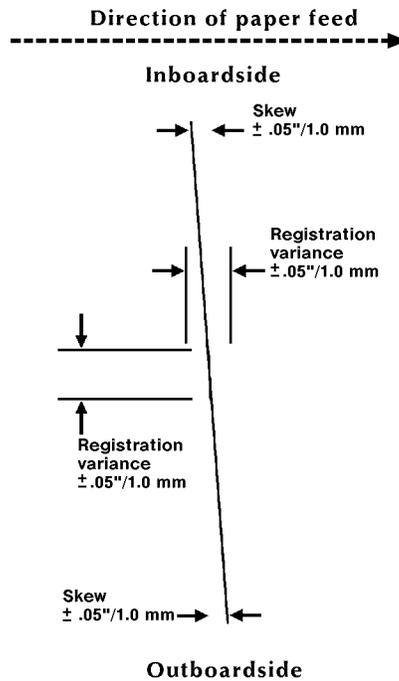
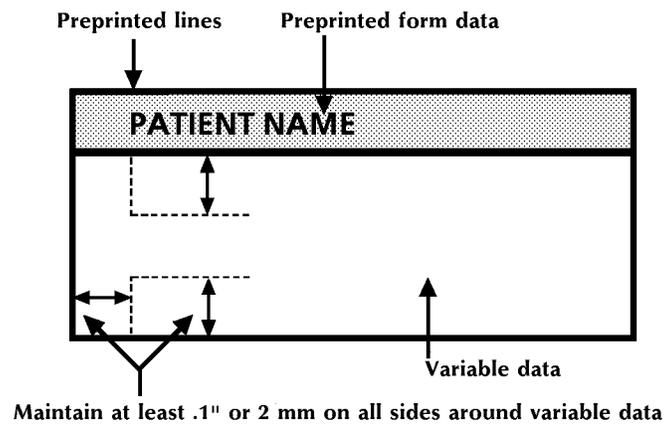


Figure 3-5. Portrait orientation shift and skew



For best results, when designing preprinted forms, allow approximately .1 inch or 2.0 mm space on all sides of any boxes, or above and below any lines onto which variable data is to be printed, as illustrated in figure 3-6.

Figure 3-6. Maintaining margins in preprinted boxes to allow for LPS registration and skew variations

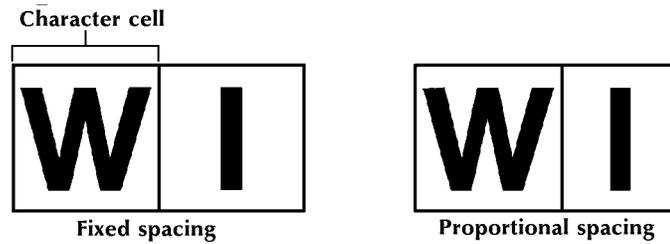


Note: Enlarged for the purpose of illustration; not to scale.

Fonts

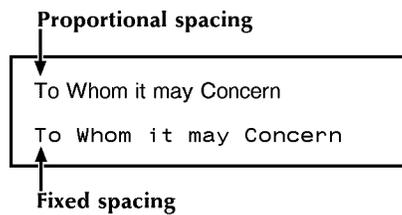
A font is a character set which has a unique type style, type size, and orientation. Both fixed and proportionally spaced fonts are available for use on an LPS. Each font character occupies an area called a character cell. All character cells in a fixed font are the same width. Character cells in a proportional font vary in width. Refer to figure 3-7.

Figure 3-7. **Character spacing**



Because the length of a line printed with a proportional font is relatively unpredictable, fixed fonts are used for variable data on a report to avoid overprinting of forms by variable data. Proportional fonts are normally used for forms data, such as titles and headings. A business letter is an example of the use of proportional fonts for variable data. An example of the difference in line length is illustrated in figure 3-8.

Figure 3-8. **Character spacing examples**



Fonts are available in various families (for example, OCR and Titan), sizes, and faces medium and bold. Refer to the Xerox 4000 Family Laser Printing Systems Font User Guide for more information on fonts and for samples of the font families, sizes, and faces available for use with your Xerox DocuPrint 96/DocuPrint 96MX LPS.

In addition to typeface, style, and size, a font can be defined by its orientation:

- Landscape
- Portrait
- Inverse landscape
- Inverse portrait

Refer to the *Xerox 4000 Family Laser Printing Systems Font User Guide* for specific font information, the *Xerox DocuPrint 96/DocuPrint 96MX LPS Forms Creation Guide* for using fonts in a form, and the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for using font editor keyword commands (used to create source font files from existing licensed and non-licensed font files).

Custom fonts, signatures, and logos may be ordered from Xerox through your Xerox sales representative.

Font and graphic memory

The LPS can print up to 128 fonts on a single page. When processing the page data, the controller stores font, as well as graphic information in a special memory cache within the AIS. The amount of memory required to store font data depends on the size of the fonts and the number of different fonts on a single page.

If your applications call for a complex mix of fonts and graphics, the increased font and graphic memory option can greatly improve the processing time required to print these documents.

Custom fonts, logos, and signature font data also consume font/graphic memory during processing.

Xerox DocuPrint 96/DocuPrint 96MX compatibility with the 4850, 4135, 4635 and 4050/4090/4650 LPS

Your Xerox DocuPrint 96/DocuPrint 96MX LPS allows you to process Xerox 4850 HighLight Color LPS print jobs which contain ink references.

The Xerox DocuPrint 96/DocuPrint 96MX LPS is also fully compatible with the 4135 and 4635 LPS with respect to JSL commands.

The Xerox DocuPrint 96/DocuPrint 96MX LPS prints all 4050, 4090, 4650, and 4635 LPS applications successfully.

The functions of the various laser printing systems are listed in table 3-3 and explained in the sections that follow.

Table 3-3. **Software version compatibility**

JDL SOURCE Downloaded to a...	V3.5 4050 4090 4650	V3.6 Rel. 1.5 4135	V3.7 4850	V3.8 4050 4090 4650	V3.9 9790	V4.0/V5.0 4850/4890	V3A 4635	DP 96/ DP96MX
Compiled on a V3.5	As Expected	As Expected	As Expected	As Expected	As Expected	As Expected	As Expected	As Expected
Compiled on a V3.6 Rel. 1.5	1	As Expected	1	2 4	1	2 4	As Expected	As Expected
Compiled on a V3.7	1	3	As Expected	3	1	As Expected	3	3
Compiled on a V3.8	1	3	5	As Expected	1	As Expected	3	3
Compiled on a V3.9	1	As Expected	1	1	As Expected	1	As Expected	As Expected
Compiled on a V4.0/V5.0	1	3 6 7	5 6 7	3 6 7	1	As Expected	7	7
Compiled on a V3A	1	1	1	1	1	1	As Expected	As Expected
Compiled on a V3B	1	1	1	1	1	1	As Expected	As Expected

1. Commands unique to this software release are not supported. Unpredictable results will occur.
2. Commands unique to this software release are not supported. However, an error message is displayed, unique commands are ignored, and the job will print.
3. Job prints in black only. Two color graphics are not supported.
4. Paper size not supported by the printer. Will cause job to abort.
5. RFEED command is not supported.
6. LOGO command in JSL is not supported.
7. Extended CRITERIA and VCODE are not supported.

Xerox 4850 and 4890 HighLight Color LPS

This section explains how the Xerox DocuPrint 96/DocuPrint 96MX LPS runs applications created for the 4850 and 4890 HighLight Color Laser Printing Systems.

Running 4850/4890 applications on your Xerox DocuPrint 96/DocuPrint 96MX LPS

If you have a Xerox 4850 or 4890 version 4.0 HighLight Color LPS, your applications probably include the use of blue, red, or green highlight color. The JDL used to print these applications contain commands and parameters which specify the location and the color ink to be used. Your Xerox DocuPrint 96/DocuPrint 96MX LPS will compile, process, and print these jobs, ignoring the ink specified and using the default black ink.

The following PDL commands, parameters, and DJDEs allow ink specifications:

PDL commands and parameters

PDL commands and parameters include the following:

- ABNORMAL command
 - IMISMATCH parameter
 - ISUBSTITUTE parameter
- CME command

This command consists of the INK parameter.
- IDR command
 - ICATALOG parameter
 - ILIST parameter
 - PALETTE parameter
- LINE command

This command consists of the INKINDEX parameter.
- OUTPUT command
 - BFORM parameter
 - CYCLEFORMS parameter
 - FORMS parameter
 - IDFAULT parameter
 - IDR parameter
 - IMAGE parameter
 - IRESULT parameter
 - NUMBER parameter
 - XMP parameter
- VOLUME command

This command consists of the INTERPRESS parameter.

DJDE commands DJDE commands include the following:

- ALTER
- BFORM
- FORMS
- GRAPHIC
- ICATALOG
- IDFAULT
- IDR
- ILIST
- IMAGE
- INKINDEX
- IRESULT
- NUMBER
- PALETTE
- XMP

For detailed information on the syntax and use of these PDL commands and DJDEs, refer to your *Xerox 4850/4890 HighLight Color LPS PDL/DJDE Reference*.

Downloading 4850/4890 LPS applications to your Xerox DocuPrint 96/DocuPrint 96MX LPS

You can also download color JDLs to your Xerox DocuPrint 96/DocuPrint 96MX LPS. When compiled on your Xerox 4850/4890, these JDLs can then be run on either your Xerox DocuPrint 96/DocuPrint 96MX or 4850/4890 LPS. Any ink specifications you have made in the JSL are printed in the default black ink on the Xerox DocuPrint 96/DocuPrint 96MX; however, they print in the desired inks on your 4850/4890. Logos are not supported on a 4890 or Xerox DocuPrint 96/DocuPrint 96MX LPS. Refer to your Xerox 4850/4890 HighLight Color LPS PDL/DJDE Reference.

The only exception to this is the RFEED command, which is supported on the Xerox DocuPrint 96/DocuPrint 96MX LPS, but not on your 4850.

4850/4890 HighLight Color LPS forms

The Xerox DocuPrint 96/DocuPrint 96MX LPS makes a distinction between color and monochrome forms. It is important to note that forms do not have to contain color-unique forms source library (FSL) commands in order to be classified as color forms. These forms may be generated by either compiling the FSL using the 4850/4890 LPS forms description language (FDL) compiler, converting the monochrome form to color with the File Conversion Utility (FCU) resident on the 4850/4890 LPS, or downloading color forms created from host or third party vendor software packages.

In addition, the Xerox DocuPrint 96/DocuPrint 96MX LPS makes a distinction between color and monochrome logos. Due to differences in file formats, the Xerox DocuPrint 96/DocuPrint 96MX LPS restricts you from specifying a color form that references a monochrome logo. The opposite is also true. That is, a monochrome form may not

reference a color logo. Unlike logos, any form may reference either color or monochrome images.

The Xerox DocuPrint 96/DocuPrint 96MX LPS FDL compiler cannot compile an FSL containing color-unique commands. However, this does not imply that the Xerox DocuPrint 96/DocuPrint 96MX LPS will not print them. If an FSL were precompiled by a 4850/4890 LPS FDL compiler and its file (FRM) downloaded to a Xerox DocuPrint 96/DocuPrint 96MX LPS, the form will print.

Refer to your *Xerox 4850/4890 HighLight Color LPS Forms Creation Guide* for detailed information on creating and using highlight color forms.

Points to note

Note the following:

- Light tints with isolated pixels of color that print on the Xerox 4850/4890 may not print on the Xerox DocuPrint 96/DocuPrint 96MX LPS.
- Color text printed over solid black background or black text printed over solid color backgrounds on the Xerox DocuPrint 96/DocuPrint 96MX LPS may not be visible. Be especially careful in using this format because this situation does not generate displayed or printed messages.
- Likewise, color text printed over gray backgrounds or black text printed over color shaded backgrounds on the Xerox DocuPrint 96/DocuPrint 96MX printer may not be clearly visible. Be especially careful in using this format because this situation does not generate displayed or printed messages.
- 600 spots per inch (spi) tints and shades printed on the Xerox DocuPrint 96/DocuPrint 96MX are finer and more uniform than 300 spi tints and shades printed on the 4850/4890 LPS.

Xerox 4050/4090/4650/9700F LPS

This section explains how the Xerox DocuPrint 96/DocuPrint 96MX LPS runs applications created for the 4050/4090/4650 Laser Printing Systems.

Creating Xerox DocuPrint 96/DocuPrint 96MX LPS applications on a 4050, 4090, 4650, or 9700F LPS

As with any application, you can create Xerox DocuPrint 96/DocuPrint 96MX JSLs on your 4050, 4090, 4650, or 9700F LPS and then compile them on your Xerox DocuPrint 96/DocuPrint 96MX LPS for printing.

Running 4050, 4090, 4650, and 9700F jobs on your Xerox DocuPrint 96/DocuPrint 96MX LPS

The Xerox DocuPrint 96/DocuPrint 96MX LPS offers full backward compatibility with the Xerox 4050, 4090, 4650, and 9700F laser printing systems. In other words, all standard JDLs produced on these printers will run successfully on the Xerox DocuPrint 96/DocuPrint 96MX LPS. However, it is advisable to recompile them.

4. Specifying input parameters

Input data is processed and temporarily written to disk for subsequent printing under control of user-selected PDL commands. The input processor decodes and formats input data from an offline magnetic tape, a host-attached channel interface, a remote communication, or Ethernet interface. The general functions of input processing are described below.

The basic PDL commands available to control input processing are BLOCK, CODE, PCC, RECORD, SEFFNT, TCODE, and VOLUME, which are defined in this chapter. The chapter "Using logical processing commands" defines commands enabling you to specify logical functions that may be performed on either a record, block, or page basis.

Table 4-1 summarizes the input PDL commands.

Table 4-1. **Summary of commands associated with input processing**

Command	Function
BLOCK	Input data block characteristics
CODE	Input code translation table
PCC	Defines printer carriage control code table
RECORD	Input data record characteristics
SEFFNT	Defines font mapping for short-edge feed jobs
TCODE	Marked comparison type assignments
VOLUME	Input medium characteristics

BLOCK command

Table 4-2 summarizes the BLOCK command parameters.

Table 4-2. **Summary of BLOCK command parameters**

Parameter	Specifies	Offline	Online	DJDE
ADJUST	Block length adjustment value	Y	N	N
CONSTANT	Block termination code	Y	N	N
FORMAT	Length field recording mode	Y	N	N
LENGTH	Maximum block size	Y	N	N
LMULT	Multiplication factor to determine block length	Y	N	N
LTHFLD	Length of field containing the block length	Y	N	N
OFFSET	Location of the block length field	Y	N	N
POSTAMBLE	Length of extraneous data at end of block	Y	N	N
PREAMBLE	Length of operating system portion of block	Y	N	N
ZERO	End of block criteria test	Y	N	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ADJUST

This parameter specifies a block adjustment value which is added to or subtracted from the contents of the block length field to determine the true block length.

Syntax BLOCK ADJUST = *option(s)*

Options Table 4-3 lists the parameter option(s) and definition(s).

Table 4-3. **ADJUST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the block adjustment length. This length is a constant integer added to or subtracted from the value in the block length field of every tape block. The resulting value is the true block length. The range for a value is -127 to 127 and must be less than the block length parameter (LENGTH). The character plus (+) or minus (-) may be used to specify a positive or negative adjustment. The default is 0.

CONSTANT

This parameter specifies that the block delimiter constant *sc* and all data following it are ignored until the end of the block is reached.

Syntax BLOCK CONSTANT = *option(s)*

Options Table 4-4 lists the parameter option(s) and definition(s).

Table 4-4. **CONSTANT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>sc</i>	A string (hexadecimal, octal, or character) constant as described in the chapter "Creating a job source library (JSL)." The length of the constant may be from one to four bytes. There is no default.

FORMAT

This parameter specifies the recording mode of the block length field.

Syntax BLOCK FORMAT = *option(s)*

Options Table 4-5 lists the parameter option(s) and definition(s).

Table 4-5. **FORMAT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>type</i>	Available type parameters are: <ul style="list-style-type: none"> • BIN (binary) • DEC (decimal) • PACK (packed with no sign) • PKSG (packed with sign). The default is BIN.

LENGTH

This parameter specifies the longest physical block being processed.

Syntax BLOCK LENGTH = *option(s)*

Options Table 4-6 lists the parameter option(s) and definition(s).

Table 4-6. **LENGTH parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the length, in bytes, of the longest physical block (an integer in the range 12 to 24,576). The default is 1330. The maximum block size that may be processed (up to 24,576 bytes) is dependent upon the available task memory and the processing features being invoked. Refer to "Points to note—BLOCK command" later in this chapter for further details. For offline processing, the tape label contents may override a coded LENGTH parameter, but this length is still limited by the above maximum value. The default is 1330.

LMULT

This parameter specifies a multiplication factor being applied to the contents of the block length field to determine the true block length.

Syntax BLOCK LMULT = *option(s)*

Options Table 4-7 lists the parameter option(s) and definition(s).

Table 4-7. **LMULT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Multiplied by the value in the length field (refer to LENGTH parameter) to compute the number of bytes in the block. A value is an integer in the range of 1 (the default) to 15. The default is 1.

LTHFLD

This parameter specifies the length of the field containing the block length.

Syntax BLOCK LTHFLD = *option(s)*

Options Table 4-8 lists the parameter option(s) and definition(s).

Table 4-8. **LTHFLD parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the length in bytes of the field containing the block LENGTH specified above. The size is an integer in the range of 0 (the default) to 5. If size is set to 0, the block length field is not considered to be part of the block, and the length of a block, on tape, is the actual block length. The default is 0.

OFFSET

This parameter specifies the location of the block length field.

Syntax BLOCK OFFSET = *option(s)*

Options Table 4-9 lists the parameter option(s) and definition(s).

Table 4-9. **OFFSET parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the block length field offset. This offset is the number of bytes from the first byte of a block to the block length field. A value is an integer in the range 0 (the default) to LENGTH-LTHFLD-1. The default is 0.

POSTAMBLE

This parameter specifies the length in bytes of the extraneous data at the end of each tape block; that is, it is an offset from the end of a block backwards to the end of the last logical record.

Syntax BLOCK POSTAMBLE = *option(s)*

Options Table 4-10 lists the parameter option(s) and definition(s).

Table 4-10. **POSTAMBLE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	An integer in the range of 0 (the default) to block length. The default is 0.

PREAMBLE

This parameter specifies the length of the operating system portion of the block, that is, the byte offset from the first byte of a tape block to the first byte of the first logical record.

Syntax BLOCK PREAMBLE = *option(s)*

Options Table 4-11 lists the parameter option(s) and definition(s).

Table 4-11. **PREAMBLE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	An integer in the range of 0 (the default) to block length. The default is 0.

ZERO

This parameter specifies the end of block indicator.

Syntax BLOCK ZERO = *option(s)*

Options Table 4-12 lists the parameter option(s) and definition(s).

Table 4-12. **ZERO parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that the end of a tape block is indicated by a value of 0 in the record length field (before applying the record length adjustment). Data that follows the record is ignored up through the end of the block.
NO	Indicates that the end of a tape block is not indicated by a value of 0 in the record length field. The default is NO.

Points to note

Note the following when using the BLOCK command:

- The LENGTH parameter may be overridden by ANSI, IBM OS/Standard, or Honeywell 2000 COBOL labels that specify block length.
- The values for LTHFLD, OFFSET, FORMAT, and PREAMBLE may be overridden if RECORD STRUCTURE is changed as the result of ANSI, IBM OS/Standard, or Honeywell 2000 COBOL label processing.
- The length on a 4-by-3 packed format tape or Honeywell 600 is the number of 6-bit bytes or characters in the tape block.
- The length of the block delimiter constant should not be coded as the BLOCK POSTAMBLE. Both lengths are subtracted from the end of the block.
- The search for the block delimiter constant starts after the block preamble and proceeds forward to the first appearance of the constant.
- The maximum block size that may be processed by the input task is 24,576 bytes. The input task is able to allocate at least one input buffer for offline tape jobs when the tapes are written in maximum size blocks.
- If a block length is specified which is less than this minimum block length found on the tape, input processing allocates input buffers, which are sized to the minimum tape block length. It is wise to specify the maximum block length in the JSL, so that input buffers are large enough to handle the largest block. However, this may slow performance. If the actual tape block length is smaller than the JSL block length, then no error message is reported; otherwise, an error is displayed.
- If a block delimiter constant is positioned and is part of a record or block, the user should use caution, since the record or block

will be truncated. As a result, the data will not be formatted as specified.

Online versus offline JDLs

An offline JDL statement set that modifies the system default values, specifically in the BLOCK or RECORD commands, may experience incorrect results if running an online job (JDE). In other words, if an online job (JDE) is called out in an offline JDL that has changed the system default values (of the BLOCK and RECORD command parameters), the job may not print correctly. It is best then to separate and run online and offline jobs (JDEs) independently from each other.

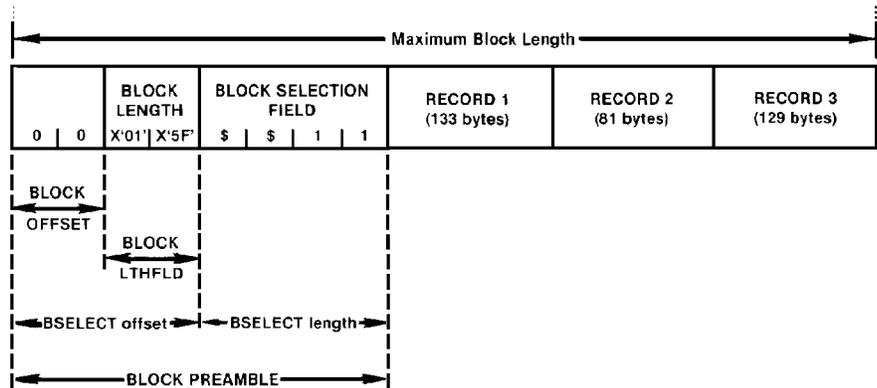
Example

The BLOCK command for the example below would be coded as follows:

```
BLOCK LENGTH=351, OFFSET=2, ADJUST=0,
      LTHFLD=2, PREAMBLE=8, FORMAT=BIN;
```

Figure 4-1 shows sample values for BLOCK command parameters.

Figure 4-1. **Sample BLOCK command**



CODE command

A CODE command is referenced by the CODE or LCODE parameters of the VOLUME command whenever a user-defined code translation table is required.

An identifier label (ac:) is optional for the first CODE command within a JDE. Thereafter, each additional CODE command within the JDE must be labeled. Therefore, when multiple CODE commands are available within a JDE or JDL, only one can be unlabeled. The syntax is ac:CODE.

Table 4-13 summarizes the use of the CODE command parameters.

Table 4-13. **Summary of BLOCK command parameters**

Parameter	Specifies	Offline	Online	DJDE
ASSIGN	User-defined code assignments	Y	Y	N
DEFAULT	Standard character translation table	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ASSIGN

This parameter defines user code assignment exceptions, or the entire user character translation table.

Syntax CODE ASSIGN = *option(s)*

Options Table 4-14 lists the parameter option(s) and definition(s).

Table 4-14. **ASSIGN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>input,(output₁ [,output₂],[...])</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> <i>input</i> Defines the input code. <i>output</i> Defines the output code, such as hexadecimal or decimal, that corresponds to the input code.

DEFAULT

This parameter specifies a base code from which code assignment exceptions can be made. The base code is specified by a code-type; the exceptions are specified using the ASSIGN parameter.

Syntax CODE DEFAULT = *option(s)*

Options Table 4-15 lists the parameter option(s) and definition(s).

Table 4-15. **DEFAULT** parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>code-type</i>	<p>This option can be one the following:</p> <ul style="list-style-type: none"> • EBCDIC • ASCII • BCD • PEBCDIC • H2BCD • H6BCD • IBMBCD • <i>value</i> <p>A <i>value</i> is a one-byte hexadecimal, octal, or alpha constant. If a value is specified, all inputs are coded to the specified <i>value</i>.</p> <p>The DEFAULT parameter must be coded prior to any ASSIGN parameter for the assignment exceptions to come into effect. A DEFAULT parameter following any defined corresponding ASSIGN input or output parameter options causes this correspondence to be replaced by the DEFAULT parameter.</p> <p>The default is EBCDIC.</p>

Point to note

Multiple user-defined code translation tables are allowed, but only one may be without a command identifier. The corresponding CODE and LCODE parameters of the VOLUME command are used to reference each user-created code table by a command identifier. The USER option of the CODE and LCODE parameters of the VOLUME command are used to reference a user-defined code translation table in which no command identifier is coded.

Examples

Following are two examples of how to use the command.

Example 1 A user's input tape is recorded in EBCDIC. On output, however, codes 5A, 5B and 5C (characters !, \$, *) respectively), are assigned to the character blank (X'40'). The command to modify the EBCDIC base table is as follows:

```
CODE DEFAULT=EBCDIC,ASSIGN=(X'5A',X'40'),  
ASSIGN=(X'5B',X'40'),ASSIGN=(X'5C',X'40');
```

Example 2 CODE DEFAULT=EBCDIC,ASSIGN=(X'5A',(X'40',X'40',X'40'));
In example 2, consecutive input codes, such as X'5B ' and X'5C', need not be specified to accomplish code modification.

PCC command

The PCC command enables the user to create a table (or modify an existing standard table) of one-byte printer carriage control codes and define their action. Line spacing, skip-to-channel, and printing actions are all defined through this parameter.

The user specifies an identifier (of the type *ac*) when defining the PCC table and references it in the *PCCTYPE* parameter of the *LINE* command.

Table 4-16 summarizes the use of the PCC command parameters.

Table 4-16. **Summary of PCC command parameters**

Parameter	Specifies	Offline	Online	DJDE
ADVTAPE	Printer action for two successive channel skips	Y	N	N
ASSIGN	User carriage control code exceptions	Y	N	N
DEFAULT	Standard printer carriage control table	Y	N	N
INITIAL	Initial reference point for first carriage control	Y	N	N
MASK	Mask of bits from carriage control byte	Y	N	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ADVTAPE

This parameter specifies if an advance to a new page is to occur when two successive channel-skip parameters are issued with no intervening print. For example, on most printers, the actions: *PSK1* (print-and-skip-to-channel-1) followed by *SK1N* (skip-to-channel-1, do not print) would cause a blank page to be output. However, on a 1403 printer, these actions would not cause a blank page to be output.

Syntax *ac:PCC ADVTAPE = option(s)*

Options Table 4-17 lists the parameter option(s) and definition(s).

Table 4-17. **ADVTAPE parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that multiple skips are honored. The default is YES.
NO	Specifies that multiple skips result in only one skip action being taken. Note that <i>SK1P</i> (skip-to-channel-1-and-print) followed by a second <i>SK1P</i> results in a page transition since printing occurred on the first page (even if blanks were printed).

ASSIGN

This parameter specifies printer carriage control exceptions to a default table (refer to DEFAULT parameter).

Syntax *ac:PCC ASSIGN = option(s)*

Options Table 4-18 lists the parameter option(s) and definition(s).

Table 4-18. **ASSIGN** parameter option(s) and definition(s)

Option(s)	Definition(s)
(byte, (ccln ₁ , [ccln ₂], ...))	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>byte</i> The printer carriage control byte being defined. Its value is in the range 0 to 255 (X00 to XFF). • <i>ccln</i> Specifies the action that should be taken when the printer carriage control byte defined in <i>byte</i> is encountered. A <i>ccln</i> may be any of the following definitions: <ul style="list-style-type: none"> — TOF When <i>byte</i> causes the bottom-of-form (BOF) to occur, instructs the printing system to go to the top-of-form (TOF) on the next page and to stop spacing. — IGN When <i>byte</i> causes the bottom-of-form (BOF) to occur, instructs the printing system to ignore it to continue spacing through the end of the physical page. The page then transitions to top-of-form and spacing continues. — OVR When <i>byte</i> causes the bottom-of-form (BOF) to occur, instructs the printing system to go to the top-of-form (TOF) on the next page and to continue spacing. The default is OVR. • Space print parameters This has the following components: <ul style="list-style-type: none"> — <i>field₁</i> Specifies the carriage action to take before printing. An example is the ANSI carriage control: <ul style="list-style-type: none"> – SP<i>m</i> Space <i>m</i> lines before printing. – SK<i>n</i> Skip-to-channel-<i>n</i> before printing. — <i>field₂</i> Specifies whether printing occurs for this record: <ul style="list-style-type: none"> – P Print the output data at the line number computed after <i>field₁</i> is processed. – N No printing occurs for this record (default). — <i>field₃</i> Specifies the carriage action to take after printing. An example is the IBM1403 carriage control). <ul style="list-style-type: none"> – SP<i>m</i> Space <i>m</i> lines before printing. – SK<i>n</i> Skip-to-channel-<i>n</i> before printing. <p>Each of the three fields is optional; however, you must specify at least one field. The range for the variables <i>m</i> and <i>n</i> is 0 to 15.</p>

Considerations You do not have to specify consecutive byte values. For example, you can code the following command as a single command:

```
ASSIGN=(X'60',SP1), ASSIGN=(X'61',SP2),  
ASSIGN=(X'62',SP3);
```

The single command is:

```
ASSIGN=(X'60',SP1,SP2,SP3);
```

If, when the LINE command byte translation is specified, the PCC control byte is translated into LPS-standard EBCDIC before being applied. You must therefore specify the byte option of the ASSIGN parameter as an EBCDIC translation of the PCC byte. An example is X'F1' or '1'.

DEFAULT

This parameter allows selection of a set of printer carriage control codes. A table may be selected and then specific control codes may be modified with the ASSIGN parameters.

Syntax *ac:PCC DEFAULT = option(s)*

Options Table 4-19 lists the parameter option(s) and definition(s).

Table 4-19. **DEFAULT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>ccln</i>	Specifies the required action when a code has not been specifically assigned. The assignment codes for various actions are described in the <i>ccln</i> option description of the ASSIGN parameter. The system default PCC tables are defined in LPS-standard EBCDIC.
<i>pcctype</i>	Parameters and the tables that can be modified are: <ul style="list-style-type: none"> • ANSI • B2500 • B2700 • B3500 • B3700 • B4700 • B6700 • H2000 • H6000 • IBM1401 • IBM1403 • US70 • XEROX • NONE <p>The default is a table of PSP1 codes (print-and-space-1).</p>

Considerations The DEFAULT parameter must precede any ASSIGN parameter. Any preceding ASSIGN parameter is not incorporated into the PCC table.

INITIAL

This parameter specifies the initial reference point (TOF or BOF) from which a report performs its first carriage control function.

Syntax *ac:PCC INITIAL = option(s)*

Options Table 4-20 lists the parameter option(s) and definition(s).

Table 4-20. **INITIAL parameter option(s) and definition(s)**

Option(s)	Definition(s)
TOF	<p>Specifies the control program to perform the first spacing, skipping, or printing action from the top of form.</p> <p>The default is TOF.</p> <p>The DEFAULT parameter must precede any ASSIGN parameter. Any preceding ASSIGN parameter is not incorporated into the PCC table.</p>
BOF	<p>Specifies the control program to perform the first spacing, skipping, or printing action from the bottom of form.</p>

MASK

This parameter specifies

Syntax *ac:PCC MASK = option(s)*

Options Table 4-21 lists the parameter option(s) and definition(s).

Table 4-21. **Parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies an 8-bit value to be ANDed with the printer carriage control byte being processed. ANDing occurs after translation, if any (refer to "Points to note—PCC command," item 4). The result of this process is to mask off bits from the carriage control byte code which are not relevant to the operation being specified.</p> <p>The default is X'FF'.</p>

Points to note

Note the following when using the PCC command:

- Multiple user-defined PCC tables are allowed, but only one may be used without a command identifier. The corresponding PCCTYPE parameter on the LINE command references each table through a command identifier. The keyword USER can be used to reference any user-defined PCC table for which no command identifier is coded.
- The user may end a PCC command with a semicolon and start another PCC command to continue specification of the carriage control codes. Multiple PCC commands may be used within a single PCC table definition as long as there are no intervening non-PCC commands.
- Consecutive byte values need not be specified. Thus, the commands

```
ASSIGN=(X'60',SP1),ASSIGN=(X'61',SP2),ASSIGN=(X'62',SP3);
```

can be coded in the single command

```
ASSIGN=(X'60',(SP1,SP2,SP3));
```

- If under the LINE command byte translation is specified (that is, the LINE command PCC parameter TRAN option is specified), the PCC control byte is translated into LPS standard EBCDIC before being applied. This means that the byte option on the ASSIGN parameter must be specified as an EBCDIC translation of the PCC byte, for example, X'F1' or 1.

Example

```
PCC1: PCC DEFAULT=IBM1403, INITIAL=TOF, ADVTAPE=NO,  
ASSIGN=(X'40',SP1P), ASSIGN=(X'F1',SK1P),  
ASSIGN=(X'F8',PSK8);
```

Added to the standard IBM1403 carriage control code table are: X'40' (space 1 line and print), X'F1' (skip-to-channel-1-and print), and X'F8' (print-and-skip to channel 8).

RECORD command

Table 4-22 summarizes the use of the RECORD command parameters.

Table 4-22. **Summary of RECORD command parameters**

Parameter	Specifies	Offline	Online	DJDE
ADJUST	Record length adjust value	Y	N	N
CONSTANT	Record termination code	Y	N	N
FORMAT	Length field recording mode	Y	N	N
LENGTH	Maximum logical record length	Y	Y	N
LMULT	Multiplication factor to determine record length	Y	N	N
LTHFLD	Length of field containing record length	Y	N	N
OFFSET	Location of record length field	Y	N	N
POSTAMBLE	Length of extraneous data at end of record	Y	N	N
STRUCTURE	Input record structure	Y	N	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ADJUST

This parameter specifies an adjustment value added to or subtracted from the contents of the record length field to determine the true record length.

Syntax RECORD ADJUST = *option(s)*

Options Table 4-23 lists the parameter option(s) and definition(s).

Table 4-23. **ADJUST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the record adjustment length. It is a constant integer added to or subtracted from the value in the length field of every record. The range of a value is -127 to 127 and must be no greater than length. The first character may be plus (+) or minus (-). The default is 0.

CONSTANT

This parameter specifies a constant string used to signal the end of a record. This record delimiter constant string signals the end of the record, but it is not included in the print line.

Syntax RECORD CONSTANT = *option(s)*

Options Table 4-24 lists the parameter option(s) and definition(s).

Table 4-24. **CONSTANT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>sc</i>	Specifies a string (hexadecimal, octal, or alpha) constant as described in the chapter "Creating a job source library (JSL)." The length of the constant may be from 1 to 4 bytes. There is no default.

FORMAT

This parameter specifies the format of the record length field.

Syntax RECORD FORMAT = *option(s)*

Options Table 4-25 lists the parameter option(s) and definition(s).

Table 4-25. **FORMAT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>type</i>	Available type parameters are: <ul style="list-style-type: none"> • BIN (binary) • DEC (decimal) • PACK (packed with no sign) • PKSG (packed with sign). The default is BIN.

LENGTH

This parameter specifies the length of the longest logical record. A value specifies the length, in bytes, of the longest logical record.

Syntax RECORD LENGTH = *option(s)*

Options Table 4-26 lists the parameter option(s) and definition(s).

Table 4-26. **LENGTH parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	An integer in the range 1 to 310 for offline (133 is the default) or 150, the online default. For offline systems, the tape label contents may override a value. For online systems, if you define a record length larger than the default block length (1330 bytes), you must also define a block length that is big enough to hold the record. If a larger block length is not specified, the default length value is used and the system produces a warning message on the system controller display. The default is 133 offline, and 150 online.

LMULT

This parameter specifies a multiplication factor applied to the contents of the record length field to determine the true record length.

Syntax RECORD LMULT = *option(s)*

Options Table 4-27 lists the parameter option(s) and definition(s).

Table 4-27. **LMULT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the multiplication factor. The value specified is multiplied by the value in the length field to compute the number of bytes in the record. A value is an integer in the range 1 (the default) to 15. The default is 1.

LTHFLD

This parameter specifies the length of the field containing the record length.

Syntax RECORD LTHFLD = *option(s)*

Options Table 4-28 lists the parameter option(s) and definition(s).

Table 4-28. **LTHFLD parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>size</i>	Specifies, in bytes, the record length field length. It must be an integer in the range 0 to 5. If the size is set equal to zero, record lengths are not contained in the records, and the record length is the maximum length (LENGTH) for each record. The default is 0.

OFFSET

This parameter specifies the location of the record length field.

Syntax RECORD OFFSET = *option(s)*

Options Table 4-29 lists the parameter option(s) and definition(s).

Table 4-29. **OFFSET parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the record length field offset. This offset is the byte offset from the first byte of the record to the record length field. A value must be an integer in the range 0 (the default) to LENGTH-LTHFLD-1. The default is 0.

POSTAMBLE

This parameter specifies the length of any extraneous data at the end of the record.

Syntax RECORD POSTAMBLE = *option(s)*

Options Table 4-30 lists the parameter option(s) and definition(s).

Table 4-30. **POSTAMBLE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>length</i>	Specifies the length in bytes. The length must be an integer in the range 0 (the default) to record length. The default is 0.

PREAMBLE

This parameter specifies the offset to the user portion of the record, that is, the record preamble length.

Syntax RECORD PREAMBLE = *option(s)*

Options Table 4-31 lists the parameter option(s) and definition(s).

Table 4-31. **PREAMBLE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>length</i>	The byte offset from the first byte of the record to the first byte of the user's portion of the record. The length must be an integer in the range 0 (the default) to record length. The default is 0.

STRUCTURE

This parameter specifies the general record structure of the input data.

Syntax RECORD STRUCTURE = *option(s)*

Options Table 4-32 lists the parameter option(s) and definition(s).

Table 4-32. **STRUCTURE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>structure-type</i>	<p>May be any of the following:</p> <ul style="list-style-type: none"> • FB (fixed length blocked) • F (fixed length) • V (variable length) • VB (variable length blocked) • U (undefined length) • UB (undefined length blocked) <p>For offline processing, the tape label contents may override this parameter.</p> <p>The default is FB.</p>

Points to note

Note the following when using the RECORD command:

- **LENGTH**

May be overridden by ANSI, IBM OS/Standard, or Honeywell 2000 COBOL labels that specify record length.

- **LTHFLD, OFFSET, FORMAT, and PREAMBLE**

The values for these may be overridden if STRUCTURE is changed as the result of ANSI, IBM OS/Standard, or Honeywell 2000 COBOL label processing.

- **CONSTANT and STRUCTURE**

CONSTANT may be enabled as the result of STRUCTURE being changed to U in label processing. However, no definition is assumed for the constant string. The default must be zero, or it must be defined in the job descriptor entry.

The LENGTH on a 4 by 3 packed format tape is the number of 6-bit bytes or characters in the record.

- **LINE command, DATA parameter, and RECORD command LENGTH parameter**

When the DATA parameter of the LINE command is changed to a value greater than the default value, the LENGTH parameter of the RECORD command must also be changed accordingly. (Refer to the "LINE command" section of the chapter "Specifying output parameters" for additional information.)

For online processing, print position indexing must not be used if LENGTH greater than 150 is specified.

- **Online versus offline JDLs**

An offline JDL statement set that modifies the system default values, specifically in the BLOCK or RECORD commands, may experience incorrect results if running an online job (JDE). In other words, if an online job (JDE) is called out in an offline JDL that has changed the system default values (of the BLOCK and RECORD command parameters), the job may not print correctly. It is best, then, to separate and run online and offline jobs (JDEs) independently from each other.

Example

The RECORD command for the sample input record illustrated below would be coded as follows:

```
RECORD LENGTH=133, OFFSET=2, LTHFLD=2,
      PREAMBLE=4, ADJUST=4, FORMAT=BIN;
```

Figure 4-2 shows sample values for RECORD command parameters.

Figure 4-2. **Sample RECORD command**

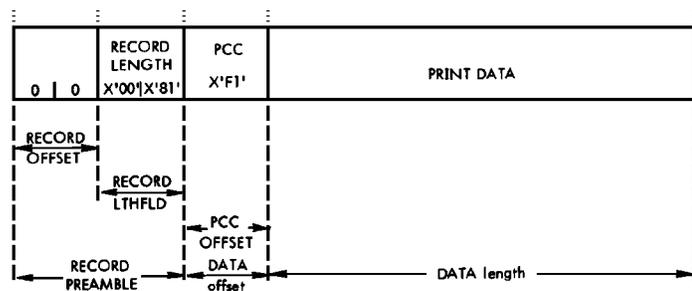
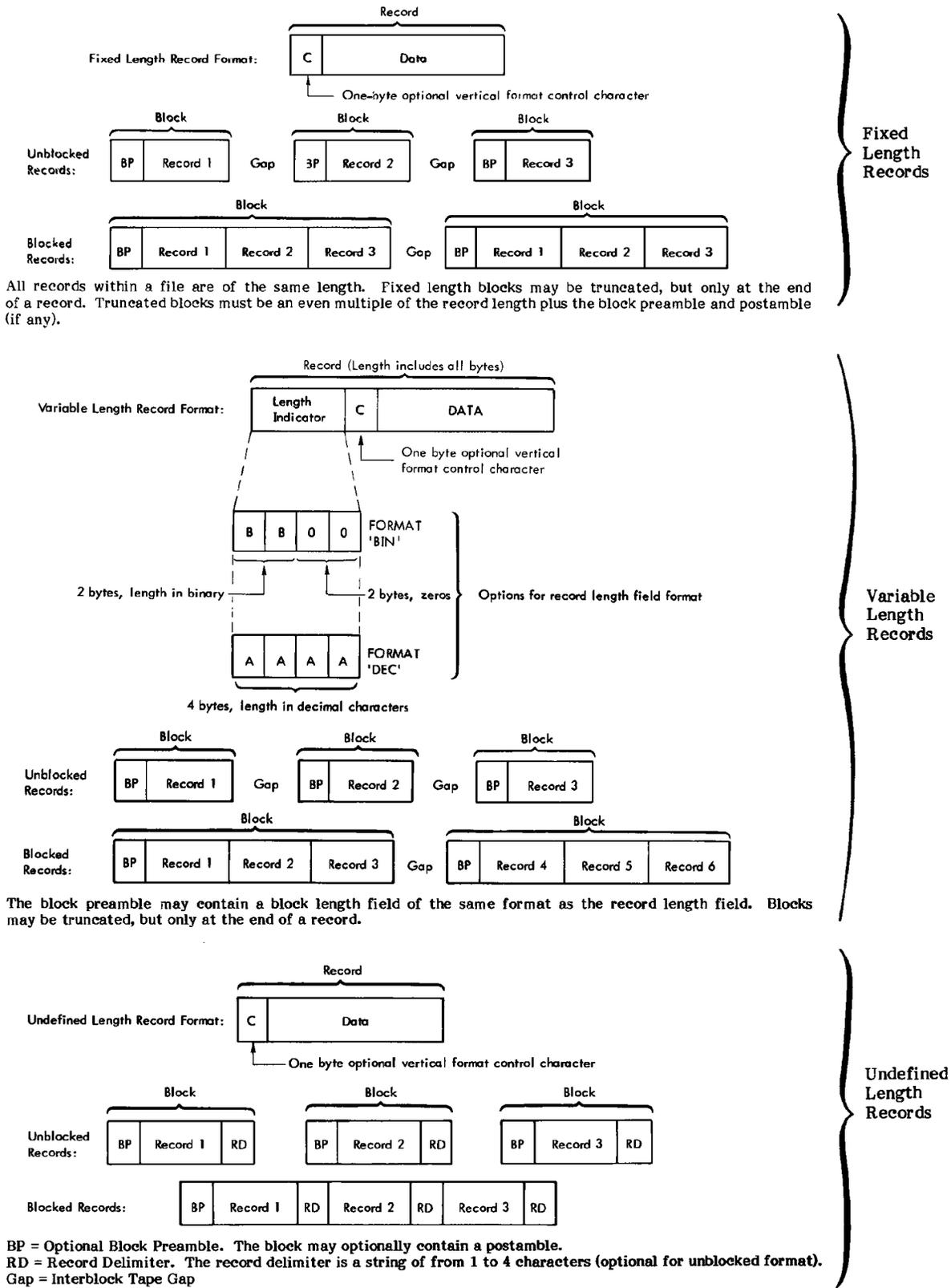


Figure 4-3 shows record format types

Figure 4-3. Record format types



SEFFNT command

Paper is fed from the short edge to allow for printing on large size paper. This is called short-edge feed (SEF). In order to maintain the current definition of landscape and portrait in these jobs, a mapping must be defined in which every landscape font used in a SEF job has a portrait equivalent and vice versa.

Table 4-33 summarizes the use of the SEFFNT command parameters.

Table 4-33. **Summary of SEFFNT command parameters**

Parameter	Specifies	Offline	Online	DJDE
MAP	A previously created font mapping	Y	Y	Y
SEFMAP	Font mapping where the second specified font is the SEF mapping for the first font	Y	Y	Y

Mapping files

A font mapping associates one font to another font. A collection of font mappings may be stored in a mapping file. Font files are files having the FNT extension and are stored on disk.

The SEFFNT command is used to enable or disable the short-edge feed feature. SEFFNT creates customized mapping files (of type LIB) or triggers the system to invoke short-edge feed processing. It must be used in conjunction with a PDE. The SEFFNT command format is as follows:

```
[label:] SEFFNT
      SEFMAP={ (font1, font2) [ , (fontm, fontn) ] | NONE }
      MAP = file-name
```

Table 4-34 lists the parameter definitions, and the parameter option(s) and definition(s).

Table 4-34. **SEFMAP and MAP parameters and parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>label</i>	Specifies the name of the mapping file that is to be created. If the label does not exist, the SEFFNT command either creates an internal table or uses an existing mapping file if MAP is specified. If the [label] option is omitted, SEFMAP puts all mapping within a JDL rather than creating a file on the system disk.
SEFMAP	Sets up the font mapping. In each specified font pair, the second font is the SEF mapping for the first font. Multiple level mappings are not allowed.
(<i>font₁</i> , <i>font₂</i>)	Font pair in which <i>font₂</i> is the SEF mapping for <i>font₁</i> .
(<i>font_m</i> , <i>font_n</i>)	Font pair in which <i>font_m</i> and <i>font_n</i> is the SEF mapping for <i>font₁</i> .
NONE	Disables font mapping and short-edge feed; stops previous SEFFNT.
MAP	References a previously defined font mapping file (<i>file-name</i>) from a [label:]SEFFNT command.

The SEFFNT command has different effects depending on the context:

- Bypassing font mapping
- Mapping contained in the JDL
- Standalone mapping files
- JDL creates a mapping file.

The following sections describe the syntax of the command parameters and explanations of the parameter options.

SEFMAP

When coded outside of a standard JSL (either prior to a JDL command or after an END command), a file containing the font mappings by the name of label.LIB is created on a disk for reference within the JSL by other JSLs or through a DJDE.

When specified without a label preceding the SEFFNT command, this SEFMAP parameter puts all font mapping definitions within the JSL, rather than creating a separate mapping file.

When coded at the catalog level, SEF mapping applies to all JOBS or JDEs that include the catalog. When coded at the job level, SEF mapping specifically assigns a mapping to be used with this JOB or JDE.

Syntax [label:]SEFFNT SEFMAP = *option(s)*

Options Table 4-35 lists the parameter option(s) and definition(s).

Table 4-35. **SEFMAP parameter option(s) and definition(s)**

Option(s)	Definition(s)
{ (<i>font</i> ₁ , <i>font</i> ₂) [, (<i>font</i> _{<i>m</i>} , <i>font</i> _{<i>n</i>})] NONE }	<p>This option has the following components:</p> <ul style="list-style-type: none"> • (<i>font</i>₁, <i>font</i>₂) Font pair, where <i>font</i>₂ is the SEF mapping for <i>font</i>₁. • (<i>font</i>_{<i>m</i>}, <i>font</i>_{<i>n</i>}) Font pair, where <i>font</i>_{<i>n</i>} is the SEF mapping for <i>font</i>_{<i>m</i>}. • NONE Disables font mapping and short-edge feeding (SEF); stops previous SEFFNT. <p>There is no default.</p>

MAP

This parameter references a previously defined font mapping file (*file-name*) from a [label:]SEFFNT command.

Syntax [label:]SEFFNT MAP = *option(s)*

Options Table 4-36 lists the parameter option(s) and definition(s).

Table 4-36. **Parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>file-name</i>	<p>References a previously created font mapping file from a [label:] SEFFNT command (assumes the label already exists).</p> <p>There is no default.</p>

Points to note

Note the following when using the SEFFNT command:

- When the system recognizes a SEF job, it scans the PDE font list and substitutes them with font mappings defined in the JSL. If you specify that font mappings are contained in a file, the input task checks to see if this file exists. If the mapping file or the font file does not exist, job processing continues using the font specified in the PDE command. If no font mapping exists, you receive a CON I or ABO I message on the screen and have the option to continue processing using the font specified in the PDE or abort the job.
- The system does not assume any mappings for fonts. You must always provide the mapping. Once the font mapping is found, range checking is based on the new orientation. The range checking determines only if the BEGIN values are within the bounds of the page—no check is made to determine whether the line will run off the edge of the page.



Note: Interpress SEF jobs through XNS, XPS, or XPAF are not supported.

- Input will use the first font in the PDE list (the list that has been substituted for the new font mappings) for the following system pages:
 - OPRINFO pages (DJDE)
 - Label
- The following system pages use predefined forms and thus are not reformatted in order to print them in the same orientation as their associated jobs (that is, no font mappings or rotations of forms performed):
 - Graphic samples
 - Graphic summary pages
 - Accounting pages.
- Short-edge feed duplex printing differs from long-edge feed (LEF) applications in that SEF uses a head-to-toe orientation for printing portrait pages, while LEF uses head-to-head orientation. Refer to the INVERT parameter of the OUTPUT command for alternate methods of rotating the page orientation.

For landscape printing, however, SEF uses a head-to-head orientation, whereas LEF landscape uses head-to-toe orientation.
- If you want to use 11 by 17 inch paper or any size that requires short-edge feed (SEF), be aware that form source libraries (FSLs), forms (in FRM files), logos (in LGO files), and images (in IMG files) are not automatically scaled or rotated. (Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for detailed information on scaling.)

- ($font_1, font_2$) - means that the fact that $font_1$ is mapped $font_2$ and $font_2$ is mapped to $font_3$ does not imply that $font_1$ is mapped to $font_3$.
- ($font_m, font_n$) - means that the fact that $font_m$ is mapped $font_n$ and $font_n$ is mapped to $font_z$ does not imply that $font_m$ is mapped to $font_z$.

Examples

Following are a few examples of how to use the command.

Example 1 FNTMAP:SEFFNT SEFMAP=(L0112B,P0612A);
SYS1:SYSTEM;
VOLUME `CODE`=ASCII;

FNTMAP.LIB is created on system disk.

Example 2 SEFFNT SEFMAP=(L0114B,P0614A);
PDE1: PDE FONT=L0114B,P0614A,PMODE=LANDSCAPE,
BEGIN=(.50,.50);

Example 3 JOB1: JOB;
OUTPUT FORMAT=PDE1;

Job 1 uses the SEFFNT command in example 2 because there is no override specified.

Example 4 JOB2: JOB;
OUTPUT FORMAT=FMT1;
SEFFNT MAP=FNTMAP;

Job 2 uses the FNTMAP.LIB file created in example 1.

Example 5 JOB3: JOB;
OUTPUT FORMAT=FMT6;
SEFFNT SEFMAP=NONE;

There is no font mapping because SEF is disabled by the SEFMAP=NONE parameter.

Example 6 JOB4: JOB;
SEFFNT SEFMAP=(L01BOB,P0812A);

Job 4 overrides the system level SEFFNT mapping with a job level definition.

TCODE command

The TCODE command defines a set of input-character-to-type assignments to be used for the masked string comparisons in a report. It must be used whenever it is not possible to use one of the standard sets of default type assignments without modification.

Table 4-37 summarizes the use of the TCODE command parameters.

Table 4-37. **Summary of TCODE command parameters**

Parameter	Specifies	Offline	Online	DJDE
DEFAULT	Initial set of character type assignments for use with masked comparisons	Y	Y	N
TASSIGN	Association of one or more specified characters with one or more specified typecodes	Y	Y	N
TRESET	Disassociation of one or more characters, or all characters, in a set from one or more typecodes	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

The TCODE command should be coded as follows:

- Specify with one parameter, using the DEFAULT parameter, an initial set of type assignments for the entire set of input character codes.
- Modify the specified initial set of type assignments by associating specific character types with specific characters in the input character set (as many occurrences as necessary of the TASSIGN parameter).
- If necessary, further modify the initial set by disassociating specific character types from specific characters in the input character set (as many occurrences as necessary of the TRESET parameter).

DEFAULT

This parameter specifies an initial set of character type assignments for use with masked comparisons. Default assignments for any standard character set may be specified by coding the appropriate keyword.

Syntax TCODE DEFAULT = *option(s)*

Options Table 4-38 lists the parameter option(s) and definition(s).

Table 4-38. **DEFAULT** parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>tcode-type</i>	<p>Available tcode-type keyword parameters are:</p> <ul style="list-style-type: none"> • ASCII • BCD • EBCDIC • PEBCDIC • H2BCD • H6BCD • IBMBCD • <i>value</i> <p>Initial typecodes are set to allow the following: all numeric characters (0-9) are type 1 characters, and all uppercase and lowercase alphabetic characters (A-Z, a-z) are type 2 characters. All other characters, if any, are untyped characters.</p> <p>There is no default.</p>

The full set of assignments for any nonstandard (user-defined) character set may be specified by coding a value with an integer number ranging in value from 0 through 7. If a 0 is specified, all the characters are untyped; if a 1 is specified, all the characters are type 1; if a 7 is specified, all the characters are type 7. The initial set of type assignments may be modified by specifying the TASSIGN and TRESET parameters.

TASSIGN

This parameter associates one or more specified characters with one or more specified typecodes. This parameter may also be used to disassociate one or more characters from all typecode assignments. The TASSIGN parameter may be specified more than once in a TCODE command.

Syntax TCODE TASSIGN = *option(s)*

Options Table 4-39 lists the parameter option(s) and definition(s).

Table 4-39. TASSIGN parameter option(s) and definition(s)

Option(s)	Definition(s)
<p>(<i>typespec</i>, '<i>inputspec</i>') ('<i>inputspec</i>₁' ['<i>inputspec</i>₂'] [...])</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>typespec</i> A <i>typespec</i> may be coded as a single integer, <i>n</i>. When coded as a single integer, the value may be in the range of 0 to 7. When coded as a series of integers, the values may be within the range of 1 through 7. • <i>inputspec</i> An <i>inputspec</i> may be coded as a single character, <i>char</i>, as a series of characters, (<i>char</i>[,...]), or as a range of characters, (<i>char</i>₁,-<i>char</i>₄). The syntax <i>char</i>₁,-<i>char</i>₂ means the set of input codes corresponding to characters bounded by and including <i>char</i>₁ and <i>char</i>₂ (refer to appendix D for relative order of characters). <p>When a <i>typespec</i> is coded as a 0, all characters specified by an <i>inputspec</i> are disassociated from all typecode assignments. Thus, the TASSIGN parameter may untype a specified set of characters. When a <i>typespec</i> is coded with a single value (ranging from 1 through 7), all specified characters are associated with that typecode. When a <i>typespec</i> is coded as a series of values (all ranging from 1 through 7), the specified characters are each associated with all of the specified types. A <i>typespec</i> may also be specified with keywords: NUMERIC in lieu of 1 or ALPHA in lieu of 2.</p> <p>There is no default.</p>

TRESET

This parameter disassociates one or more characters (or all characters) in a set from one or more types.

Syntax TCODE TRESET = *option(s)*

Options Table 4-40 lists the parameter option(s) and definition(s).

Table 4-40. **TRESET parameter option(s) and definition(s)**

Option(s)	Definition(s)
typespec,'inputspec') ('inputspec1',['inputspec2'], [...])ALL	This option has the following components: <ul style="list-style-type: none"> • <i>typespec</i> A <i>typespec</i> may be coded as a single integer, <i>n</i>, within the range of 1 through 7. • <i>inputspec</i> When an <i>inputspec</i> is coded with the keyword ALL, all characters in the code table are disassociated from the specified types. The TRESET parameter may be specified more than once in the TCODE command. • ALL Specifies that all characters in the code table are disassociated from the specified types. There is no default.

Points to note

Note the following when using the TCODE command:

- The LPS accepts lowercase characters expressed literally or symbolically, that is, preceded by the pound sign (#) functioning as a switch case operator.
- A hyphen (-) specifies the range of contiguous characters (as in the examples below), where the contiguity is based upon actual codes. If input codes are associated with ASCII characters, the letter A can be represented as hexadecimal 41; the letter B as 42,..., the letter Z as 5A. If input codes are associated with EBCDIC characters, the contiguity of the input codes is interrupted between the letters I and J, between the letters R and S and, thus, must be specified as three separate ranges.
- Coding of a TCODE command is dependent on:
 - Operational requirements, such as using a defined character set, for example, ASCII or EBCDIC, in which type 1 and type 2 character sets are defined by default and using TASSIGN parameters to define additional character sets, for example, type 3, type 4...
 - Capability of the operator's console to process and display lowercase characters literally rather than symbolically.

The examples in table 4-41 illustrate how the TCODE command may be coded to specify the following types of character sets.

Table 4-41. **Specifying various types of character sets**

Type	Interpretation
1	Defines numeric characters (0-9) by default.
2	Defines alphabetic characters (A-Z and a-z) by default.
3	Defines uppercase alphabetic characters (A-Z) by TCODE command.
4	Defines lowercase alphabetic characters (a-z) by TCODE command.
None	Characters which are neither numeric nor alphabetic.

Examples

Following are a few examples of how to use the command.

- Example 1** Basic character set used for default types is ASCII.
Operator's keyboard display has uppercase and lowercase character capability:

```
T1: TCODE DEFAULT=ASCII,
TASSIGN=(3, (A 'A', - ,A 'Z')),
TASSIGN=(4, (A 'a', - ,A 'z'));
```

- Example 2** Basic character set used for default types is EBCDIC.
Operator's keyboard display has uppercase and lowercase character capability:

```
T2: TCODE DEFAULT=EBCDIC
TASSIGN=(3, ('A', - , 'I'), ('J', - , 'R'), ('S', - , 'Z')),
TASSIGN=(4, ('a', - , 'i'), ('j', - , 'r'), ('s', - , 'z'));
```

- Example 3** Basic character set used for default types is ASCII.
Operator's keyboard display is limited to uppercase only:

```
T3: TCODE DEFAULT=ASCII
TASSIGN=(3, (A, 'A', - ,A 'Z')),
TASSIGN=(4, (A, '#A', - ,A '#Z'));
```

The # symbol functions as a case switching operator.

- Example 4** Basic character set used for default types is EBCDIC.
Operator's keyboard display is limited to uppercase only:

```
T4: TCODE DEFAULT=EBCDIC
TASSIGN=(3, ('A', - , 'I'), ('J', - , 'R'), ('S', - , 'Z')),
TASSIGN=(4, ('#A', - , '#I'), ('#J', - , '#R'), ('#S', - , '#Z'));
```

VOLUME command

Table 4-42 summarizes the VOLUME command parameters.

Table 4-42. **Summary of BLOCK command parameters**

Parameter	Specifies	Offline	Online	DJDE
BMULT	Multiplication factor to determine true block length	Y	N	N
CODE	Input data translation	Y	Y	N
EOV	End-of-volume processing	Y	N	N
HOST	Source of input data for processing	Y	Y	N
INTERPRESS	Input tape contains Interpress data	Y	N	N
LABEL	Type of tape label processing	Y	N	N
LCODE	Input tape label translation	Y	N	N
LPACK	Label packing specification for undefined labels	Y	N	N
MAXLAB	Maximum label length for undefined labels	Y	N	N
MINLAB	Minimum label length for undefined labels	Y	N	N
OPTIMIZE	Throughput enhancement for online	N	Y	N
OSCHN	OS Writer end-of-report channel	Y	N	N
OSHDP	OS Writer header banner page count	Y	N	N
PLABEL	Printing of labels to sample tray	Y	N	N
RMULT	Multiplication factor to determine true record length	Y	N	N
RSAT	Handling of disk saturation on multivolume job	Y	N	N
TCODE	Masked comparison type assignments	Y	Y	N
UNPACK	Input data unpacking method	Y	N	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

BMULT

This parameter specifies a multiplication factor you can use to determine true block length.

Syntax VOLUME BMULT = *option(s)*

Options Table 4-43 lists the parameter option(s) and definition(s).

Table 4-43. **BMULT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies a multiplication factor that you extract from the tape label and apply to the block length to determine the true block length. The value is an integer and can range from 1 (default) to 15 The default is 1.

CODE

This parameter specifies which code translation table the system uses to interpret the input data.

Syntax VOLUME CODE = *option(s)*

Options Table 4-44 lists the parameter option(s) and definition(s).

Table 4-44. **CODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>keyword</i>	<p>Specifies a specific, standard, system-defined code translation table. The available options are:</p> <ul style="list-style-type: none"> • ASCII • BCD • EBCDIC • H2BCD • H6BCD • IBMBCD • PEBCDIC • NONE • USER <p>USER refers to the single unlabeled CODE command defined within the JDE or JDL.</p> <p>If you specify more than one CODE parameter, use identifiers so that they can be referenced in your JSL. You must then specify the variable identifier (<i>id</i>) to correspond to the identifier label assigned to the defining CODE command.</p> <p>The default is EBCDIC.</p>
<i>id</i>	<p>Specifies a variable identifier you use in the CODE command to refer to a specific user-defined code translation table. The identifier <i>id</i> is required when multiple CODE commands are contained within the JDL.</p>
NONE	<p>Specifies that data is not to be translated by input processing.</p>

Considerations If the system reports missing fonts when running XPPI jobs with CODE=NONE specified, the operator can elect to continue by pressing the <CON> key. The system then reverts back to the previous PDE, which may have contained fewer specified fonts. In this case, a nonexistent font could have been specified for this PDE, causing a rollover to occur.

When you run a CODE=NONE job with DUPLEX=YES and SHIFT=YES, you must specify MARGIN in m inches.

Input data is translated according to the CODE selected on the VOLUME command. Any character that is translated into an ASCII X'20' is truncated after the last significant (non-ASCII X'20') character in the print record.

EOV

This parameter specifies the action the system takes when it encounters an end-of-volume (EOV) on the input data tape.

Syntax VOLUME EOV = *option(s)*

Options Table 4-45 lists the parameter option(s) and definition(s).

Table 4-45. **EOV parameter option(s) and definition(s)**

Option(s)	Definition(s)
([PAUSE NOPAUSE], [EOF NOEOF])	This option has the following components: <ul style="list-style-type: none"> • PAUSE Displays a message when EOV is encountered. A CONTINUE response by the operator causes the tape to rewind and the normal volume change sequence to proceed. • NOPAUSE Issues a rewind instruction as soon as the EOV label is processed. • EOF End-of-file. Specifies that the end-of-volume label is treated as an end-of-file label. When this occurs, the first part of the page spanning the volumes is output as the last page of the job. The second part of the spanned page is printed as the first page when the next volume is started, with possible page format irregularities. To use the EOF parameter effectively, special user-formatting of the multireel tapes is required to avoid these page format problems. • NOEOF Specifies normal end-of-volume processing. The default is (NOPAUSE,NOEOF).

HOST

This parameter specifies the computer or host operating system generated the input data file.

Syntax VOLUME HOST = *option(s)*

Options Table 4-46 lists the parameter option(s) and definition(s).

Table 4-46. **HOST** parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>host-type</i>	Specifies one of the following options: <ul style="list-style-type: none"> • IBMOS • ANSI • B2500 • B2700 • B3500 • B3700 • B4700 • B6700 • DEC • PDP11 • DUMP • GRASP • H2000 • H6000 • IBMDOS • IBMONL • ICL2900 • NCR • OCTDUMP • OLDUMP • OSWTR • POWER • POWERVS • RSX11 • UNDEF • UNIVAC 1100 • US70 • XEROX • ENET The default is IBMOS.

Considerations

For online systems, you must specify IBMONL or OLDUMP.

For Ethernet, you must specify ENET.

For HOST=ICL2900, the PCCTYPE is always specified as NONE.

For HOST=RSX11(FLX), you must specify the DATA command.

For HOST=DUMP, two tape marks at the end of the tape are required to dump the data tape using DUMP.JSL.

If a selected HOST type is inconsistent with a selected LABEL type, the PDL issues an error message and automatically substitutes a valid label. Do not code the LABEL parameter of the VOLUME command in cases where it is not appropriate for a particular HOST type. Refer to the "Offline specifications" appendix for more information on offline formats.

If an online job (JDE) is called out in an offline JDL that changed the system default values, the job may print incorrectly. If this happens, separate and run online and offline jobs independent from each other

INTERPRESS

This parameter provides additional control over the Interpress environment.

Syntax VOLUME INTERPRESS = *option(s)*

Options Table 4-47 lists the parameter option(s) and definition(s).

Table 4-47. **INTERPRESS parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that the data stream is Interpress. NO is not an option.
[CHECKSUM NOCHECKSUM]	<p>This option has the following components:</p> <ul style="list-style-type: none"> • CHECKSUM Enables error-checking procedures on a job basis. If CHECKSUM is specified and verification fails, the job aborts. • NOCHECKSUM Disables the error-checking procedures. <p>This default is NOCHECKSUM.</p>
[BREAKPAGE NOBREAKPAGE]	<p>This option has the following components:</p> <ul style="list-style-type: none"> • BREAKPAGE Requests a trailing break page is printing instructions are present in the job. • NOBREAKPAGE Specifies a break page only if the system detects errors. <p>This default is NOBREAKPAGE.</p>
INTEGRAL	<ul style="list-style-type: none"> • INTEGRAL Requests integral scale factors from 1 to 8 for IMG, IM6, and most RES (Xerox Compressed) images.
PERFORMANCE	<ul style="list-style-type: none"> • PERFORMANCE Specifies, with the raster imager processor (RIP) that raster encoding standard (RES) files containing sampled color are imaged by the graphics hardware whenever possible. The RIP is used only when graphics hardware capabilities are exceeded.

LABEL

This parameter specifies the type of label on the input tape.

Syntax VOLUME LABEL = *option(s)*

Options Table 4-48 lists the parameter option(s) and definition(s).

Table 4-48. **LABEL parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>label-type</i>	Specifies one of the following options: <ul style="list-style-type: none">• NONE (the input tape is unlabeled)• ANSI• STANDARD• SPR (Honeywell 2000 System Print Tape)• COBOL (Honeywell 2000 COBOL tape with 120-byte labels)• UNDEF (Input tape label is not defined)

LCODE

This parameter specifies the code translation table to use in interpreting the tape label.

Syntax VOLUME LCODE = *option(s)*

Options Table 4-49 lists the parameter option(s) and definition(s).

Table 4-49. **LCODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>keyword</i>	<p>Specifies a specific, standard, system-defined code translation table. The available options are:</p> <ul style="list-style-type: none"> • ASCII • BCD • EBCDIC • H2BCD • H6BCD • IBMBCD • PEBCDIC • NONE • USER <p>USER refers to the single unlabeled CODE command allowed within the JDE or JDL.</p> <p>You must specify the variable identifier <i>id</i> to correspond to the identifier label assigned to the defining CODE command.</p> <p>The default is EBCDIC.</p>
<i>id</i>	<p>Specifies a variable identifier you use in the CODE command to refer to a specific user-defined code translation table. The identifier <i>id</i> is required when multiple CODE commands are contained within the JDL.</p>
NONE	<p>Specifies that data is not to be translated by input processing. For example, you select NONE for XPPi jobs that contain meta codes.</p>

LPACK

This parameter specifies whether undefined labels (LABEL=UNDEF) are packed or unpacked.

Syntax VOLUME LPACK = *option(s)*

Options Table 4-50 lists the parameter option(s) and definition(s).

Table 4-50. **LPACK parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Indicates that no unpacking operation is to be performed. If YES is specified, the UNPACK parameters specify the required unpacking method for both the labels and the input data. The labels do not have to be packed even though the data may require unpacking. The default is NO.
YES	Indicates that the labels are packed.

MAXLAB

This parameter specifies the longest physical block to treat as a label for undefined labels (LABEL=UNDEF).

Syntax VOLUME MAXLAB = *option(s)*

Options Table 4-51 lists the parameter option(s) and definition(s).

Table 4-51. **MAXLAB parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies in bytes the maximum length of the longest physical block. The value can range from 2 to 4096 bytes. The specified value must be greater than the MINLAB value and must not exceed the BLOCK LENGTH. The default is 81.

Considerations Any data block whose length is between MINLAB and MAXLAB (inclusive) is treated as a label.

MINLAB

This parameter specifies the smallest physical block to treat as a label for undefined labels (LABEL=UNDEF).

Syntax VOLUME MINLAB = *option(s)*

Options Table 4-52 lists the parameter option(s) and definition(s).

Table 4-52. **MINLAB parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies in bytes the maximum length of the smallest physical block. The value can range from 1 to 4095 bytes. The specified value must be less than the MAXLAB value and must not exceed the BLOCK LENGTH.</p> <p>The default is 80.</p>

OPTIMIZE

This parameter enhances the performance of the data stream.

Syntax VOLUME OPTIMIZE = *option(s)*

Options Table 4-53 lists the parameter option(s) and definition(s).

Table 4-53. **OPTIMIZE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>keyword</i>	<p>Specifies one of the following options:</p> <ul style="list-style-type: none"> • NCCHECK Disables channel 9 and channel 12 detection. The host is not notified by the LPS each time channel 9 or 12 is recognized when this option is selected. Any skips to channel 9 or 12 are allowed as for any other channel. • NDCHECK Disables reporting the detection of an unprintable character as defined by the UCSB. The host computer system is not notified by the LPS of this condition, even if the host sent an ALLOW DATA CHECK parameter. The unprintable character is printed as a blank by the LPS whether or not this parameter is specified. • NPRECOVER Enhances online data stream performance by minimizing the number of disk accesses on the LPS. If you specify this option to recover from a system abort, the user may have to retransmit the last several pages preceding the system abort. <p>The maximum performance occurs when all the parameters are selected, but there is still significant improvement when you specify only NCCHECK and NDCHECK.</p> <p><i>(keyword₁ [,keyword₂][,keyword₃])</i> allows you to specify more than one of the following options: NCCHECK, NDCHECK, and NPRECOVER. The maximum performance occurs when all the parameters are selected, but there is still significant improvement when you specify only NCCHECK and NDCHECK.</p> <p>When you specify both NCCHECK and NDCHECK, data stream performance is significantly enhanced because input processing of a record and the subsequent data transfer of the next record occurs immediately.</p>
NONE	Does not invoke the optimization parameter.

OSCHN

This parameter specifies the channel the system uses to signal the end of a report generated by an IBM OS Writer.

Syntax VOLUME OSCHN = *option(s)*

Options Table 4-54 lists the parameter option(s) and definition(s).

Table 4-54. **OSCHN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the vertical format unit (VFU) channel the system uses to signal the end of a report. When a skip (determined by the printer carriage control (PCC) field within a logical record) to the specified channel occurs, the IBM OS Writer banner page is considered found. The default is 9.

OSHDP

This parameter specifies the number of header banner pages generated by an IBM OS Writer.

Syntax VOLUME OSHDP = *option(s)*

Options Table 4-55 lists the parameter option(s) and definition(s).

Table 4-55. **OSHDP parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the number of header pages which precede the report. The default is 0.

OSTLP

This parameter specifies the number of trailer banner pages generated by an IBM OS Writer.

Syntax VOLUME OSTLP = *option(s)*

Options Table 4-56 lists the parameter option(s) and definition(s).

Table 4-56. **OSTLP parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the number of trailer pages which precede the report. The default is 0.

PLABEL

This parameter specifies whether the tape labels are printed.

Syntax VOLUME PLABEL = *option(s)*

Options Table 4-57 lists the parameter option(s) and definition(s).

Table 4-57. **PLABEL parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Specifies that no tape label printing results. The default is NO.
YES	Specifies that all tape labels, (except those encountered during a volume change) are printed on an output page and delivered to the sample print tray. Labels are truncated if they exceed the line width limits of the page.

RMULT

This parameter specifies a multiplication factor the record length uses to determine the true record length.

Syntax VOLUME RMULT = *option(s)*

Options Table 4-58 lists the parameter option(s) and definition(s).

Table 4-58. **RMULT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	An integer from 1 to 15. The default is 1.

RSAT

This parameter specifies whether to make a complete pass over the multivolume set for each copy of a report that is processing (REMOUNT) or whether a report split (SPLIT) occurs at the point of disk saturation.

Syntax VOLUME RSAT = *option(s)*

Options Table 4-59 lists the parameter option(s) and definition(s).

Table 4-59. **RSAT parameter option(s) and definition(s)**

Option(s)	Definition(s)
REMOUNT	Specifies the vertical format unit (VFU) channel that the system uses to signal the end of a report. When a skip (determined by the printer carriage control (PCC) field within a logical record) to the specified channel occurs, the IBM OS Writer banner page is considered found. The default is 9. The default is REMOUNT.
SPLIT	Indicates that only one pass is made over the input data and that the reports are in sections and therefore require manual merging. Input processing is faster for SPLIT since only one pass is required over the input data. SPLIT is recommended for graphics processing when using page interleaved or move mode, since moving a large amount of graphic data impacts data stream performance.

TCODE

This parameter specifies the set of type assignments to use for masked comparisons.

Syntax VOLUME TCODE = *option(s)*

Options Table 4-60 lists the parameter option(s) and definition(s).

Table 4-60. **TCODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>tcode-type</i>	<p>Specifies one of the following standard character sets for which type assignments are defined by default:</p> <ul style="list-style-type: none"> • ASCII • BCD • EBCDIC • PEBCDIC • H2BCD • H6BCD • IBMCD <p>By coding any one of the keywords, you are furnished with a set of character-to-type assignments.</p>
<i>tcode-id</i>	Specifies an identified label of a TCODE parameter that defines a set of type assignments.

UNPACK

This parameter specifies the unpacking method of the input data when required.

Syntax VOLUME UNPACK = *option(s)*

Options Table 4-61 lists the parameter option(s) and definition(s).

Table 4-61. **UNPACK parameter option(s) and definition(s)**

Option(s)	Definition(s)
NONE	Does not perform the unpacking operation. The default is NONE.
T4X3 or T4X3H2	Specifies unpacking routines in which 6-bit characters are extracted and restored as 8-bit bytes. T4X3 is used primarily for Honeywell 6000 tapes while T4X3H2 is used for Honeywell 2000 tapes.
UNIVAC	Specifies an unpacking routine used for UNIVAC tapes, although you can specify any of these routines independent of the HOST type.

Points to note

Note the following when using the VOLUME command:

- The character code assignment tables in appendix C summarize the label specifications that are valid for each host type. If a selected HOST type is inconsistent with a selected LABEL type, an error message is output by PDL, and a valid label is automatically substituted. The LABEL parameter of the VOLUME command should not be coded in cases where it is not appropriate for a particular host type.
- A JDL is supplied with the printing system software to dump the contents of a tape. Six standard tape formats may be dumped with this JDL (H6000 BCD, EBCDIC, ASCII, H2000 BCD, IBM BCD, UNIVAC FIELDDATA). Check the listing of the DUMP.JSL file to determine the appropriate JDE for the tape to be dumped. Other tape formats may be dumped by specifying the host-type (DUMP or OCTDUMP) and appropriate parameters, such as CODE and UNPACK, in a user-coded JDL.
- The MAXLAB and MINLAB on a 4 by 3 packed format tape are the number of 6-bit bytes or characters in the tape blocks that are treated as labels.
- Input data is translated according to the CODE parameter option selected on the VOLUME command. Any character which is translated into an ASCII X'20' is truncated after the last significant (non-ASCII X'20') character in the print record.
- When running a CODE=NONE job with DUPLEX=YES and SHIFT=YES, MARGIN must be specified in *m* inches.

For example:

```
m=-V2-BEGIN-ALI + .01
```

Table 4-62 lists the components of this example and their definitions.

Table 4-62. **Example components and definitions**

Component	Definition
<i>m</i>	Positive decimal number of the form nnn.dd (up to two digits to the right of the decimal point)
V ₂	Even (back) side shift value in inches
BEGIN	Scan line in inches as specified in the PDE (hpos in portrait, vpos in landscape)
ALI	Anticipated scan alignment (estimated inches equivalent, 3 scans=.01 in.)

If *m* is computed as less than or equal to zero, MARGIN should be specified as zero or ignored altogether. Failure to do so results in printing blank even-numbered pages (back sides). Note that varying the margin does not alter the appearance of the printed page because of the absolute positioning information embedded in the data.

Examples

The following are sample VOLUME commands for several different host types.

```
VOLUME HOST=UNIVAC , LABEL=STANDARD , UNPACK=T4X3 ,  
        CODE=ASCII , LCODE=ASCII ;  
VOLUME HOST=IBMOS ;  
VOLUME HOST=IBMONL , LABEL=NONE , INTERPRESS=YES ,  
        CODE=EBCDIC ;  
VOLUME HOST=B6700 , LABEL=ANSI , BMULT=6 , RMULT=6 ,  
        PLABEL=YES ;  
VOLUME HOST=H2000 , LABEL=SPR , CODE=H2BCD ,  
        UNPACK=T4X3H2 ;  
VOLUME HOST=OCTDUMP , CODE=H6BCD , UNPACK=T4X3 ;
```

Input sources

Data to be processed by the printing system may originate from several sources. These sources are magnetic tape (offline processing), a host computer (online processing in 3211 and 4245, XPAF, or XPMF-VMS modes or communication through the Ethernet). For information on XPAF, refer to *Printing Documents with XPAF User Reference*. For information on XPMF-VMS, refer to the *Xerox Print Management Facility—VMS version User Guide*. For data sources on XNS, refer to *Printing Protocol*. The input-related concepts for online and offline processing are presented in the following sections.

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: remote dial up communication interfaces, an Ethernet communication interface, and SNA/SDLC input through the host interface processor (HIP).

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 360 or 370, system 3000 or 4300 series, 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 an IBM 3211 printer and 3811 printer Control Unit is already supported on the host system.

A channel-attached LPS can receive input from the host computer in online 3211 or 4145 mode. In online 3211 mode, the data stream emulates IBM 3211 line printer format, with or without embedded DJDEs.

Online 3211/4245 mode

Basic LPS input processing functions are augmented by the following functions for online 3211 or 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 or 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.

Online-specific commands

This section explains how to create a JDE or JDL, DJDE processing, online optimization and copy-sensitive copy modification entries (CME).

Creating a JDE or 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 other entries. The one PDL command that must be specified for normal online processing is `VOLUME HOST=IBMONL`. The `HOST` parameter of the `VOLUME` command 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` commands.

The defaults which are automatically selected when `VOLUME HOST=IBMONL` is specified are as follows:

```
RECORD  LENGTH = 150
LINE    PCCTYPE = IBM3211 or IBM4245
        DATA = (0,150)
```

If ANSI carriage control commands are used in the host application program, the host operating system converts each command to a valid 3211 or 4245 command before sending it to the printer. Any commands other than those listed are rejected by the LPS online interface (and by the 3211/4245). Therefore, the only appropriate PCCTYPES for an online JSL file are IBM3211 or IBM4245. Figure 4-4 shows a sample online JDL file that includes the PCCTYPE parameter.

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` parameter of the `LINE` command has a default offset of zero rather than one.

Figure 4-4. 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=IGNORE;
          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 PAGES
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,HCOUNT=1;
POWER: JOB;
      BANNER  TEST=C5,HCOUNT=2;
/*
      END;

```

DJDE processing

Printing is controlled through parameters from the user-defined JDE file which may be dynamically modified by dynamic job descriptor entries. DJDEs are user created and are processed by the LPS as part of the print data from the host machine. The chapter "Specifying dynamic job descriptor entries (DJDEs)" 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.

Online optimization

The OPTIMIZE parameter of the VOLUME command allows the user to select options for enhancing online throughput.

Copy-sensitive copy modification entries (CME)

Copy modification entries (CMEs) which are copy sensitive (that is, a copy range has been specified on the MODIFY parameter of the OUTPUT command) 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 parameter of the OUTPUT command in the "Specifying output parameters" chapter.

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 logical processing commands which can be used to control online report separation are BANNER and RSTACK. Refer to the chapter "Using logical processing commands."

Universal character set buffers (UCSBs)

The IBM 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 parameter of the LINE command. 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 parameter 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, however, a report using lowercase characters is printed using a font without lowercase, the user may:

- Switch to UCSB=PROCESS in a selected JDE, and
- 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 command in the JDE file.

UCSB processing

When a UCSB is transmitted from the host, it is saved on a PFILR\$.SYS file. When system generation is executed, the system looks for the UCSB saved in the file. If the system does not find the UCSB, it reinitializes the file 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.

Forms control buffer (FCB)

The 3211/4245 forms control buffer (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 parameter of the LINE command.

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. However, you 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 IBM 3211/4245. 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 parameter of the VOLUME command which provides for online buffering for improved performance.

To select a record length longer than 150 bytes, the user must specify the new value as the LENGTH parameter on the RECORD command and must also modify the DATA parameter of the LINE command to print the extended character or record length.

Points to note

Note 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.
This action is consistent with the IBM 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 parameter in the RECORD command cannot be changed using a DJDE since this would require dynamic rebuffering. Therefore, you should consider modifications to the default online JDL to simplify operations in the online environment.
- The OPTIMIZE parameter of the VOLUME command can be changed through the DJDE JDE= subject to the restrictions listed in OPTIMIZE.

Online recovery

In the event of a system failure, data that is formatted for print and stored on 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 in the VOLUME command OPTIMIZE parameter and DJDEs that 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. Therefore, 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.



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 allows you 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 terminated by either the operator ENDJOB command or the offline operator command. 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.

The HEADER segment consists of six fields, shown in table 4-63.

Table 4-63. Header segment fields

Segment	Definition
HOST COMMAND	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 and TIME	Date and 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 command. 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 operator ENDJOB command must be entered to end the dump session.
- 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, or 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 operator 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 operator SAMPLE command with no parameters.
- Operator "HIP START ..." command is accepted, but HIP is 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.

Three methods of downloading files are available:

- HOSTCOPY utility
- DJDE FILE command
- LPS and host file transfer

The HOSTCOPY utility requires the system to be placed offline and idle, then rebooted from disk to enter the operator 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; however, 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 and 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, \$\$\$, and TSK. In addition, card-image format files are restricted to destination file types CMD, FSL, JSL, MSC, PCH, TMP, and TPF.

DJDE FILE command

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 parameter that is used to signal the start of the file data.

The FILE DJDE is record oriented and is applied immediately. It can be included with other record-oriented or page-oriented DJDE commands in a DJDE packet. Refer to the 4050/4090/4650 LPS Command Reference for additional information on card-image file processing.



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.

Offline mode

For offline operations, the input medium to the printing system is magnetic tape, which may be recorded in one of a variety of standard vendor formats. As a programmer, you define the tape input deblocking and record format parameters that reduce physical tape blocks first to logical records, then to print lines. Special processing commands can also be selected that facilitate report processing by providing for logical functions to be performed on the input data described later in this chapter.

Before selecting the PDL commands that are to describe a specific job tape, certain tape structure concepts must be understood. These concepts, discussed in this chapter, enable the PDL programmer to define job tape characteristics such as host format, block/record structure, tape translation code, packed data formats, and several other parameters.

Host computer tape formats

The printing system processes input tapes produced by those standard hosts listed in the "Host system JDLs on the system software tape" section in appendix D, "Offline specifications," of this manual. A specific input tape is selected by the HOST parameter of the VOLUME command. The format of each tape is described in the *Xerox LPS Tape Formats Manual*.

Tape codes

Tape codes recognized by the printing system are EBCDIC, ASCII, and several versions of BCD. If these codes are not sufficient for a particular tape, the user may create a new code translation table or modify (customize) an existing code translation table. Tables showing the correspondence between standard recording codes and printed characters are contained in appendix C, "Character code assignment tables." Within a JDL, tape codes are selected by the CODE parameter of the VOLUME command.

Packed data formats

Six-bit characters may be written onto a 9-track tape in a 4-by-3 packed (or compressed) format. That is, four 6-bit data bytes are compressed into three 8-bit data bytes. Two methods of packing these bits together exist. One method is used by Honeywell 6000 users (T4X3), while Honeywell 2000 users employ a slightly different method of packing (T4X3H2).

Whenever an unpacking method is included in the JDL, the system unpacks the characters before data processing. Each 6-bit character is extracted, and two high-order zeros are appended. Normally, after data is unpacked, it must be translated. The character code set is defined in the CODE parameter of the VOLUME command. For a 4-by-3 unpacking method, the data is generally encoded in BCD, and one of the three standard BCD CODE parameters (H2BCD, H6BCD, and IBMBCD) can be used.

The “Offline specifications” appendix shows an example of how 6-bit characters packed in the T4X3 method (also T4X3H2) are unpacked and then translated to ASCII by the system.

Record formats

All tape records input to the printing system are either blocked or unblocked with a fixed length, a variable length, or an undefined format. The PDL commands BLOCK and RECORD define the format of the input data. Tape label contents may also describe blocking and record structure and in some cases override BLOCK and RECORD commands specified in the JDL source file. These labels are described in this chapter and in the *Xerox LPS Tape Formats Manual*.

Record structure

A record is arbitrarily divided into two portions: operating system and user. The operating system portion of the record contains information supplied by the host operating or spooling system. The user's portion of the record contains information provided by the application or user's program running on the host system. The boundary between the two portions of the record is traditionally between the record length and printer carriage control (PCC) field. If there is no record length field, there is no operating system portion of the record. The PDL commands that define the components of a record are described in the “Input record characteristics—RECORD command” section of this chapter.

Multivolume processing

All multivolume reports that force input to make multiple passes over the data are handled in the following manner. For each copy requested, a complete pass is made over the group of volumes that make up the current report. The input processor is forced to make multiple passes over the input data for multicopy reports that specify copy-sensitive CMEs, and (optionally) for multicopy reports that exceed the size of the print file. CMEs are discussed in the “Copy modification entries” section in the chapter, “Specifying output parameters.” The choices available for handling print file saturation are discussed under the VOLUME command RSAT parameter in this chapter.

If a multivolume report requires multiple passes, messages appear on the system controller display with instructions on which action must be performed next.

Users with multiple volume jobs and multiple laser printing systems may wish to print each volume on a separate system. This requires that each volume be processed independently. Refer to the EOVS parameter for a discussion of this processing.

In the event of a system rollover while processing a multivolume report, input (as a portion of the recovery process) displays the following message, before performing a forward tape space to the recovery point:

```
ENTER 'CON I' WHEN VOLUME n IS MOUNTED
```

As before, volume 1 is the start-of-job, and the volume specified by *n* is the volume containing data for the last page written to the print file before the system rollover.

5. Defining clusters

The Xerox DocuPrint 96/DocuPrint 96MX LPS provides several methods for feeding paper. There are internal trays (trays 1-3) and a bulk input enablement (tray 7).

Each of the input source trays can be addressed and used individually. They can also be addressed in groups, such as pairs, or groups of three and used in groups as clusters.

Cluster features

Clusters are a systematic method provided for feeding specific paper stock to print jobs. 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 trays as they become empty or as page size requirements change.
- 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.
- Trays actually loaded with the stock are the "current" trays.
- Operator can override the preferred tray recommendation.
- Data stream can specify the cluster used to feed a page.
- 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 processes the job in two steps:

- Identifying the stocks requested by the application
- Selecting a feeder tray for each page by checking the cluster name and paper size associated with the page.

What clusters do for the programmer and operator

As the programmer, you group any combination of trays into a cluster. Group the trays together because they hold the same stock or they hold stock which differs only in size. You can specify the cluster's paper stock by any combination of weight, color, size, and so on. Use clusters to include instructions within the report identifying:

- Clusters needed by the report, that is, the report "stockset"
- Particular cluster to be used to print each particular page
- Recommended trays for loading the stock.

Then you store the defined clusters in the CLUSTR.LIB file through the cluster (CLP or CLU) commands described in your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*. 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 also define cluster groups as "stocksets." The stocksets can also be called by all jobs run on the system.

The operator uses clusters to decrease how often you load stocks. The operator verifies job and cluster status and overrides tray specifications for clusters through the cluster (CLU) commands described in your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*. The cluster definitions inform you which stocks to load in which trays for queued jobs.

When an operator loads 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

The print description language OUTPUT command FEED parameter is valid with FEED=MAIN and FEED=AUX because the system defines the MAIN and AUX trays as clusters to support backwards compatibility. Both clusters are always active regardless of whether they are specified in the active stockset.

Note the following:

- Do not, however, use FEED=MAIN and FEED=AUX in conjunction with stocksets.
- You must use the FEED=MAIN operator command at the start of a job if the data stream is controlling trays, cluster information, and so on.
- 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

There are several ways for you to specify clusters in your applications:

- 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 the STOCKSET command (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 and stockset definitions and changing MAIN and AUX to logicals for each cluster as required by the jobs.

The OTEXT parameter of the MESSAGE command is described in detail in the chapter “Specifying output parameters.”

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.

Stockset changes require the cluster checks described in table 5-1.

Table 5-1. **Cluster checks following stockset changes**

Stockset change	Cluster checks required
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? Refer to your <i>Xerox DocuPrint 96/DocuPrint 96MX LPS Operator Guide</i> for more information.

A cluster without these errors has an "in use" status, that is, 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.

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 an operator enters FEED commands that override the data, for example, the FEED=DJDE, then the printer cycles down and displays:

```
OS2289 Warning: Current FEED command overrides data
FEED= value.
```

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

The PDL commands and DJDEs listed here relate to clusters. For more detailed information on each command, refer to the chapters “Specifying output parameters” and “Specifying dynamic job descriptor entries (DJDEs).”

PDL commands

The STOCKSET and OUTPUT commands specify cluster and stocks in your JSLs.

The STOCKSET and OUTPUT commands and their syntax are shown in tables 5-2 and 5-3.

Table 5-2. **STOCKSET parameter(s) and option(s)**

Parameter(s)	Option(s)
ASSIGN =	{ <i>stock-descriptor</i> (<i>stock-descriptor</i> ₁ [, <i>stock-descriptor</i> ₂][,...])}
INIFEED =	{ <i>stock-name</i> <i>stock-reference</i> }
SYSPAGE =	{ <i>stock-name</i> <i>stock-reference</i> MAIN AUX}

Table 5-3. **OUTPUT parameter(s) and option(s)**

Parameter(s)	Option(s)
FEED =	{ <i>stock-reference</i> (<i>stock-name</i> MAIN AUX OPR)}
STOCKS =	<i>stockset-name</i>

DJDEs

The FEED and STOCKS dynamic job descriptor entries change stocks and clusters on a page-by-page basis.

The FEED and STOCKS commands and their syntax are shown in table 5-4.

Table 5-4. **STOCKSET parameter(s) and option(s)**

Parameter(s)	Option(s)
FEED =	{ <i>stock-reference</i> <i>stock-name</i> MAIN AUX OPR}
STOCKS =	<i>stockset-name</i>

Points to note

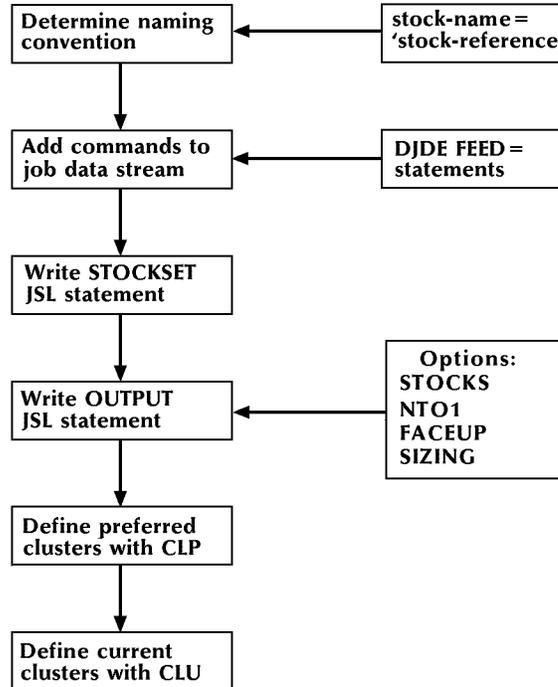
Note the following when defining clusters and stocksets with PDL and DJDE commands:

- 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.
- 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.

Steps for creating clusters

The steps you should take to create a cluster are listed below. Figure 5-1 provides a visual representation of these steps.

Figure 5-1. Steps for creating clusters



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 instance, COVER in the example below). Associate each cluster reference with a cluster name easily recognized by the operator as a particular stock (for instance, CV102 in the example below).

```

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. This way, you avoid modifying the data stream or the application. The cluster reference in JDL files and PDE files thus remains valid.

- Step 4.** Use the FEED= DJDE in the JSL file to specify the cluster to be used. For example:

```

...
...data for cover letter...
DJDE FEED=BILPAG,END;
...data for bill...
DJDE FEED=SUMMARY,END;
...data for summary sheet
(and in the JSL)
OUTPUT STOCKS=BILLS, FEED=COVER;

```



Note: The specified cluster must be identified in the active stockset.

If clusters are used in the print job, the OUTPUT command requires the STOCKS= parameter; the FEED= parameter is optional. The INIFEED= parameter of the STOCKSET command takes effect when no FEED= parameter is coded. If the STOCKSET command has no INIFEED= parameter, the first cluster specified in the STOCKSET command is used. The following example produces the same results as the previous example.

```

BILLS: STOCKSET ASSIGN=( 'CV102' , COVER) ,
                ASSIGN=( 'BL231' , BILPAG) ,
                ASSIGN=( 'SM233' , SUMMARY) ,
                INIFEED=COVER;
        OUTPUT  STOCKS=BILLS;

```



Note: The STOCKS=DJDE makes the STOCKSET command INIFEED= parameter take effect, thereby canceling the previous FEED= parameter.

The STOCKSET command has an optional SYSPAGE=parameter. 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' , SUMMARY) ,
                ASSIGN=( 'RED00' , RED) ,
                INIFEED=COVER,
                SYSPAGE=RED;
        OUTPUT  STOCKS=BILLS;

```

Keeping stockset changes to a minimum

Each formatted page is associated with the active stockset and the active FEED= option stock identified by the INIFEED parameter of the STOCKSET command, the FEED parameter of the OUTPUT command, 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= specifications, but only one stockset change. Report boundaries are not treated as stockset changes unless use of the stockset specified in a JDE or 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 or 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 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.

6. Print format commands

Print format commands serve many different functions, the most common of which specify the physical characteristics of a print job, define the placement of data on the page, and identify system responses to error conditions.

Table 6-1 summarizes these PDL commands. The function, syntax, and usage of the parameters available with these commands are described in individual sections of this chapter.

Table 6-1. Summary of print format commands

Command	Function
ABNORMAL	Specifies operations security and page offset on jam recovery.
ACCT	Defines user and system accounting requirements.
CME	Allows certain parts of report output to be replaced on copies with predefined static data or to specify font changes within variable data.
EXPORT	Directs printing of segments and segment recovery.
LINE	Controls margin, overprinting, carriage control, and so forth.
MESSAGE	Sends message to operator during input or output processing.
OUTPUT	Controls printing mode, forms, paper size and type, offsetting and output destination.
PDE	Defines page format (page orientation, fonts).
ROUTE	Sends message and form for a routing page.
STOCKSET	Defines a set of stocks used in a report.
VFU	Defines vertical format control.

ABNORMAL command

The ABNORMAL command allows you to restrict certain operator functions and define system responses to error conditions. Table 6-2 summarizes ABNORMAL command parameter functions.

Table 6-2 summarizes the ABNORMAL command parameters.

Table 6-2. **Summary of BLOCK command parameters**

Parameter	Specifies	Offline	Online	DJDE
ERROR	System response to abnormal conditions detected while processing or compiling DJDEs	Y	Y	N
IMISMATCH	The action to take when a specified ink requires a primary that is not currently loaded	Y	Y	N
ISUBSTITUTE	Specifies whether operator-initiated ink substitution is allowed	Y	Y	N
OTEXT	Optional WAIT message following recovery	Y	Y	N
PAGES	Specifies the maximum number of successive simplex pages on which recovery to the page boundary is to be attempted	Y	Y	N
REP	Page offset and/or slip-sheet after recovery	Y	Y	N
SECURITY	Restricted use of certain operator commands	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ERROR

This parameter specifies the required system response to abnormal conditions detected in input while processing or compiling DJDEs.

Syntax ABNORMAL ERROR = *option(s)*

Options Table 6-3 lists the parameter option(s) and definition(s).

Table 6-3. **ERROR parameter option(s) and definition(s)**

Option(s)	Definition(s)
CONTINUE	Displays the normal system message defining the abnormal condition and continues processing the report without applying the offending DJDE parameter and without offering the operator the parameter of continuing or aborting the job.
ABORT	Displays the message for the abnormal condition and aborts only the report it is processing (not the entire job) without offering the operator the parameter of continuing or aborting the job.
STOP	The default for this parameter and indicates that the system stop processing the report and wait for the operator to respond with a CONTINUE or ABORT command. The default is STOP.

Considerations The CONTINUE and ABORT parameters are referred to as non-stop modes. The STOP parameter is the stop mode. Any abnormal condition detected, such as syntax error in DJDE specifications, missing file referred to by a DJDE, RTEXT error, or GRAPHIC error in either mode results in an appropriate error message being displayed. In addition, during the stop mode, the displayed error message is followed by a DJDE error message:

```
OS2006 ENTER CONTINUE I OR ABORT I
```

If abnormal conditions are detected while processing or compiling DJDEs, the DJDE parameter set containing the offending entry is set up to print on the OPRINFO page and an error message is displayed. Errors detected while processing RTEXT data are not reported to the user. The graphics processing errors are reported separately on the graphics exception page printed at the end of the report.

IMISMATCH

This parameter indicates the action to take when a specified ink requires a primary that is not currently loaded.

An ink mismatch occurs when a report requests an ink that requires a primary color not currently loaded. It can occur within a report and may be detected after printing has started.

If the creator of a report did not include an ink list in the JDE for a report, printing may start before the system is aware of the primary colors needed. For example, page 100 of a report, which was designated black-only calls for the green primary and the ink currently loaded in the printer is red.

Syntax ABNORMAL IMISMATCH = *option(s)*

Options Table 6-4 lists the parameter option(s) and definition(s).

Table 6-4. **IMISMATCH parameter option(s) and definition(s)**

Option(s)	Definition(s)
STOP	<p>Prompts the operator to load the requested primary or abort the report or job. If SUBSTITUTE=ANY is specified for the report, the operator may also enter a SUBSTITUTE INK command followed by the CONTINUE command to resume printing.</p> <p>The appropriate action for an operator to take for an ink mismatch condition when IMISMATCH=STOP is specified depends on a number of factors:</p> <ul style="list-style-type: none"> • If the report has not been sent for Output processing yet, the operator can enter ABORT I to abort report Input processing. However, if the operator then enters the CONTINUE I command, the report prints with black ink in place of the missing ink. If this is not the desired result, the operator should enter ABORT I and ABORT O, followed by the CONTINUE command. • If the report is already being processed for printing, the system responds to the mismatch condition by displaying the following messages: <pre>OS6213 X HOUSING IS NEEDED FOR THIS REPORT OS2682 CHANGE HOUSING FOR PRINTING IN REQUESTED INK OR OS2000 ABORT THE REPORT OR PRINT JOB OS2000 ENTER 'CONTINUE O' TO RESUME PRINTING</pre> <p>When this series of messages appears, the operator has three choices:</p> <ul style="list-style-type: none"> • Change the color housing and enter CONTINUE O • Abort the report and enter CONTINUE O. If the operator elects to abort the report, there are two possible methods: <ul style="list-style-type: none"> — Enter ABORT O — Enter CANCEL ENTRY #<i>entry-string</i>, where <i>entry-string</i> is the number or numbers of the reports in the queue to be cancelled. • Abort the entire job that contains the report by entering ABORT job-id and CONTINUE O. Refer to the <i>Xerox 4850/4890 HighLight Color LPS Command Reference</i> for information on these commands. <p>The default is STOP.</p>
ABORT	Printing of the report is aborted.
CONTINUE	Uses the currently loaded primary regardless of the ISUBSTITUTE parameter or the FDL INKS command NOSUBSTITUTION parameter.

Example

```
ABNORMAL IMISMATCH=CONTINUE ;
IDR      ILIST=( 'BLACK' , 'BLUE' ) ;
```

In this example, if 'RED' is currently loaded and ready, the report is printed in red with no operator intervention.

Considerations Ink mismatch errors are not handled by the ERROR parameter. Ink mismatch is a special type of error that requires specific operations and is processed independently of ERROR parameter specifications.

Refer to the "Coding for efficient queue management" section in the "Highlight color printing" chapter for information on the relationship between queue management and the IMISMATCH parameter.

ISUBSTITUTE

This parameter specifies whether operator-initiated ink substitution is allowed.

Syntax ABNORMAL ISUBSTITUTE = *option(s)*

Options Table 6-5 lists the parameter option(s) and definition(s).

Table 6-5. **ISUBSTITUTE parameter option(s) and definition(s)**

Option(s)	Definition(s)
ANY	Any operator-initiated substitution is allowed. This is the default. The default is ANY.
NONE	No operator-initiated substitution is allowed. The report is either printed without operator-initiated substitution or aborted by the operator.

Example ABNORMAL ISUBSTITUTE=NONE ;

Considerations Refer to the “Efficient coding for queue management” section in the “Highlight color printing” chapter for information on the relationship between queue management and the ISUBSTITUTE parameter.

If the ISUBSTITUTE parameter specifies NONE, the printer cycles down and displays messages on the system controller screen. These messages indicate that a print quality adjustment may be performed before continuing to print the job or report with the requested primary, or that the job or report is being aborted.

OTEXT

This parameter specifies that the system is to stop and display a WAIT message when the recovery marker page is printed. This delay allows the operator to verify the output report in the near vicinity of the marker page. The default is NOWAIT, which means that the message is still displayed upon printing of the marker page, but printing does not stop.

Syntax ABNORMAL OTEXT = *option(s)*

Options Table 6-6 lists the parameter option(s) and definition(s).

Table 6-6. **OTEXT parameter option(s) and definition(s)**

Option(s)	Definition(s)
WAIT	Displays a WAIT message.
NOWAIT	Displays a WAIT message but printing does not stop. The default is NOWAIT.

Example ABNORMAL OTEXT=WAIT,
ERROR =ABORT;

PAGES

This parameter specifies the maximum number of successive simplex pages on which recovery to the page boundary is to be attempted.

Syntax ABNORMAL PAGES = *option(s)*

Options Table 6-7 lists the parameter option(s) and definition(s).

Table 6-7. **PAGES parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>n</i>	For duplex printing, <i>n</i> is the number of sheets or front-side, back-side pairs of pages to be attempted. The default is 0.

REP

This parameter specifies that the first data page delivered to the bin following a printer jam recovery or a system crash may be offset from the rest of the report.

Syntax ABNORMAL REP = *option(s)*

Options Table 6-8 lists the parameter option(s) and definition(s).

Table 6-8. **REP parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Specifies that the page will not be offset.
YES	Specifies that the page will be offset.
AUX	Specifies that a blank sheet, picked from the auxiliary tray, is to be placed as a slip-sheet before any data pages printed after a recovery from a jam, crash, or potential output integrity problem.

If the **YES, AUX** option is specified, the AUX sheet is the sheet that is offset (and not the first data sheet).

SECURITY

This parameter specifies that certain operator functions are restricted.

Syntax ABNORMAL SECURITY = *option(s)*

Options Table 6-9 lists the parameter option(s) and definition(s).

Table 6-9. **SECURITY parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies the following system level parameters (as described in the <i>Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference</i>): <ul style="list-style-type: none"> • SAMPLE (sample print) • MOVE BLOCKS (block spacing of tape input) • SPACE PAGES (skipping over output pages).
NO	Specifies that none of the above restrictions are in effect. The default is NO.

If SECURITY is invoked in a selected JDE, such as a DJDE JDE, the SAMPLE and SPACE PAGES restrictions do not take effect until after the first page of the report has been imaged (or spaced past).

Recovery

The automatic recovery procedures of the LPS are based on the availability of valid checkpoint data independently saved by both the output processor and the input processor. As a result, most recoveries are performed with no operator assistance. In certain cases, the system may detect that the checkpoint data is invalid, and operator assistance is requested, that is, the operator or user is requested to mount a preceding tape volume (offline) or return to point of origin or last valid checkpoint (online).

Marker page

The system produces a marker page (refer to figure 6-1) as a function of the input processor and inserts it into the print job at the point of crash. It is printed after the output processor has printed data that was already processed by input before the crash occurred.



Note: Actual size is 8.5 by 11 inches, and the squares at the top and bottom of the page are black.

Figure 6-1. Sample marker page



```
*****RECOVERY MARKER PAGE RECOVERY MARKER*****  
*E*  
*C*  
*O*  
*V*  
*E*  
*R*  
*Y*  
*  
*M*  
*A*  
*R*  
*K*  
*E*  
*****RECOVERY MARKER PAGE RECOVERY MARKER*****  
CRASHING TASK: INPUT  
CRASH CODE: BPT  
DATE: 26 APR 85, 18:25  
JOB NUMBER: 2565  
REPORT NUMBER: 1
```



ACCT command

The system maintains report processing information in two files which are accessible to the user. Both files can be saved on tape or transmitted to the host through the host file transfer facility for further processing. One method accumulates overall usage statistics on a department or JDL basis. Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for further information on the accounting file printout, initialization, and structure. In the second method, the status file accumulates report information, in the order that it is processed, on an individual report basis. Refer to the system level parameters JOBS and SFS and the "Report status information" section in your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for further information on the status file display, initialization, and structure.

Accounting information is automatically accumulated by the system on a report basis and also for overall system usage. The ACCT command enables the user to include a printout of the accounting summary with each report printed. Table 6-3 summarizes ACCT command parameter usage. The system level REPORT USER parameter provides an installation with a printout of overall system usage statistics. The following sections define how accounting information may be accessed and interpreted.

Table 6-10 summarizes the ACCT command parameters.

Table 6-10. **Summary of ACCT command parameters**

Parameter	Specifies	Offline	Online	DJDE
DEPT	Accumulation of user accounting statistics	Y	Y	Y
USER	Output destination of accounting page with a report	Y	Y	N

An example of an offline accounting page is illustrated in figure 6-2.

Accounting data on report basis

The ACCT command enables you to request that an accounting page which contains a summary be output at the end of the printing for each report. This summary consists of a single page of information containing job setup information and counts of processing events.

Overall usage accounting

The system automatically accumulates accounting data in a file on disk for each report printed. An installation has the parameter of having the system accumulate this information under a specified name, such as department code, account, or accumulate it under a system default name.

The DEPT parameter of the ACCT command allows an installation to define a name under which the accounting data is accumulated. The system ACCOUNT command (refer to the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*) must also be used with the DEPT parameter to create a name in the accounting file which corresponds to the name specified in the DEPT specification. It is necessary to enter this name only once with the ACCOUNT parameter, but it must be done before a print job is run using the DEPT name. If the name is not entered with the ACCOUNT parameter, the accounting data for this report is accumulated in the NODEPT entry in the accounting log. However, the DEPARTMENT entry on the accounting page for this report lists the requested department name followed by an asterisk to denote that no specific accounting by this name was performed.

If the DEPT parameter is not specified, data is accumulated under a name made up of the identifier on the JDL SYSTEM (or JDL) parameter with the characters :JDL appended to it, assuming that this same name exists in the accounting file. Thus, if a JDL identifier is H2SYS, and no DEPT was specified, the default accounting name on the accounting page is H2SYS:JDL. Data is accumulated under this default name only if this name exists in the accounting file. If the name is not in the accounting file, the accounting information goes to NODEPT. ACCOUNT parameter use is described in the "Accounting file maintenance" section of your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*.

The following example illustrates the use of the PDL command ACCT and the system level ACCOUNT parameter.

When the system level parameter ACCOUNT ADD, OPERATIONS is entered at the keyboard display, the name OPERATIONS is added to the current list of names in the system accounting file.

The following PDL command requests the system to deliver one accounting page to the sample print tray after each report is printed. Each time a print job is run with the above command in the JDE, report accounting statistics are updated in the system accounting file under the name OPERATIONS:

```
ACCOUNT USER=TRAY, DEPT='OPERATIONS' ;
```

Restricting LPS usage to only those departments explicitly added to the accounting file is accomplished by deleting the NODEPT entry. If NODEPT is deleted, jobs are aborted unless a name has been entered in the accounting file that corresponds with the department name as specified on the ACCT command. The NODEPT entry must be deleted before any accounting data has been entered under this name or after the accounting file has been cleared.

Installation accounting report

The REPORT USER system level parameter is used to print a copy of system accumulated accounting statistics. It prints out accounting data for each name entered into the accounting file by the ACCOUNT parameter. Use of the REPORT USER parameter is discussed in the chapter on tracking system activity in the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*. A sample of the output is shown in figure 6-2.

DEPARTMENT/JDL entries as shown in figure 6-3 are generated by the OSS (NODEPT and TASKS\$:JDL) and entered by you with the ACCOUNT parameter. NODEPT is the name under which accounting statistics are accumulated for print jobs without name entries in the accounting file. TASKS\$:JDL is a name under which accounting is accumulated for printing done by system tasks such as FDL, PDL, and the Editor. OPERATIONS is a user-supplied DEPARTMENT/JDL entry (in figure 6-3).

Other entries on this report, such as PAGES and LINES, are defined the same as those on the end-of-report accounting page.

Figure 6-2. Sample accounting page printed with job

```

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 OCT 97   AT   19:35:32
DEPARMENT:     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
Lines Printed:        66
Online Idle:          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 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

```

Figure 6-3. Customer usage accounting report

DATE	10/27/97	USER ACCOUNTING REPORT	PAGE 1
DEPARTMENT/JDL		PRINTING SUMMARY	TIME SUMMARY HR, MIN
NODEPT		PAGES = 21 LINES = 2200 PAGES TO TRAY = 0	INPUT = 0 ,1 OUTPUT = 0 ,2
		PROCESSING SUMMARY	TAPE SUMMARY
		JOBS = 2 FILES = 2 REPORTS = 2	MOUNTS = 0 BLOCKS READ = 220 BLOCKS SKIPPED = 0
DEPARTMENT/JDL		PRINTING SUMMARY	TIME SUMMARY -HR, MIN
OPERATIONS		PAGES = 105 LINES = 1700 PAGES TO TRAY = 0	INPUT = 0 ,5 OUTPUT = 0 ,7
		PROCESSING SUMMARY	TAPE SUMMARY
		JOBS = 1 FILES = 7 REPORTS = 7	MOUNTS = 1 BLOCKS READ = 1100 BLOCKS SKIPPED = 0
DEPARTMENT/JDL		PRINTING SUMMARY	TIME SUMMARY HR, MIN
TASKS\$:JDL		PAGES = 3 LINES = 100 PAGES TO TRAY = 0	INPUT = ,0 OUTPUT = ,0
		PROCESSING SUMMARY	TAPE SUMMARY
		JOBS = 3 FILES = 3 REPORTS = 3	MOUNTS = 0 BLOCKS READ = 0 BLOCKS SKIPPED = 0

The following sections describe the syntax of the command parameters and explanations of the parameter options.

DEPT

This parameter defines the name under which accounting statistics should be accumulated for this report job.

Syntax ACCT DEPT = *option(s)*

Options Table 6-11 lists the parameter option(s) and definition(s).

Table 6-11. **DEPT parameter option(s) and definition(s)**

Option(s)	Definition(s)
sc	A string constant of up to 31 characters representing a department code or name under which accounting information is maintained. These 31 characters and including the character “:” are allowed for the department code. Refer to the “Overall usage accounting” section for further information on DEPT parameter usage. The sc name should already be created in the accounting file prior to processing the job. The default is the JDL name.

Example ACCT USER=BIN, DEPT='MARKETING' ;

USER

This parameter specifies destination of the one-page accounting summary for each report. This page is always printed in the mode of the report.

Syntax ACCT USER = *option(s)*

Options Table 6-12 lists the parameter option(s) and definition(s).

Table 6-12. **USER parameter option(s) and definition(s)**

Option(s)	Definition(s)
BIN	Directs an accounting page to the output bin. The default is BIN.
TRAY	Directs an accounting page to the sample tray.
BOTH	Directs an accounting page to the output bin and the sample tray.
NONE	Specifies that no accounting page is delivered.



Note: When running in duplex, line counts for the back sides of the last five (or fewer) pages of each report are not reflected in the system accumulated accounting statistics. **SELECT E** specifies that all accounting pages are forced to tray.

Points to note

Note the following when using the ACCT command:

- If the report completion code on the accounting page is non-zero, an accounting page is delivered to the tray even if USER=NONE is coded. This is done to notify the user of a possible report integrity problem.
However, if the report is duplex and the first integrity problem occurs within the last eight back sides, the report completion code on the forced accounting page is zero.
- Jobs aborted by input as a result of job recovery produce an accounting page with as much accounting information as can be salvaged. The accounting page is delivered to both the bin and the sample tray.
- When ACCT USER=BOTH is specified, the two accounting pages show different values for "Output Processing Time." The sheet sent to the stacker will show a time about one second longer than the sheet sent to the sample tray. All other values should match.
- Hexadecimal strings in a DEPT parameter are interpreted as standard EBCDIC and translated to Extended ASCII.
- A completion code of 0 indicates that the report may have completed successfully, except when a report integrity problem occurs within the last eight duplex pages.

Example

```
ACCT DEPT=' BUDGET ' ,USER=( BIN , TRAY ) ;
```

After each report is printed, an accounting page is delivered to the output bin and sample print tray. Accounting information for each job (run under this ACCT command) is accumulated in the system accounting log under the department name BUDGET. This assumes the name BUDGET has been entered into the list of accounting names (through the system level ACCOUNT parameter).

ac:CME command

The copy modification feature (also referred to as spot-carbon) allows certain parts of report output to be replaced on selected copies with predefined static data or to specify font changes within the variable data.

CMEs define a rectangular space on the printed page within which printed data is replaced with a substitution string or in which a font change occurs. More than one CME may be applied to a job. CMEs may be coded as part of the JDL or created as separate files so that they may be referenced by one or more JDLs. This latter feature is described more fully in the “Cataloged CMEs” section of this chapter.

An identifier (of the type ac) is specified when defining the CME and references it with the MODIFY parameter of the OUTPUT command. The CME must precede the OUTPUT command.

Table 6-13 summarizes the CME command parameters.

Table 6-13. **Summary of ac:CME command parameters**

Parameter	Specifies	Offline	Online	DJDE
CONSTANT	Copy modification character string to be printed	Y	Y	N
FONT	Selection of font for printing input or CME data	Y	Y	N
LINE	Line range of copy modification	Y	Y	N
POSITION	Initial character position of copy modification	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

Short form CME specifications

CME specifications may be given in short form to minimize the coding length. In short form, only the first character of a keyword need be given, equal signs are omitted, and commas are not inserted except where necessary to avoid ambiguity.

Cataloged CMEs

CMEs need not be part of a user's JDL. They may be created as separate disk files and used as if they were part of the JDL that references them. This is done by creating a JSL file containing only CMEs and using the PDL processor to compile it. For each CME, PDL creates an object file on disk, cataloged in the CME directory. The CME identifier is used as the object file name. When a CME source file is modified, it must be recompiled with PDL, but it is not necessary to recompile the associated JDLs.

When the CME is referenced by the OUTPUT command MODIFY parameter in a JDL and the CME does not exist within the JDL, the system searches the directory for the file type xxxx.CME. It is then loaded into memory for use in processing the report.

The DJDE MODIFY parameter can also be used dynamically to associate a cataloged CME file with report processing. CMEs must be cataloged as separate disk files if they are referenced by a DJDE MODIFY command.

CONSTANT

This parameter specifies the character string to print.

Syntax *ac:CME CONSTANT = option(s)*

Options Table 6-14 lists the parameter option(s) and definition(s).

Table 6-14. **CONSTANT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>sc</i>	Represents the string constant to be printed (refer to the chapter "Print Description Language (PDL)" for definition of string constants). The width of the copy modification rectangle is determined by the number of characters specified by <i>sc</i> . More than one <i>sc</i> is allowed. There is no default.

Considerations There is no default for the CONSTANT parameter of the CME. It must be specified unless all that is required is a font change in the variable data. Within a text string, the character # may be used as a lowercase toggle. When a text string is encountered, it is assumed that characters are to be inserted into the print line as they appear in the text string (in uppercase normally). If a # is encountered, the lowercase mode is invoked, and all letters after # are considered lowercase until another # is encountered. The sequence ## is used to indicate that the character # is to be inserted and is not to be treated as toggling lowercase mode.

FONTS

This parameter specifies an index into the font list (PDE command or DJDE) for font switching on input data or CME data. A PDE command is selected by the FORMAT parameter of the OUTPUT command.

Syntax *ac:CME FONTS = option(s)*

Options Table 6-15 lists the parameter option(s) and definition(s).

Table 6-15. **FONTS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>May range from 1 to <i>n</i>, where <i>n</i> is the number of different fonts specified by the FONTS parameter of the PDE command. A value of 1 specifies the first font in the FONTS parameter, 2 the second, and so forth. The default for a value is the current font (specified with the last FONTS parameter; initially the value is 1).</p> <p>There is no default.</p>

Considerations A font specification applies to input variable data as well as static CME data. If a line number (LINE) and character position (POSITION) but no insertion text (CONSTANT) are specified, the font change specified applies to input variable data at the position specified.

LINE

This parameter specifies the line range of the CME.

Syntax `ac:CME LINE = option(s)`

Options Table 6-16 lists the parameter option(s) and definition(s).

Table 6-16. **LINE parameter option(s) and definition(s)**

Option(s)	Definition(s)
{ <i>n</i> (<i>n,m</i>) (<i>n,-</i>)}	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>n</i> The initial line number of the copy modification rectangle • <i>m</i> The number of lines to repeat the information. If not specified, the information applies only to the starting line. • - A dash character indicates that information is to apply to all lines on a page beginning with the line indicated by <i>n</i>. <p>There is no default.</p>

Considerations The line numbers specified in the LINE parameter must always be in ascending order. In other words, you cannot specify one line number then list another one of lower value in the same command. For example, the following command is incorrect and will not be processed:

```
CME1: CME LINE=( 2 , 3 ) , POS=15 ,
        CONSTANT=( 6 ) 'HELLO' , FONTINDEX=3 ;
        LINE=1 , POS=15 ;
```

As you can see, the first line parameter specifies starting at line 2, while the second LINE parameter specifies line 1, which contradicts the ascending order.

POSITION

This parameter specifies the initial character position of the CME in the print line.

Syntax *ac:CME POSITION = option(s)*

Options Table 6-17 lists the parameter option(s) and definition(s).

Table 6-17. **POSITION parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>n</i>	<p>An integer value between 1 (the first position of the print line) and the value specified by the length right-part of the DATA= parameter in the LINE command.</p> <p>The default is 1.</p>

Points to note

Note the following when using the CME command:

- The CME LINE and POSITION parameters are order dependent in that LINE must always precede POSITION.
- **Data lines**
Data lines are not properly aligned if proportional fonts are being used in landscape mode and a switch is made to a larger font in the middle of the line. This restriction is a result of the functioning of the printer imaging hardware.
- **Multiple copies**
If a multicopy is aborted before all the pages are set up in the page log, one copy gets printed if the CMEs are copy-sensitive. Otherwise, all copies of the setup pages are printed.
- **OVERPRINT, FONTINDEX, and CMEs**
CMEs are not applied if OVERPRINT=MERGE and FONTINDEX are both specified.

CME parameters are applied to all lines within the range regardless of whether or not variable data appears on those lines.

- **Multiple lines and columns**

Multiple lines may be specified and multiple columns may be specified for each line. Multiple line specifications must be given in ascending (top to bottom of page) order. Multiple column specifications for a line range must be given in ascending (left to right) order. There also may be multiple text specifications following a column specification. These are combined to form a single text string. Font specifications may be specified at any point. The last font specified remains in effect until another font is specified. An example of a CME with these characteristics is as follows:

```
XYZ: CME
LINE=( 1 , 10 ) , POS=40 , FONT=2 , POS=80 , FONT=3 ,
LINE=( 11 , 20 ) , POS=1 , FONT=3 , POS=40 , FONT=1 , POS=80
      , FONT=2 ,
LINE=( 31 , 50 ) , POS=1 , FONT=2 , POS=40 , FONT=3 , POS=80
      , FONT=1 ;
```

- **Form and variable data**

Under certain circumstances, a form and its associated variable data do not align properly, even though it appears that both use the same line spacing. Variable data has a line-spacing value that is computed as dots per line, and rounding is done on that value. In FDL, the rounding takes place only when it is needed to resolve to a dot address and, therefore, might involve more than one line at a time. Refer to the "Rounding measurements" section in the *Xerox DocuPrint 96/DocuPrint 96MX LPS Forms Creation Guide* for further information.

- **CME string constant**

Whether a font switch is applied to a CME string constant depends upon the order in which FONT and CONSTANT are specified in the CME. If FONT is specified before CONSTANT, the font switch applies to the string constant. If FONT is specified after CONSTANT, the font switch occurs at the line position immediately after the string constant.

Where a CME string constant is printed on a line depends upon where the POSITION and CONSTANT parameters are specified in the CME. If POSITION is specified before CONSTANT, the string constant is printed at the specified position. If POSITION is specified after CONSTANT, the string constant is printed at the current position (position 1 if no other POSITION or CONSTANT parameter has been specified), and a subsequent CONSTANT or FONT parameter will take effect at the specified POSITION.

- **RSX11host type**

When the host type is RSX11, CMEs are often applied twice per line because of the way the carriage return/line feed is processed. When mixing font switches and string constants, the string constants may appear twice and be overprinted with two different fonts. The recommended approach for RSX11 host types is to specify the string constant font before the string constant itself, such as:

```
FONT=1 , CONSTANT='string' , FONT=2 ;
```

Examples

This section shows two examples of using the CME command.

Example 1

The following is an example of copy-sensitive CMEs.

```
CME1: CME LINE=3,POS=59,CONSTANT='FIRST QUARTER';  
CME2: CME LINE=(37,3),POS=81,CONSTANT=(6)'*';  
CME3: CME LINE=(1,60),POS=5,FONT=2,POS=12,FONT=1;
```

```
OUTPUT COPIES=3,MODIFY=(CME1,1,1),  
        MODIFY=(CME2,2,1),MODIFY=(CME3,3,1);
```

For the first copy of the report, line 3 is modified by the text 'FIRST QUARTER'. For the second copy, the character '*' is inserted in lines 37, 38, and 39 for 6 character positions (starting in column 81). For copy 3, fonts are changed for lines 1 to 60 at character positions 5 and 12. These fonts must be defined in the FONTS parameter of the PDE command.

Copy-sensitive CMEs are not supported for online processing or when COLLATE=NO is specified in the OUTPUT command.

Example 2

CME parameters specified in standard form:

```
CME12: CME LINE=47,POSITION=1,FONT=5,  
        CONSTANT=(5)'*'  
        LINE=48,POSITION=1,FONT=1;  
        LINE=49,POSITION=10,CONSTANT='ABCD' I2;
```

CME parameters specified in short form:

```
CME12: CME L47,P1,F5,(5)'*',L48,P1,F1,L49,P10,  
        C'ABCD';
```

or

```
CME12: CME L47P1F5(5)'*''L48P1F1L49P10'ABCD';
```

EXPORT command

A “segment” is a set of consecutively delivered sheets whose management as a set or segment has been specified by PDL or DJDE command.

The EXPORT statement enables specification in PDL of the new Segment Management capabilities, including the following:

- Division of reports into segments
- Generation of separator sheets
- Use of segment numbers on separator sheets
- Use of segment recovery in place of page recovery.

Although the EXPORT command is primarily intended to be used with the bypass transport option, it functions with any allowable destination. If the OUTPUT command DESTINATION parameter specifies BIN, segments are offset relative to each other when delivered, and offsetting resulting from other commands is not processed.

Table 6-18 summarizes the EXPORT command parameters.

Table 6-18. **Summary of EXPORT command parameters**

Parameter	Specifies	Offline	Online	DJDE
SEPARATORS	Retain an internal copy of the first data page and use it to generate segment separator sheets	Y	Y	Y
SNUMBER	Print the segment number on generated segment separator sheets	Y	Y	Y
SPLIT	The conditions for ending segments and turning segment management on	Y	Y	Y
SRECOVER	Recovery mode	Y	Y	Y
STIMING	This command is listed for compatibility only. It is not supported on the Xerox DocuPrint 96/DocuPrint 96MX LPS	Y	Y	Y

The following sections describe the syntax of the command parameters and explanations of the parameter options.

SEPARATORS

This parameter specifies that the printer software save an internal copy of the first data page of the report and use it to generate segment separator sheets.

Syntax EXPORT SEPARATORS = *option(s)*

Options Table 6-19 lists the parameter option(s) and definition(s).

Table 6-19. **SEPARATOR parameter option(s) and definition(s)**

Option(s)	Definition(s)
FIRST LAST (LAST,D) BOTH (BOTH,D) NONE	<p>This option has the following components:</p> <ul style="list-style-type: none"> • FIRST Specifies that a printer generated separator sheet be the first sheet of the second and subsequent segments. Note: The first segment begins with the first data page, so an additional sheet is not generated. • LAST Specifies that at least one copy of the saved data page should be generated (on a separate sheet of paper) as a hard copy segment separator and counted as the final sheet in the segment. Note: In an online environment, a true banner trailer page, if encountered, will be used to image any LAST separators of the final segment in place of the saved data page. • BOTH Specifies both FIRST and LAST, as above. • NONE Specifies that no segment separator sheets be generated. • D Specifies that the system prints the separator sheet on the back of any LAST sheets. If the report is simplex, the system still processes the last page of the segment as a duplex page. The parentheses are required when using the D parameter option. <p>The default is NONE.</p>

Example SEPARATORS=(BOTH , D) ;

SNUMBER

This parameter specifies that the printer merge the current segment sequence number (beginning with 001) into any separator sheets produced. The format of the printed segment sequence number for all separators except a LAST separator in the last segment is:

PART *nnn* (8 characters). *nnn* is a three digit number with leading zeros.

The format for the LAST separator in the last segment is: PART *nnn* of *nnn* (15 characters).

Syntax EXPORT SNUMBER = *option(s)*

Options Table 6-20 lists the parameter option(s) and definition(s).

Table 6-20. **SNUMBER parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>Inum</i> , <i>cnum</i> [, <i>findex</i>])	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>Inum</i> Specifies the integer line number on which the segment sequence number string is to start. • <i>cnum</i> Specifies the integer column number on which the segment sequence number string is to end. • <i>findex</i> Specifies the fontindex of the font to be used for printing the segment sequence number string. If not specified, the first font in the page's font list is used. <p>There is no default.</p>

Example SNUMBER=(1 , 75) ;

Considerations Note that *Inum* and *cnum* are interpreted using the metrics of the font specified by *findex* (the same as with page numbering).

Refer to the "Points to note" section in this chapter for additional information.

SPLIT

This parameter specifies the use of segment management, and is required in order for the other EXPORT parameters to work.

Syntax EXPORT SPLIT = *option(s)*

Options Table 6-21 lists the parameter option(s) and definition(s).

Table 6-21. **SPLIT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(min, max)</i>	<p>Specifies the minimum and maximum numbers of sheets, including any separator sheets, allowed in a segment. The minimum available number is 1 and the maximum number is 65,535.</p> <p>If a segment end has not occurred by the time max sheets are generated, then the printer declares end-of-segment. If end-of-segment occurs before min sheets are generated, then the printer generates padding of either blank or multiple LAST sheets (if LAST separators have been specified).</p> <p>There is no default.</p>

Example SPLIT=(2 , 7) ;

Refer to the "Points to note" section in this chapter for additional information.

SRECOVER

This parameter specifies the mode of jam, or error, recovery.

Syntax EXPORT SRECOVER = *option(s)*

Options Table 6-22 lists the parameter option(s) and definition(s).

Table 6-22. **SRECOVER parameter option(s) and definition(s)**

Option(s)	Definition(s)
PAGE	Specifies standard recovery. The printer resumes with the first sheet not successfully delivered. The default is PAGE.
SEGMENT	Specifies reprinting of the entire currently printing segment. Since it is possible for sheets of multiple segments to be in the paper path at the same time, the segment most advanced in the paper path without being successfully delivered is the starting recovery position.
ASK	Specifies that the operator be requested through the user interface to select PAGE or SEGMENT recovery.
DEVICE	Places recovery mode selection responsibility with a Device Finishing Architecture (DFA)-compliant finishing device through the LPS bypass transport option. If the DEVICE recovery option is specified and output is directed to a destination other than the bypass transport, the PAGE recovery option prevails.

Example SRECOVER=SEGMENT ;

STIMING

This parameter is provided for possible compatibility with other Xerox software supporting third party finishing devices. This parameter is ignored by the Xerox DocuPrint 96/DocuPrint 96MX LPS.

Syntax EXPORT STIMING = ([INTERVAL,*sec*][,DELAY,*sec*])

Points to note

Note the following when using the EXPORT command:

- **SNUMBER**

If you specify a line and column value for SNUMBER that is off the page, the segment number will not print. Furthermore, if the system encounters a valid SNUMBER dynamic job descriptor entry (DJDE) under this condition, unpredictable results may occur.

- **SPLIT = OFF**

Segment management processing is turned off when the LPS encounters a SPLIT=OFF dynamic job descriptor entry (DJDE). In addition, all segment management parameters are affected in the following ways:

- SRECOVER defaults to PAGE recovery
- SEPARATORS are forced to NONE.
- Segment numbering restarts to PART 001 if segment management processing is turned back on.

You must use the DJDE SPLIT = (min, max) to reinstate segment management processing after it has been disabled by the DJDE SPLIT=OFF.

- **SPLIT = NOW**

If segment processing is not in effect (that is, it either was not specified in the JDL or the DJDE SPLIT=OFF was invoked), the system ignores the DJDE SPLIT=NOW.

- **Ending a segment**

A segment may be ended under any of these conditions:

- The maximum allowable value is reached. This value is taken from the specified value in the SPLIT command.
- The end-of-report condition is met, including the logical end-of-report as specified in the RSTACK command.
- An end-of-data condition is met.
- SEPARATORS is specified in a DJDE.
- The system encounters a SPLIT=NOW dynamic job descriptor entry.
- The system encounters a SPLIT=OFF dynamic job descriptor entry.

- **Using a new separator page**

You may use a new separator page under these conditions:

- A SEPARATORS dynamic job descriptor entry (DJDE) is specified.
- An end-of-report condition is met, including the logical end-of-report as specified in the RSTACK command.

- **Resetting the segment number**

You may set a new segment number after the current one is terminated. This is accomplished when the system encounters a series of two DJDEs: SPLIT=OFF followed by SPLIT=(min,max).

In an online environment, banner pages are routed to the current job output destination. For example, if a job is being sent to a third party finishing device, the banner page will be sent there, also. The system does not redirect banner pages to the sample tray or output bin unless you specify that they be routed to a tray other than the job destination. Refer to the "BANNER command" section of the chapter "Using logical processing commands," for more information on specifying banner pages.

Examples

The following are sample EXPORT commands:

```
EXPORT SPLIT=( 2 , 7 ) , SNUMBER=( 1 , 75 ) ,  
SEPARATORS=NONE ;
```

```
EXPORT SPLIT=( 2 , 7 ) , SNUMBER=( 1 , 75 ) ,  
SEPARATORS=FIRST , SRECOVER=SEGMENT ;
```

LINE command

The LINE command provides parameters which allow the user to define the characteristics of the user portion of the input data record and how it is to be printed.

Table 6-23 summarizes the LINE command parameters.

Table 6-23. **Summary of LINE command parameters**

Parameter	Specifies	Offline	Online	DJDE
DATA	Input record printable data portion	Y	Y	Y
FCB	Host transmitted FCB handling	N	Y	N
FONTINDEX	Position of font switch index	Y	Y	Y
MARGIN	Left margin on a physical page	Y	Y	Y
OVERPRINT	Manner in which overprint lines are handled	Y	Y	Y
PCC	Location of the printer carriage control field	Y	N	N
PCCTYPE	Printer carriage control table	N	Y	N
UCSB	Host transmitted UCSB handling	N	Y	N
VFU	Vertical format table to be used	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

DATA

This parameter specifies the location and length of the print line data within an input data record.

Syntax LINE DATA = *option(s)*

Options Table 6-24 lists the parameter option(s) and definition(s).

Table 6-24. **DATA parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(pdo, length)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>pdo</i> (Print data offset) is the number of bytes between the start of the user portion of the logical record and the first character of the record to be printed. • <i>length</i> Specifies the maximum length of printable data within each logical record. For offline systems the default is (1,132). For online systems the default is (0,150). For HOST=RSX11(F LX), the DATA parameter must be included in the user's JSL since the default pdo value causes the first data byte to be skipped. <p>The default is (1,132) offline. The default is (0,150) online.</p>

FCB

This parameter specifies whether to suppress or accept the processing of the host transmitted forms control buffer (FCB).

Syntax LINE FCB = *option(s)*

Options Table 6-25 lists the parameter option(s) and definition(s).

Table 6-25. **FCB parameter option(s) and definition(s)**

Option(s)	Definition(s)
IGNORE	Suppresses the processing of the host transmitted forms control buffer (FCB).
PROCESS	<p>Accepts FCB processing. The VFU parameter defines the vertical upspacing if the FCB is ignored. For online systems only.</p> <p>The default is PROCESS.</p>

FONTINDEX

This parameter specifies that a field within a user data record defines the index to a specific font to be used for that line. If FONTINDEX is not specified, none is used.

Syntax LINE FONTINDEX = *option(s)*

Options Table 6-26 lists the parameter option(s) and definition(s).

Table 6-26. **FONTINDEX parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p>{<i>offset</i> (<i>offset</i> [<i>init-val</i>][<i>bit-opt</i>]) NONE}</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>offset</i> Indicates the byte offset in the data record where the font index number is to be found. • <i>init-val</i> Can be one of the following: ONE or ZERO. ONE specifies that an index value of 1 is associated with the first font in the font list. When the bit-opt parameter is used, the init-val ONE parameter must also be used. ZERO specifies that an index value of 0 is associated with the first font in the font list, an index value of 1 is associated with the second font in the font list, and so forth. The default is ONE. • <i>bit-opt</i> A numeral having a value in the range of 1 through 7, which specifies the number of low-order bits within the font index byte, which, in turn, specifies an index value into the font list of the current PDE. The default value is 4. <p>If <i>init-val</i> is ONE or is not specified, the value of the font index byte in the data record is a number in the range 1 to <i>n</i>, where <i>n</i> is the number of fonts specified in the PDE parameter (1 to 128). If ZERO is specified, the font index is a number in the range 0 to <i>n</i>, where <i>n</i> is the number of fonts specified in the PDE parameter minus one (0 to 127). The font index value must be present in every record. If the FONTINDEX in the data record is greater than the number of fonts specified, the first font in the list is used.</p> <ul style="list-style-type: none"> • NONE Specifies that there is no font index. <p>The default is NONE.</p> <p>Refer to “Points to note” later in this section for discussion of FONTINDEX usage with overprint lines and maximum number of fonts, and the examples which follow.</p>

MARGIN

This parameter specifies the left margin on a physical page.

Syntax LINE MARGIN = *option(s)*

Options Table 6-27 lists the parameter option(s) and definition(s).

Table 6-27. **MARGIN parameter option(s) and definition(s)**

Option(s)	Definition(s)
{ <i>value</i> (<i>value</i> , <i>value-type</i>)}	This option has the following components: <ul style="list-style-type: none"> • <i>value</i> The form <i>nnn.mm</i> (a positive decimal number with up to 2 digits to the right of the decimal point) which is the distance from the left margin. • <i>value-type</i> Indicates whether a value is specified in inches (IN), centimeters (CM), or character positions (POS). A value must be specified as an integer (<i>nnn</i>), if a value-type is character positions. The default is (1,POS). Refer to the "Points to note" for the VOLUME command in the chapter "Specifying input parameters" for additional information.

OVERPRINT

This parameter specifies the manner in which overprint lines are handled. (Overprint lines are print lines whose carriage control specifies printing with no line spacing since the last printed line.)

Syntax LINE OVERPRINT = *option(s)*

Options Table 6-28 lists the parameter option(s) and definition(s).

Table 6-28. **OVERPRINT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(over-opt, disp)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>over-opt</i> Options are: <ul style="list-style-type: none"> — PRINT — IGNORE — MERGE — PRINT2. <p>If IGNORE is coded, all overprint lines are ignored. PRINT specifies that all overprint lines are to be printed as they would be on an impact printer; that is, the second line is printed over the top of the first with no regard to the previous data, including character spacing, which may vary between the two lines of data. PRINT2 specifies that up to two consecutive lines are printed per line—one line and one overprint. Other overprints for the line are ignored. MERGE is the same as PRINT except when used with FONTINDEX or CME processing (refer to the “Points to note” section and example 2 of the “Examples” section).</p> • <i>disp</i> Options are: <ul style="list-style-type: none"> — DISP — NODISP <p>These are Xerox 1200 Computer Printing System parameters, which are preserved here for compatibility purposes. Neither is functional. The number of overprint lines is always printed on the accounting page.</p> <p>The default is (PRINT,NODISP).</p>

Example LINE OVERPRINT=(MERGE,NODISP),VFU=V1,
DATA=(1,132),
PCCTYPE=ANSI,PCC=(0,NOTRAN);

PCC

This parameter specifies the position (and possible translation) of the printer carriage control field.

Syntax LINE PCC = *option(s)*

Options Table 6-29 lists the parameter option(s) and definition(s).

Table 6-29. **PCC parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(offset, trans-type)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>offset</i> Specifies the byte offset of the user portion of the record within each logical record to the printer carriage control (PCC) field. 0 is the default. • <i>trans-type</i> Options NOTRAN/TRAN specify whether or not the printer carriage control byte is to undergo code translation. TRAN indicates that the byte is to be translated into LPS standard EBCDIC before being applied, using the translation defined in the CODE parameter of the VOLUME command. NOTRAN (the default) prohibits translation. <p>The default is (0,NOTRAN).</p>

PCCTYPE

This parameter specifies a carriage control set used in printing a job.

Syntax LINE PCCTYPE = *option(s)*

Options Table 6-30 lists the parameter option(s) and definition(s).

Table 6-30. **PCCTYPE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>pcc-name</i>	<p>Options are:</p> <ul style="list-style-type: none"> • ANSI • B2500 • B2700 • B3500 • B3700 • B4700 • B6700 • H6000 • H2000 • IBM4245 <p>Creation of a user-defined PCC table referenced by either an identifier <i>id</i> or the keyword USER is defined using the PCC command. Refer to the “PCC command” section in this chapter. For offline systems the default is <u>ANSI</u>. For online systems the default is <u>IBM3211</u>. Note that there is no functional difference between IBM3211 and IBM4245 for PCC types. For HOST=ICL2900 the PCCTYPE is forced to NONE.</p> <p>The INITIAL parameter (refer to the “PCC command” section in this chapter) for any selected PCCTYPE (except ANSI or HOST=RSX11, USER, and <i>id</i>) is TOF. For ANSI or HOST=RSX11, the INITIAL parameter is BOF. For USER or <i>id</i> the INITIAL parameter is set by the user in the PCC command.</p> <p>The ADVTAPE parameter (refer to PCC parameter in this chapter) for any selected PCCTYPE (except IBM1403, IBM3211, USER, and <i>id</i>) is YES. For IBM1403 and IBM3211/IBM4245, ADVTAPE is set to NO. For USER or <i>id</i> the ADVTAPE is set by the user in the PCC parameter. Refer to the <i>Xerox LPS Tape Formats Manual</i> for more information on PCCTYPE tables.</p> <p>The default is ANSI offline.</p> <p>The default is IBM3211/IBM4245 online.</p>

UCSB

This parameter specifies UCSB options.

Syntax LINE UCSB = *option(s)*

Options Table 6-31 lists the parameter option(s) and definition(s).

Table 6-31. **UCSB parameter option(s) and definition(s)**

Option(s)	Definition(s)
IGNORE	Suppresses the host transmitted universal character set buffer (UCSB). If IGNORE is coded, the CODE parameter of the VOLUME command defines the character translation. For online systems only.
PROCESS	Accepts UCSB for PROCESS, use of a FOLDED or UNFOLDED translate table is controlled from the host system. For UNFOLD, characters not defined as printable in the UCSB are converted to blanks. The default is PROCESS.

VFU

This parameter specifies VFU options.

Syntax LINE VFU = *option(s)*

Options Table 6-32 lists the parameter option(s) and definition(s).

Table 6-32. **VFU parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>vfu-id</i>	The command identifier of the VFU table, which must precede this reference to it. (Refer to the "Defining print line positions" section.) The VFU table defines print line positions corresponding to skip-to-channel parameters for the job to be processed.
NONE	Indicates that any skip-to-channel parameter is to be replaced by a carriage control of print and space 1. The default is NONE.

Points to note

Note the following when using the LINE command:

- **FONTINDEX**

Although the maximum bit-opt value may be 7 (allowing a maximum font index value of 127), the maximum number of fonts is constrained by the size of available input and output dynamic memory, font memory size, and the CME or PDE FONTS parameter. Also, internal table structures limit the number of fonts which can be invoked on a single page to 94 fonts.

- **FONTINDEX and OVERPRINT**

FONTINDEX, used in conjunction with overprinting, is handled as follows: For OVERPRINT=PRINT, the LPS overprints records analogously to an impact printer if the fonts are the same size. If the fonts differ, records are overprinted without regard to character spacing. For OVERPRINT=MERGE, the LPS replaces characters in the previous record which are blank. Character spacing values are adjusted; thus proportionally spaced or different size fonts may be used, and the LPS performs the character placement.

Characters in the previous record which are blank are replaced only when they are represented by an EBCDIC X40 or an ASCII X20. Binary character representations other than an EBCDIC X40 or an ASCII X20, which produce a blank character when printed with a particular font, is not replaced except in the following case. During input processing, when a binary character representation is greater than the highest character in the specified font, a blank (EBCDIC X40 or ASCII X20) is substituted into that print position. (A sample of the specified font shows the highest binary character representation.) In this case, the LPS replaces this character since it is now an EBCDIC X'40' or an ASCII X'20' blank.

If FONTINDEX and OVERPRINT=MERGE are specified and proportional fonts are being used, data lines are not properly aligned if printing in landscape mode, and the data switches to a larger font in the middle of the line.

- **Multiple fonts**

When multiple fonts of various sizes are used on a single page, the line spacing of any particular line is determined by the largest font in use in the previous line. The current line can also be adjusted downward by the difference in height between the first and the largest font in the current line. The line spacing of the first line in a report is determined by the first font in the PDE, provided the FONTINDEX parameter has not been specified. When FONTINDEX is specified, the line spacing of the first line of a report is determined by the font indicated by the font index value in the first line of the report. If there is no font index value in the record and FONTINDEX is specified, the line spacing for the first line of the report is determined by the first font in the PDE. When the FONTINDEX or CME parameters are specified, the line spacing of the first line of subsequent pages is determined by the line spacing of the font called out by the last font index of the previous page into the current PDL font list. If a PDE DJDE has occurred prior to the page transition, the font used for the line spacing comes from this list.

The height of the last font used determines the initial base line of that first print line.

The FONTINDEX byte, if selected, is processed for every record, DJDEs, RPAGE, and all criteria records. A valid FONTINDEX byte should be present in all records since it controls line spacing and override line spacing at page transitions.

- **Font index bytes**

It is advisable to put the font index byte in the beginning of a variable record to conserve tape space. The byte must be present in every record.

If the offset to the font index value exceeds the logical record size, the first font in the font list is used, except when HOST=UNIVAC is specified. If HOST=UNIVAC is specified, the byte located at the specified font index offset is processed. The font index byte must be located at the beginning of the record if Univac Fielddata and ASCII records are intermixed within the file.

- **Print line data**

When changing the print line data to a value greater than the default value (150), the LENGTH parameter of the RECORD command must also be changed accordingly. (Refer to the "RECORD command" section of the chapter "Specifying input parameters.")

- **PCCTYPE**

The PCCTYPE=IBM3211/IBM4245 is not intended to be used when processing offline jobs. Checks that the LPS online interface performs, such as for bad or NO-OP parameters, are not done in the offline mode. There is no check for bad or NO-OP parameters, and they are treated as undefined carriage control parameters in the offline mode and default to the carriage control parameter of print-and-space-1-line.

Examples

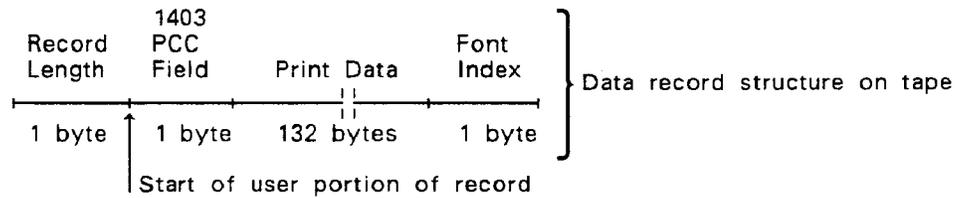
This section shows examples of using the LINE command.

Example 1

```
LINE DATA=(1,132),PCC=(0,NOTRAN),
      PCCTYPE=IBM1403,FONTINDEX=133;
```

The LINE command above defines the characteristics of the following record structure shown in figure 6-4.

Figure 6-4. **Sample data record structure**



The print-data-offset parameter of the DATA parameter (the number of bytes between the start of the user portion of the record and the first character of the record to be printed) is 1 byte. A sample is shown in figure 6-5. The print-length parameter (the number of characters in the longest print line in the record) is defined as 132 bytes. The carriage control is defined as IBM1403, and the position of the carriage control character is the first byte within the user portion of the record. The position of the font index (an index value into a font list that indicates which font is to be selected) is byte 133 (relative to 0).

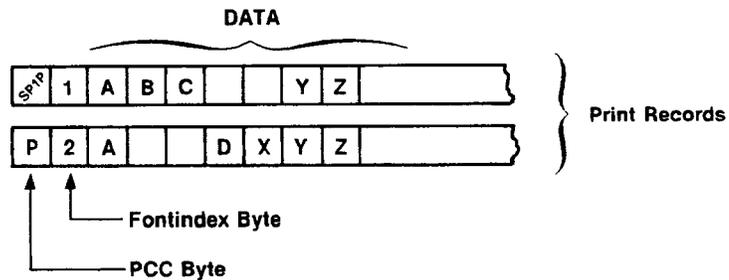
Example 2

```
LINE PCCTYPE=NONE, DATA=(1,60), VFU=V1;
```

Example 3

LINE PCCTYPE=NONE, DATA=(1,57), VFU=V1;

Figure 6-5. Sample print-data-offset parameter of the DATA parameter



Font 1	{	. A . B . C Y . Z	
Font 2	{	. A D . X . Y . Z	No Overprinting
		A B . C Y . Z . D . X . Y . Z	Overprinted Using PRINT
		. A . B . C . D . X . Y . Z	Overprinted Using MERGE

MESSAGE command

The MESSAGE and ROUTE commands permit the user to inform the operator of special conditions. The MESSAGE command displays user-defined text to the operator.

Example The following MESSAGE command informs the operator that blue paper is required for copy 2 of a 4 copy report. Printing is suspended at the appropriate points so that the operator can load the paper.

```
OUTPUT    COPIES = 4;
MESSAGE  ITEXT = ('COPY 2 WILL NEED BLUE PAPER'),
         OTEXT = ('LOAD BLUE PAPER', 2, WAIT),
         OTEXT = ('LOAD WHITE PAPER', 3, WAIT);
```

Table 6-33 summarizes the MESSAGE command parameters.

Table 6-33. **Summary of MESSAGE command parameters**

Parameter	Specifies	Offline	Online	DJDE
ITEXT	Message to the operator during input processing	Y	Y	Y
OTEXT	Message to the operator during job printing	Y	Y	Y

The following sections describe the syntax of the command parameters and explanations of the parameter options.



Note: All MESSAGE command parameters must be specified in a single command. If multiple message commands are specified, the parameters in the final command will be applied.

ITEXT

This parameter specifies a text message to be output to the operator during input processing.

Syntax MESSAGE ITEXT = *option(s)*

Options Table 6-34 lists the parameter option(s) and definition(s).

Table 6-34. ITEXT parameter option(s) and definition(s)

Option(s)	Definition(s)
NONE	Specifies that no text message is output to the operator during input processing. The default is NONE.
<i>sc</i>	Specifies a text message of up to 80 characters.
<i>passnum</i>	Specifies the pass (copy ply) to which the message text applies. The message is output to the operator just before processing of the indicated pass (copy ply) is begun. If no pass number is specified, the indicated message is output at the beginning of the first pass.
WAIT	Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.

Considerations Hexadecimal strings used in a MESSAGE command are interpreted as standard EBCDIC and are translated to extended ASCII.

OTEXT

This parameter specifies a text message to be output to the operator during job printing.

Syntax MESSAGE OTEXT = *option(s)*

Options Table 6-35 lists the parameter option(s) and definition(s).

Table 6-35. **OTEXT parameter option(s) and definition(s)**

Option(s)	Definition(s)
NONE	Specifies that no text message is output to the operator during job printing. The default is NONE.
<i>sc</i>	Specifies a text message of up to 80 characters (maximum of 400 characters per report).
<i>passnum</i>	Specifies the pass (copy ply) to which the text applies. Multiple <i>sc</i> messages, one per <i>passnum</i> , may be specified in a JSL. The message is output to the operator prior to the beginning of printing the specified report ply. If no pass number is specified, the text is output once at the beginning of printing the entire report.
WAIT	Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.
END	Specifies that the text is displayed after the last copy of the report is printed.

OUTPUT

The OUTPUT command controls the organization and format of a report. This includes control of the number of copies of a report, the forms to be printed with the variable data, report offsetting, and the use of simplex or duplex mode. Many graphics features may be invoked with this command; these are explained in greater detail later in this chapter and in the *Xerox DocuPrint 96/DocuPrint 96MX Laser Printing System Operations Reference*.

Table 6-36 summarizes the OUTPUT command parameters.

Table 6-36. **Summary of OUTPUT command parameters**

Parameter	Specifies	Offline	Online	DJDE
BFORM	Form only on back side of a duplex page	Y	Y	Y
COLLATE	Collated or uncollated report pages	Y	Y	Y
COPIES	Number of copies of a report	Y	Y	Y
COVER	Cover pages picked from the auxiliary tray	Y	Y	N
CYCLEFORMS	Associates forms with report pages in a cyclic fashion	Y	Y	N
DENSITY	IG local density functionality is listed for compatibility purposes. It is not supported on the Xerox DocuPrint 96/DocuPrint 96MX LPS	Y	Y	N
DESTINATION	Output destination of print jobs	Y	Y	Y
DUPLEX	Printing on both sides of a page	Y	Y	N
FACEUP	Face up delivery of pages	Y	Y	N
FEED	Paper tray selection	Y	Y	Y
FORMAT	Selection of page format descriptor	Y	Y	Y
FORMS	Associates forms with report copies	Y	Y	Y
GRAPHICS	Whether graphics are used and how	Y	Y	N
IMAGE	Initial scaling and positioning for batch mode graphics	Y	Y	Y
INVERT	Allows control of image orientation	Y	Y	Y
MODIFY	Associates CMEs with report copies	Y	Y	Y
NTO1	On a report basis, whether all copies of a particular report are printed first page to last (1-N) or last page to first (N-1)	Y	Y	Y
NUMBER	Automatic page numbering control	Y	Y	Y
OFFSET	Report offset control	Y	Y	N
OSTK	Specification of ordered stock parameter	Y	Y	N
PAPERSIZE	Paper size to be used for job	Y	Y	N
PURGE	Default disposition of document interleaved graphics	Y	Y	N
RESOLUTION	Output coordinate system in spots per inch (spi)	Y	Y	N
SHIFT	Shifts image in the y-direction	Y	Y	Y
SF1FUNCTION	Controls the sheet finisher function number 1	Y	Y	Y
SF2FUNCTION	Controls the sheet finisher function number 2	Y	Y	N
SIZING	Controls matching the paper specified when the page is formatted with the available paper sizes in the trays	Y	Y	Y
STOCKS	Calls out a specific STOCKSET to be used in the report	Y	Y	Y

Table 6-36. Summary of OUTPUT command parameters

Parameter	Specifies	Offline	Online	DJDE
SYSPPR	Specifies the system page paper size for a job	Y	Y	N
TMODE	Specifies system throughput parameters	Y	Y	Y
TRANS	Specifies whether transparencies are used for the job	Y	Y	N
UNITS	Specifies user-selected units for positioning graphics	Y	Y	N
XSHIFT	Shifts image in the x-direction	Y	Y	Y

The following sections describe the syntax of the command parameters and explanations of the parameter options.

BFORM

This parameter prints a form on the back side of a duplex data page. In association with the duplex printing mode DUPLEX=YES, a page containing only a form (no variable data can be printed with this page) may be printed on the back side of a user page. This feature can be used to print static data on the back of each page in a report without the use of DJDEs or the repetitive processing of that data. Different forms may be associated with different copies of a report by the use of multiple BFORM left parts on the same OUTPUT command. If BFORM specifications include some, but not all, copies of a report, those copies not included have blank back sides, that is, no form.

Syntax OUTPUT BFORM = *option(s)*

Options Table 6-37 lists the parameter option(s) and definition(s).

Table 6-37. **BFORM parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>form-id</i> (<i>form-id</i> , [<i>init</i>] [, <i>copies</i>] NONE	This option consists of: <ul style="list-style-type: none"> • <i>form-id</i> Specifies a one-to-six character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, which is called an FSL file, with the FDL system task (refer to the <i>Xerox LPS Forms Creation Guide</i>). A <i>form-id</i> may be the keyword NONE. • <i>init</i> If the <i>copies</i> parameter is not defined, the last (or only) form specified will apply to all copies beginning with copy number <i>init</i>. If the form is not the last one specified, <i>copies</i> defaults to 1. If neither <i>init</i> nor <i>copies</i> are specified, the form applies to all copies of the report. Note that <i>init copies</i> is only valid offline. • <i>copies</i> Specifies the number of plies (passes) to which a specified form applies. • NONE Specifies that no form is to be added to the associated report page of variable data. <i>init</i> parameter specifies the beginning ply (pass) number to which a specified form applies. This defaults to the first or next copy. The default is NONE.

COLLATE

This parameter specifies whether pages are to be collated or uncollated.

Syntax OUTPUT COLLATE = *option(s)*

Options Table 6-38 lists the parameter option(s) and definition(s).

Table 6-38. **COLLATE parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that the output pages are collated The default is YES.
NO	Specifies that the output is not collated. If DUPLEX=YES, then COLLATE=NO is ignored. Note that if NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES and notes this action with a warning in the JSL.

Refer to the "Points to note" section in this chapter for additional information.

COPIES

This parameter specifies the number of report copies to be printed.

Syntax OUTPUT COPIES = *option(s)*

Options Table 6-39 lists the parameter option(s) and definition(s).

Table 6-39. **COPIES parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>number</i>	An integer value with range 1 (the default) to 32767. For the copies DJDE parameter, a value of 0 may be specified for the DJDE COPIES parameter, in which case no copies are printed. The copies parameter on the START command allows an operator to override this parameter when initiating a print job. The default is 1.

COVER

This parameter specifies that cover pages are to be picked from the AUX cluster. These cover pages may be placed at the front or back of each copy of a report.

Syntax OUTPUT COVER = *option(s)*

Options Table 6-40 lists the parameter option(s) and definition(s).

Table 6-40. **COVER parameter option(s) and definition(s)**

Option(s)	Definition(s)
NONE	Specifies that no cover pages are to be picked. The default is NONE.
FRONT	Specifies that a cover page is to be picked (from the aux cluster) at the front of each copy and is the first page of the copy.
SEP	Specifies that each front cover does not have any data printed on it; also, no report data is printed on the back of a SEP cover.
BACK	Specifies that a cover page is to be picked at the end of each copy. No report data is printed on back covers.
BOTH	Specifies that both front and back cover pages are to be picked and the front of each copy is the first page of the copy.

Considerations

The FEED MAIN operator command (refer to the *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*) should be used in conjunction with this parameter to prevent the AUX tray from being automatically selected when the MAIN tray becomes empty. This can be used in combination with the FEED parameter on the OUTPUT command and the FEED DJDE parameter (refer to the chapter “Specifying dynamic job descriptor entries (DJDEs)”).

If RTEXT is specified alone with (FRONT, SEP) or (BOTH, SEP), RTEXT pages are printed on the covers, and front covers are picked only on copies for which there is RTEXT. (Refer to ROUTE commands for further details on RTEXT.)

Refer to the “Job parameter modification restrictions” section in the chapter “Specifying dynamic job descriptor entries (DJDEs)” for more information on COVER specified in a selected JDE.

CYCLEFORMS

This parameter specifies a set of forms to be associated with report pages in a cyclical fashion. Refer to the examples at the end of this section.

Syntax OUTPUT CYCLEFORMS = *option(s)*

Options Table 6-41 lists the parameter option(s) and definition(s).

Table 6-41. **CYCLEFORMS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>form-id</i> (<i>form-id</i> ₁ , <i>form-id</i> ₂ [,...]) NONE	This option has the following components: <ul style="list-style-type: none"> • <i>form-id</i> Specifies a 1 to 6 character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, called an FSL file with the FDL system task (refer to the <i>LPS Forms Creation Guide</i> for further details on FDL). • NONE Means that no form is to be added to the associated report page of variable data. The default is NONE.

If multiple CYCLEFORMS left parts on the same OUTPUT command are specified, a single list consisting of all forms specified in the OUTPUT command is created.

The order of the list is the order in which the names are encountered. The number of form-ids is limited only by the number of forms retained on the disk.

Example OUTPUT CYCLEFORMS=(*form*₁, *form*₂,);

DENSITY

This parameter is listed for compatibility purposes only; it is not supported on the Xerox DocuPrint 96/DocuPrint 96MX LPS.

Syntax OUTPUT DENSITY = *option(s)*

Options Table 6-42 lists the parameter option(s) and definition(s).

Table 6-42. **DENSITY parameter option(s) and definition(s)**

Option(s)	Definition(s)
FIX	
NOFIX	
DEFAULT	The default is DEFAULT.

Considerations

DESTINATION

This parameter specifies the output destination of paper.

Syntax OUTPUT DESTINATION = *option(s)*

Options Table 6-43 lists the parameter option(s) and definition(s).

Table 6-43. **Destination parameter option(s) and definition(s)**

Option(s)	Definition(s)
BIN	Specifies that the currently selected bin is used. The default is BIN.
TRAY	Sends output to the sample tray. Transparencies should always be sent to the sample tray. NOTE: Use caution when selecting the TRAY option because of the sample tray's limited capacity.
EXPORT	Specifies that sheets be sent to the bypass transport. The EXPORT option takes effect only if the current operator commands SELECT E or SELECT AUTO are specified. SELECT AUTO places output destination under data stream control. SELECT E causes output to be sent to the bypass transport option.

DUPLEX

This parameter specifies whether printing is to occur in duplex (print on both sides of a sheet) or simplex (print on a single side).

Syntax OUTPUT DUPLEX = *option(s)*

Options Table 6-44 lists the parameter option(s) and definition(s).

Table 6-44. **DUPLEX parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Selects duplex printing.
NO	Specifies simplex printing. The default is NO.

FACEUP

This parameter specifies faceup delivery of pages.

Syntax OUTPUT FACEUP = *option(s)*

Options Table 6-45 lists the parameter option(s) and definition(s).

Table 6-45. **FACEUP parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that the sheets are delivered to the tray faceup. Specify FACEUP=YES for labels. Delivering the sheets faceup prevents delamination when stacking.
NO	Specifies sheets are delivered facedown to the stacker tray. The default is NO.

Refer to the "Points to note" section for additional information.

FEED

This parameter controls the stock on which the page is printed.

Syntax OUTPUT FEED = *option(s)*

Options Table 6-46 lists the parameter option(s) and definition(s).

Table 6-46. **FEED parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stock-reference</i>	Specifies the stock assigned to a stock-name by the STOCKSET command in effect at the time the page is printed. FEED=stock-reference allows users to change stocks associated with a job, without altering the stock references in the data application itself.
<i>stock-name</i>	Bypasses the reference feature, but still requires that the stock-name is specified as present in the current STOCKSET. (FEED=stock-reference should always be used.)
MAIN,AUX, OPR	These options provide compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN. The default is OPR.

FORMAT

This parameter specifies a page descriptor entry (PDE) to be used in formatting the printed output, such as location of starting print line for each logical page on the physical page, font usage, and page orientation.

Syntax OUTPUT FORMAT = *option(s)*

Options Table 6-47 lists the parameter option(s) and definition(s).

Table 6-47. **FORMAT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>pde-id</i>	<p>References a PDE that must have been defined previously in a JDL or may make reference to a PDE file separately cataloged in the PDE directory on disk.</p> <p>Standard <i>pde-ids</i> are defined in the section "PDE command" in this chapter, such as FMT1 and FMT2. These standard <i>pde-ids</i> are part of the LPS and may be used unless a specialized PDE must be defined. An active PDE may subsequently be replaced entirely or modified in part through DJDEs (refer to the chapter "Specifying dynamic job descriptor entries (DJDEs)."</p> <p>Details on the PDE parameter are discussed in the "Page descriptors" section of this chapter. Creating and compiling PDE files with the PDL processor is discussed in the "Compiling the JSL" section in the chapter "Creating a job source library (JSL)."</p> <p>The default is FMT1.</p>



Note: To ensure reliability, enter the entire parameter FORMAT. Do not abbreviate to "FOR".

FORMS

This parameter specifies forms to be associated with the report copies. Different forms may be associated with different copies of a report by the use of multiple FORMS left parts on the same OUTPUT command.

Syntax OUTPUT FORMS = *option(s)*

Options Table 6-48 lists the parameter option(s) and definition(s).

Table 6-48. **FORMS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>form-id</i> (<i>form-id</i> [, <i>init</i>] [, <i>copies</i>] NONE</p>	<p>This option consists of:</p> <ul style="list-style-type: none"> • <i>form-id</i> Specifies a 1 to 6 character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, called an FSL file, with the FDL system task (refer to the <i>Xerox LPS Forms Creation Guide</i> for further details on FDL). • <i>init</i> Specifies the beginning ply (pass) number to which a specified form applies. This defaults to the first or next copy. If the copies parameter is not specified, the last (or only) specified form applies to all copies beginning with copy number <i>init</i>. If the form is not the last one specified, <i>copies</i> defaults to 1. If neither <i>init</i> nor <i>copies</i> is specified, the form applies to all copies of the report. • <i>copies</i> The number of copies of the report to be printed. • NONE Means that no form is added to the associated report page of variable data. <p>The default is NONE.</p>



Note: To ensure reliability, enter the entire parameter FORMAT. Do not abbreviate to “FOR”.

Example OUTPUT FORMS=(FORM3, 3, 2,) ;

GRAPHICS

This parameter specifies how graphics are to be processed in a job.

Syntax OUTPUT GRAPHICS = *option(s)*

Options Table 6-49 lists the parameter option(s) and definition(s).

Table 6-49. **GRAPHICS parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Indicates that there are no graphics in this job. The default is NO.
YES	Indicates there are graphics.
MOVE	Indicates that all referenced graphic disk files are to be copied into the print file.
BATCH	Means that the input stream contains no text, only graphics which are to be imaged, one per page.
NOSUB	Must be specified in the JDE using the GRAPHICS parameter if no substitution for the missing graphic image is desired. No substitution may be desired where graphics are specified for imaging with a reference scale factor that is greater than 4. Because the maximum effective scale factor is 8, graphic images in this case would be printed smaller than if resolution were 300 spi, (RESOLUTION=300). Note that this appearance error would not be indicated by the system.

Considerations If you are using the IMAGE parameter in conjunction with GRAPHICS, make sure to list the GRAPHICS parameter first, as these two parameters are order dependent.

The Xerox DocuPrint 96/DocuPrint 96MX LPS does not process 2-color graphics.

Examples OUTPUT GRAPHICS=BATCH,
RESOLUTION=300;

OUTPUT GRAPHICS=MOVE,
RESOLUTION=300,
PURGE=NO;

IMAGE

This parameter specifies, for batch mode processing only, the initial graphic imaging parameters to be used.

Syntax OUTPUT IMAGE = *option(s)*

Options Table 6-50 lists the parameter option(s) and definition(s).

Table 6-50. **IMAGE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>vpos</i>	<p>Specifies the vertical position of the top edge of the graphic, as an offset, relative to 0,0 on the current physical page (the position that would be specified by a PDE BEGIN of 0,0). This parameter is specified as a decimal number with up to three digits to the right of the decimal point and may be in the following units:</p> <ul style="list-style-type: none"> • DOTS • centimeters (CM) • inches (IN) • user-defined unit (UN). <p>If no units are specified, inches are assumed. An xdot is 1/600 unit of measurement.</p>
<i>hpos</i>	<p>Specifies the horizontal position of the left edge of the graphic, as an offset, relative to 0,0 on the current physical page. The form of the specification is the same as for <i>vpos</i>.</p>
[,n/d]	<p>Specifies the reference scale factor. Each parameter, <i>n</i> and <i>d</i>, must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied with that specified at the time the graphic was digitized and the product is rounded to the nearest integer in the range 1 to 8. That integer becomes the effective scale factor with which the graphic is imaged.</p>

The default is top left corner scaled at 1/1.

Considerations If the UN option is used to define positioning, the UNITS parameter (described subsequently) must precede this command in the parameter line.

The IMAGE parameter must follow GRAPHICS=BATCH parameter, as these two parameters are order dependent.

If two IMAGE parameters are listed consecutively, the system will process only the second one.

Examples OUTPUT GRAPHIC=BATCH,
IMAGE=(1 IN,2 CM);

INVERT

This parameter allows you to invert the current position of the page. A page can effectively be rotated 180°—head-to-head and head-to-toe. Inversion is also used for page orientation for finishing functions.

Syntax OUTPUT INVERT = *option(s)*

Options Table 6-51 lists the parameter option(s) and definition(s).

Table 6-51. **INVERT parameter option(s) and definition(s)**

Option(s)	Definition(s)
FRONT	Specifies page inversion on front sides.
BACK	Specifies page inversion on back sides.
BOTH	Specifies page inversion on front and back sides.
NONE	Specifies turning page inversion off.

MODIFY

This parameter specifies CMEs to be associated with report copies.

Syntax OUTPUT MODIFY = *option(s)*

Options Table 6-52 lists the parameter option(s) and definition(s).

Table 6-52. **MODIFY parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>cme-id</i> (<i>cme-id</i>, [<i>init</i>], [<i>copies</i>]) NONE</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>cme-id</i> The command identifier of a CME. The CME may be coded within a JSL or coded and cataloged separately for use by several JSLs. Any CME invoked by a DJDE must be cataloged separately (refer to the “Cataloged CMEs” section in this chapter for further information). If the CME is within the JSL, it must precede a reference to its identifier by the MODIFY parameter. Details on CME usage and syntax are discussed in the “CME” section of this chapter. • <i>init</i> Specifies the initial ply (pass) to which the associated CME is to be applied. • <i>copies</i> Specifies the number of plies (passes) on which to apply the CME. If copies is not specified, then the CME applies to all copies beginning with the copy number specified by <i>init</i>. • NONE Specifies that data are processed without modification. <p>The default is NONE.</p>

Different CMEs may be associated with different copies of a report by the use of multiple MODIFY parameters on the same OUTPUT command. CME copy ranges may not overlap.

If different CMEs are associated with different copies of a report, or if a CME is applied to some but not all copies of a report, then the CMEs are termed copy sensitive. Copy-sensitive CMEs may not be used if COLLATE=NO, or online processing is specified. Refer to the MODIFY DJDE in the chapter “Specifying dynamic job descriptor entries (DJDEs)” for a discussion of how CMEs may be changed on a page-by-page basis within a report ply.

NTO1

This parameter specifies, on a report basis, whether all copies of a particular report are printed last page to first (*n*-to-1). NTO1 is the command parameter keyword.

Syntax OUTPUT NTO1 = *option(s)*

Options Table 6-53 lists the parameter option(s) and definition(s).

Table 6-53. **NTO1 parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies last page to first printing. To print <i>n</i> -to-1, output processing must print the report starting with the last page. Therefore, output processing cannot begin until input processing has finished with the entire report. It is possible that input processing does not recognize the end of a report until the start of the next report (header banner page) is detected. When YES is specified, the system assumes a 50 sheet threshold, or maximum report size.
NO	Specifies first page to last (1TON) printing. The default is NO.
<i>n</i>	The NTO1 threshold. It specifies the maximum number of pages a report may have in order for it to be printed NTO1. If, however, the threshold count is reached and an end-of-report condition is not met, the system prints the entire report 1TON. The default value of NTO1 is 50 physical sheets, but <i>n</i> may range from 1 to 32767. When a value is greater than 100, the system generates a warning message.

Considerations When NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES, and notes this action with a warning in the JSL listing.

N-to-1 printing takes longer to process than 1-to-N printing because the system must process the entire report before beginning to print the job.

NT01=YES causes FACEUP=YES to be specified.

NUMBER

This parameter specifies page numbering on the output pages of a report. The page number character string is placed on the page at the specified line number based on the line spacing of the specified font. If override line spacing is specified for the font, it does not affect where the page number character string is to be printed.

Syntax OUTPUT NUMBER = *option(s)*

Options Table 6-54 lists the parameter option(s) and definition(s).

Table 6-54. **NUMBER parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p>(<i>pnum</i>,<i>lnum</i> ,<i>cnum</i> [,<i>findex</i>]) NO</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>pnum</i> Page number: specifies the starting number (an integer) for page numbering. The beginning page number may be non-positive. The number is incremented at page transitions but not printed until it goes positive. The maximum page number that will be printed is 2,147,483,647. • <i>lnum</i> Line number: specifies an integer line number on which the page number is to be placed. • <i>cnum</i> Column number: specifies an integer ending column number for the page number sequence. • <i>findex</i> Font index: the list of fonts specified in the FONTS parameter of the PDE command. If not specified, the first font in the list is used. • NO Specifies that no page numbering is to be performed. <p>The default is NO.</p>

Example NUMBER=(1 , 1 , 40 , 1 , 'RED');

If both BFORM and NUMBER are specified, BFORM pages are numbered.

OFFSET

This parameter specifies offset control on a report basis.

Syntax OUTPUT OFFSET = *option(s)*

Options Table 6-55 lists the parameter option(s) and definition(s).

Table 6-55. **OFFSET parameter option(s) and definition(s)**

Option(s)	Definition(s)
ALL	Results in an offset of each copy of each report. The default is ALL.
FIRST	Specifies that an offset is to occur only on the first copy of a report.
NONE	Specifies that there is to be no offset at any time.

The OFFSET control of FIRST, ALL, or NONE may be modified by the ROFFSET command (refer to the "Report offsetting on a record" section in the chapter "Using logical processing commands"). If OFFSET=FIRST or ALL and ROFFSET criteria is satisfied on page one of a report, the normal offset from the preceding report is nullified. No change should be made to the OFFSET parameter in a selected JDE.

OSTK

This parameter specifies ordered stocks.

Syntax OUTPUT OSTK = *option(s)*

Options Table 6-56 lists the parameter option(s) and definition(s).

Table 6-56. **OSTK parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p>[(<i>clu-def</i>],[<i>mod</i>],[<i>pos</i>],[RES NORES],[TABS],[<i>size</i>[DOTS IN CM]][,<i>L</i> <i>P</i>]]</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>clu-def</i> The name of the cluster containing the ordered stock (either a <i>cluster-name</i> or a <i>cluster-reference</i>). No default. • <i>mod</i> The number of sheets in the ordered stock. For example, 5 cut tabs would have a mod value of 5. The default is 1 and the range of values is from 1 to 254. • <i>pos</i> The initial position relative to the mod value. The default is 1 and the range of values is between 1 and mod. • RES or NORES Keyword to indicate if the pos value should be reset at the start of the next report. The default is RES. • TABS Keyword to indicate a tab stock. The default is no tabs. • <i>size</i> Width of the tab. The value can be specified in DOTS, IN or CM. The default is 0.5 in. This parameter only applies when tabs are specified. • L or P Indicates which edge of the paper the tab is located. If L, it is located in the scan counting direction. If it is P, it is located in the dot counting direction. The default is P.

The default is 1 for *mod* and *pos* RES no tabs>0.5 for *size* P.

PAPERSIZE

This parameter specifies the paper size to be used for printing the job.

Syntax OUTPUT PAPERSIZE = *option(s)*

Options Table 6-57 lists the parameter option(s) and definition(s).

Table 6-57. **PAPERSIZE parameter option(s) and definition(s)**

Option(s)	Definition(s)
A3	16.54 by 11.69 inches (297 by 420 mm)
A4	8.27 by 11.69 inches (287 by 270 mm)
B4	10.12 by 14.33 inches (257 by 364 mm)
USLEGAL	8.5 by 14 inches (216 by 356 mm)
USLETTER	8.5 by 11 inches (216 by 279 mm)
(x, y)	a variable paper size, that is, a paper size that is not one of the standard sizes.

If papersize is not specified, the LPS uses the system default paper size specified during system generation.

Considerations When centimeters (CM) are used to specify paper size, the maximum allowable value is 32.767 CM.

Paper size cannot be changed through DJDE on a page basis. Paper sizes can be changed on a report basis, however.

Whether you use the keyword or the (x, y) method for specifying papersize in your JSL, you must use the same method in your forms source library (FSL).

PAPERSIZE=B4 and PAPERSIZE=(10.12,14.33) produce identical results in that they indicate long-edge feed, as opposed to short-edge feed format. Refer to the "SEFFNT" section in the chapter "Specifying input parameters," for detailed information on short- and long-edge feed.

B5 paper size is an option that requires that a retrofit kit be installed.

Include performance considerations when switching paper sizes across pitch modes. Refer to "Using pitch and the TMODE parameter effectively" section in this chapter.

Example

```
OUTPUT  PAPERSIZE=USLETTER ,
        FEED=MAIN ,
        DUPLEX=NO ,
        NUMBER=( 1 , 1 , 80 ) ,
        FORMAT=SMPPOE ;
```

PURGE

This parameter specifies whether or not the system should delete graphic disk files at the end of report. This is only true for document interleaved.

Syntax OUTPUT PURGE = *option(s)*

Options Table 6-58 lists the parameter option(s) and definition(s).

Table 6-58. **PURGE parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Specifies that graphic disk files should not be deleted.
YES	Specifies that all IMG graphics files created or replaced by document interleaved graphic processing for this report should be deleted at the end of the report printing. The default is YES.

A DJDE SAVE specification may be used to override the purging of specific files.

RESOLUTION

This parameter defines the output coordinate system (300 or 600 spots per inch). 300 specifies that output, including fonts, forms, images, and logos, be printed at 300 spots per inch (spi). 600 refers to all fonts, forms, images, and logos in FN6, FR6, LG6, and IM6 directories. The Resolution statement is only included for compatibility. The Xerox DocuPrint 96/DocuPrint 96MX LPS accepts 300 spi fonts, forms, images and logos and converts them to 600 spi using the majority interpolation scheme for printing. The Xerox DocuPrint 96/DocuPrint 96MX LPS cannot accept 600 spi fonts, forms, images, and logos. On systems printing at 600 spi, references in the JSL to 1IN, 2.54 cm, or 300 dots are converted to a pixel count of 600 spi in the JDL, the system converts all measurements to XDOTS.

Syntax OUTPUT RESOLUTION = *option(s)*

Options Table 6-59 lists the parameter option(s) and definition(s).

Table 6-59. **RESOLUTION parameter option(s) and definition(s)**

Option(s)	Definition(s)
300	Specifies 300 spots per inch.
600	Specifies 600 spots per inch.

The default is sysgen parameter (300 spi on the Xerox DocuPrint 96/DocuPrint 96MX LPS).

SF1FUNCTION

This parameter provides control over third-party finishing devices that conform to the DFA standard. Using this command, you can invoke sheet finisher function number 1 through the DFA channel C6.

Syntax OUTPUT SF1FUNCTION = *option(s)*

Options Table 6-60 lists the SF1FUNCTION parameter option(s) and definition(s).

Table 6-60. **SF1FUNCTION parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies to the finisher to begin a particular finishing function on the current sheet.
NO	Specifies to the finisher not to begin a particular finishing function on the current sheet. The default is NO.

SF2FUNCTION

This parameter provides control over third-party finishing devices that conform to the DFA standard. Using this command, you can invoke sheet finisher function number 2 through the DFA channel C7.

Syntax OUTPUT SF2FUNCTION = *option(s)*

Options Table 6-61 lists the SF2FUNCTION parameter option(s) and definition(s).

Table 6-61. **SF2FUNCTION parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies to the finisher to begin a particular finishing function on the current sheet.
NO	Specifies to the finisher not to begin a particular finishing function on the current sheet. The default is NO.

SHIFT

Image shift is the shifting of the image data (system page) relative to the physical page. This shifting is in the scan counting direction (vertical for landscape and horizontal for portrait). The SHIFT parameter specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data in the y-direction for three-hole paper, binding, finishing, and edgemarking.

Syntax OUTPUT SHIFT = *option(s)*

Options Table 6-62 lists the parameter option(s) and definition(s).

Table 6-62. **SHIFT parameter option(s) and definition(s)**

Option(s)	Definition(s)
v ₁	An integer value in dots for the amount of shift on the simplex page or the odd (front) side of the duplex page. (Each dot is 1/300 of an inch.) Range for v1 is -75 to 75.
v ₂	A value in dots for the amount of shift on the even (back) side of a duplex page. Range for v2 is -75 to 75. A standard size shift occurs if YES is specified (v1=75 and v2=-75).
YES	Specifies that a shift occurs. A standard size shift occurs if YES is specified (v1=75 and v2 =-75).
NO	Specifies that no shift occurs (v1=0 and v2=0). The default is NO.

If the origin of a print line falls off the page, the entire line of data fails to print. This is especially important for duplex portrait-oriented pages, where a miscalculated shift could result in the entire page failing to print. When duplex shift is used, the left margin must be at least .25 inches (75 dots) wide. When using the SHIFT parameter, make sure that the data is not shifted past the margin and off the page.

Refer to the "Points to note" section of the VOLUME command in the chapter "Specifying input parameters."

SIZING

This parameter specifies how the system matches the paper sizes in bins when the LPS is unattended.

Syntax OUTPUT SIZING = *option(s)*

Options Table 6-63 lists the parameter option(s) and definition(s).

Table 6-63. **SIZING parameter option(s) and definition(s)**

Option(s)	Definition(s)
SEMIAUTO	The standard mode of operation. The default is SEMIAUTO.
BEST	Intended for use when the LPS is unattended. Called a nonstop mode because the LPS does not cycle down. The system selects the largest available paper size to print the report if the exact paper size is not available.
EXACT	Intended for use when the LPS is unattended because the LPS does not cycle down. If the exact paper size is not available, the system aborts the report.

A paper size is equal in size to another paper size if neither dimension differs by more than 15/600 of an inch.

Can be used to print on paper larger than the specified paper size. Data may be lost if the paper size is smaller than the image size.

Note that A4 paper is treated as smaller than 8.5 by 11 inch paper because it is narrower.

Regardless of the parameter selected, if an exact match in paper size is found, the matching tray is used. The parameters differ only when no match is found.

If there is not an exact paper size match in the effective cluster, but there is paper in the effective cluster larger than the specified size, then the following occurs:

- SEMIAUTO cycles down and requires operator confirmation. When the message CON O displays, and the larger size is confirmed, the larger paper is used. Further mismatches of this type are ignored until:
 - The STOCKS=parameter changes, or
 - End-of-report and no STOCKS=is in effect, or
 - End-of-job
 - The printer cycles down. The operator must either press CONTINUE to conform the larger paper, change the paper or the effective cluster before the printer will cycle up. The operator can also abort the report.
- BEST causes the following result: The larger paper is automatically used.
- EXACT causes the report to be aborted if the printer does not cycle down.

STOCKS

This parameter defines the stockset and its associated stock(s) to be used in a report. If the stockset-name does not reference a STOCKSET command coded earlier in the same JDL, the system assumes that the stockset exists globally as an STK file, which is read at print time. In this case, PDL inserts a message in the JSL file listing, indicating that an STK file will be used at print time.

Whenever a new stockset is chosen at start of report or through a DJDE JDE and JDL switch, each stock is checked to determine that the stock exists and can be made active. This provides an automatic method of changing stocks in the LPS as required by the data stream.

Refer to the "STOCKSET command" section in the chapter "Defining clusters" for additional information on defining stocks.

Syntax OUTPUT STOCKS = *option(s)*

Options Table 6-64 lists the parameter option(s) and definition(s).

Table 6-64. **STOCKS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stockset-name</i>	Specifies the stockset to be used. There is no default.

SYSPPR

This parameter specifies the system page paper size for a job.

Syntax OUTPUT SYSPPR = *option(s)*

Options Table 6-65 lists the parameter option(s) and definition(s).

Table 6-65. **SYSPPR parameter option(s) and definition(s)**

Option(s)	Definition(s)
USLEGAL	8.5 by 14 inches (216 by 279 mm)
A3	16.54 by 11.69 inches (420 by 297 mm)
A4	8.27 by 11.69 inches (210 by 297 mm)
B4	10.12 by 14.33 inches (257 by 364 mm)
USLETTER	8.5 by 11 inches (216 by 279 mm)
(x, y)	x defines the slow scan direction in inches. y defines the fast scan direction in inches.

There is no default.

Considerations

When centimeters (CM) are used to specify paper size, the maximum allowable value is 32.767 cm.

SYSPPR is not available as a DJDE.

Whether you use the keyword or the (x, y) method, for specifying papersize in your JSL, you must use the same method in your forms source library (FSL).

PAPERSIZE=B4 and PAPERSIZE=(10.12,14.33) produce identical results in that they indicate long-edge feed, as opposed to short-edge feed format. Refer to the "SEFFNT" section in the chapter "Specifying input parameters" for detailed information on short- and long-edge feed.

TMODE

This parameter specifies paper sizes in order to improve throughput efficiency.

Syntax OUTPUT TMODE = *option(s)*

Options Table 6-66 lists the parameter option(s) and definition(s).

Table 6-66. **TMODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>width</i> [, <i>width unit</i>])	This option has the following components: <ul style="list-style-type: none"> • <i>width</i> The process direction measurement that determines the default pitch mode. • <i>width unit</i> Defines the unit of measurement for the default pitch mode in IN, CM, DOTS, and XDOTS. IN (INCHES) is the default.

Width values can range from the 8-inch equivalent to the 17-inch equivalent.

If you use inches (the default), refer to the table in the “Points to note” section for the value ranges, pages-per-minute speeds, and uses available for the various pitch modes.

The default depends on the specified paper size; the default value is in inches.

TRANS

This parameter specifies whether transparencies are used for the print job.

Transparencies cannot be handled by the high-capacity feeder/stacker. Due to this restriction, transparency jobs must feed from the processor feed trays (trays 1 or 2; also known as MAIN and AUX feed) and stack to the sample tray.

Syntax OUTPUT TRANS = *option(s)*

Options Table 6-67 lists the parameter option(s) and definition(s).

Table 6-67. **TRANS parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Indicates that transparencies will be used in the job.
NO	Indicates that transparencies will not be used in the job. The default is NO.

UNITS

This parameter specifies the position of a graphic when referenced by the IMAGE parameter of the OUTPUT command or an IMAGE, ALTER, or GRAPHIC DJDE parameter.

Syntax OUTPUT UNITS = *option(s)*

Options Table 6-68 lists the parameter option(s) and definition(s).

Table 6-68. **UNITS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>size in dots</i>	Defines the dot size of a unit. The size defines the number of dots in one unit and is specified as a decimal number with up to two digits to the right of the decimal point. When used to compute the position of a graphic, the result is rounded to the nearest dot. There is no default.

XSHIFT

This parameter specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data for two-hole paper, binding, finishing, and edgemarking. XSHIFT allows shifting in the x-direction for short-edge binding.

Syntax OUTPUT XSHIFT = *option(s)*

Options Table 6-69 lists the parameter option(s) and definition(s).

Table 6-69. **XSHIFT parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Specifies that no shift occurs.
<i>value</i>	An integer that specifies a shift amount on the front and back side. The shift is in the x-direction and each dot is 1/300 of an inch. The allowable range is 75 to -75.

Example

```

INVPDE:  PDE  PMODE=PORTRAIT,
          BEGIN=( 1.3 , .37 ) ,
          FONTS=P0612A;

INVJDE:  JDE ;
          OUTPUT  INVERT=BACK ,
                  SHIFT=YES ,
                  XSHIFT=-40 ,
                  FORMAT=INVPDE ,
                  PAPERSIZE=( 8.5 , 11 ) ;
    
```

XMP

Xerographic mode persistence (XMP) controls xerographic mode switching (XMS) to use for the report. Depending on the option specified, the effect is to maximize throughput or minimize toner usage.

Syntax OUTPUT XMP = *option(s)*

Options Table 6-70 lists the parameter option(s) and definition(s).

Table 6-70. **XMP parameter option(s) and definition(s)**

Option(s)	Definition(s)
DEFAULT	Specifies that the sysgened XMP is used for the report. The default is DEFAULT.
REPORT	Specifies that when the system determines that a report requires highlight color mode for any page of the report, all pages remaining unprinted are printed in the highlight color mode.

Points to note — OUTPUT command

Note the following when using the OUTPUT command. These points to note list parameters followed by the applicable action or system response.

- **DUPLEX=YES**

- Any job that runs in simplex mode runs in duplex mode except transparencies.
- A recoverable dispatching or imaging error on an even-numbered page results in one additional hole in the paper path and up to seven aborted pages to the sample tray. A recoverable dispatching or imaging error on an odd-numbered page results in one additional hole in the paper path with no aborted pages to the sample page.

- **COLLATE**

Once you have specified multiple copies, the value of the COLLATE parameter cannot be changed again within the report.

- **NT01, FACEUP, DUPLEX, COLLATE**

The NT01, FACEUP, DUPLEX, and COLLATE parameters follow a specific hierarchy by which one command parameter forces changes in certain other command specifications when a YES option is selected. This hierarchy of forced conditions is shown in table 6-71. The first column gives the condition that will force the parameters shown in the columns to the right.

Table 6-71. **OUTPUT command parameter hierarchy of forced conditions from highest to lowest priority**

Parameter specification	Results in NT01=	Results in FACEUP=	Results in COLLATE=	Results in DUPLEX=
NT01=YES	—	—	YES	<i>specified option</i>
FACEUP=YES	—	—	—	<i>specified option</i>

Example: FACEUP=YES , COLLATE=NO ;

Example: NT01=YES , FACEUP=NO ;

- **NT01 and 1TON system defaults**

- During system generation, you may select either 1TON or NT01 as the system default in the DEFAULT ORDER = {1-N | N-1} window.
- When the system default is 1TON, NT01 is processed only if it is invoked in the JDE or JDL. The system ignores NT01 in DJDE-selected JDEs or JDLs.
- When the system default is NT01, the system assumes each report is NT01 until the first data page, even if 1TON is specified in the JDE or JDL. This arrangement allows you to use DJDE-selected JDEs and JDLs that invoke NT01 after a series of banner pages.

- The following examples illustrate these conditions:

Example 1: The default order is 1TON, OUTPUT NTO1=NO is specified, and a JDE or JDE dynamic job descriptor entry (DJDE) containing NTO1=YES is invoked.

The system does not process the DJDE.

Example 2: The default order is 1TON and OUTPUT NTO1=YES is specified.

The system allows NTO1 printing if NTO1 threshold values (*n*) are within permitted limits. Note that NTO1 DJDEs are never allowed with the 1TON system default.

Example 3: The default order is NTO1 and OUTPUT NTO1=NO is specified.

The job prints 1TON unless a JDE or JDL DJDE specifying NTO1 is invoked in the first data record after banner pages.

Example 4: The default order is NTO1, OUTPUT NTO1=YES is specified, and a JDE or JDL DJDE specifying NTO1=NO is invoked.

The DJDE is processed and the job prints 1TON, if the system encounters it in the first data record after banner pages.

- **NTO1, OSTK**

When printing ordered stock in NTO1 sequence, note the following:

The ordered stock must be reverse collated in the feeder so that the first sheet of the set is on the bottom, the second sheet next, and so on, with the last sheet of the set on top to be fed first. If the report is larger than the N to 1 threshold, it will print in 1 to N order and the reverse collated stock is then inappropriate. The system displays the following message and cycles down when this condition occurs:

```
OS6908 Page ordering has been reversed. Check
ordered stock.
```

- **NTO1 and segments**

When finishing segments are defined in an NTO1 report, the entire report prints N to 1. The system does not process the segments separately. If the entire report contains more pages than the N to 1 threshold allows, even though individual segments may be less than the threshold, the entire report is printed 1 to N.

- **NTO1=NO, FACEUP=NO, DUPLEX=NO**

- In this sequence, the NTO1 and FACEUP parameters specify the NO option so other potentially affected parameters, such as DUPLEX, cannot be overridden. The DUPLEX parameter, therefore, maintains either the YES or NO option.
- If hierarchical overrides occur, the system issues warning messages in the JDL and then forces parameters lower in the hierarchical structure to their workable values.

- **DUPLEX=NO, ACCT USER=TRAY, BFORM ignored, RPAGE SIDE=NUBACK, forced to NUFONT**

If transparency (TRANS) is specified, the input task forces these parameters. Any operator key-ins, however, override the JSL specifications.

- **CYCLEFORMS, FORMS**

Both may be used within a report; however, only the last parameter specified is in effect at any given time. If a multicopy report specifies CYCLEFORMS, a FORMS DJDE encountered midreport overrides the CYCLEFORMS parameter from that point in every copy.

- **RAUX command, FEED parameter of OUTPUT command**

If they specify two different cluster names, RAUX has higher priority.

- **SYSPAGE parameter of STOCKSET command**

System pages are checked for ordered stocks through the SYSPAGE parameter. If there is no cluster name active for the current page, the input task checks to see if there was an operator FEED cluster-name specified. If so, the input task tests on the operator's specified cluster-name.

- **Ordered stock (OSTK)**

All applications that are printed on the tab have pages formatted for tab printing. If there is a PDE switch in the job stream that causes the BEGIN value to be shifted off the page, (BEGIN=(.18, 8.6)), the cluster criteria must already be met before processing the new BEGIN. This allows the input task to bypass BEGIN value checking for tab stocks.

PDL ensures the minimum value for MOD is 1 and the maximum is 254 (decimal). It also ensures the value of POS is greater than or equal to 1 and less than or equal to MOD.

- **Printing on a tab**

- The Xerox DocuPrint 96/DocuPrint 96MX LPS prints on tabs in both simplex and duplex modes.

When printing on 9-inch stock or 8.5-inch stock with 1/2 inch tabs, the full imaging area of 9-inch is enabled. It is limited, however, by the amount of image shift enabled.

- When running a tab stock job, you must not include the width of the tab when specifying the PAPERSIZE command. However, the width of the tab must be factored in when reporting the paper size to the system with the SET TRAY SIZE command on the "Tray size" screen or on the printer monitor.
- A command DJDE FEED=Cluster name/cluster reference is required to switch to tab stock or whatever it would need to switch the current cluster to a tab stock cluster. This may include RAUX, RFEED, COVER=FRONT, operator keyin, etc.

- INPUT processes tabs by keying off the current cluster name/reference at the time that the page is formatted for OUTPUT. For example, if the page satisfies the RAUX criteria and AUX is specified as the cluster name in the ordered stock statement, INPUT recognizes this as a tab stock. Having recognized it, INPUT then adds *n* dots (width of the tab) to the paper size in the slow scan direction. The IOT disables any dynamic width checking whenever it encounters a tab.

The portrait/landscape parameter of the OSTK statement allows either to be specified. The only one supported by the Xerox DocuPrint 96/DocuPrint 96MX LPS is portrait (P), and this is the default condition. If landscape (L) is selected, and the actual tab stock is loaded such that the tab is located on the registration edge (outboard), this combination will not run due to skew induced jams within the IOT.

In ordered stock tab jobs, care must be exercised when selecting the PFEED tray. Ordered stock jam recovery does not work properly if the tray selected for PFEED contains ordered stock.

Example: Job 1 starts with Tab 1 (located in the upper right corner of a portrait page) and Job 2 must start with Tab 5 (located in the lowest right corner of the portrait page). The number of tabs Job 1 will use is 2.

In order to run the job in succession, you must code the JSL with MOD=5 and POS=1 for Job 1 and MOD=5 and POS=5 for Job 2. The Xerox DocuPrint 96/DocuPrint 96MX LPS does keep track of the last tab position in tray 2 at the end of Job 1. Upon starting Job 2, the Xerox DocuPrint 96/DocuPrint 96MX LPS purges Tab 3 and Tab 4 because Job 2 starts with POS=5 and Job1 used Tab 1 and Tab 2.

You can improve performance by ordering application dependent tab stock. By using ordered tab stock which supports only those number of tabs required for a specific job, eliminates the need for purging unused tabs.

- At the time the system makes validity checks for BEGIN values and graphic positioning, input processing cannot determine if the current physical page is tab stock. However, if the OSTK command is specified, the input task will allow a tolerance of *n* inches (where *n* is the width of the tab specified in the OSTK command when performing validity checks on BEGIN values and graphic positioning). In addition, the FEED=tab-stock DJDE must precede any printable text data which will be printed on the tab.
- When running tab stock jobs:
 - Do not include the width of the tab when specifying paper size in the job setup.
 - Include the width of the tab when reporting it to the printer on the printer control console “Tray size” screen or the system UI Set Tray Size command.
 - Dedicate one tray for tab stock.
- When Using DJDEs, specify:

```
FEED=tab-stock
SIMPLEX=YES or DUPLEX=YES
```

- When feeding from tab stock:
 - Be sure to revert the DJDEs back to:
`FEED=regular-stock`
`SIMPLEX=YES` or `DUPLEX=YES`
after printing the tab stock.
 - Ensure that PFEED does not specify the tray that contains the tab stock.
 - Use the OSTK command before running a tab job.
- **Specifying a tab as a form**

To define a form that is to be used as a tab in a form call out, it is recommended that you code your applications with the following specifications:

- JSL must specify `PAPERSIZE=USLETTER` (or 8.5, 11)
- FSL must specify `PAPERSIZE IS 9 BY 11` (or 8.5 by 11) with 1/2 inch tab.
- One tray must be dedicated for tabs. The JSL must specify which cluster name/reference contain the tab stock. Be sure to select an alternate tray as the PFEED source.
- A DJDE JDE, JDL switch that defines the following:

`PAPERSIZE = (9, 11) or 8.5 by 11 with 1/2 inch tab`

`FEED=tab-cluster`

`DUPLEX=YES` or `DUPLEX=NO`

`FORM=XXXXXX`

Width specified for ordered stock must be zero.

Specify TABS in the ordered stock statement so that dynamic width checking will be disabled.

A corresponding DJDE JDE, JDL switch to print regular paper stock. Note that the ordered stock statement specified in a DJDE JDE, JDL and a regular stock DJDE JDE, JDL must be exactly alike.

You may define the ordered stock statement at the job or the system level. If you define it at the job level, you must ensure that the ordered stock command defined from one JDE, JDL to the other are exactly alike. The reason is because the input task contains an internal mod/pos table that keeps track of the current mod/pos values for the job. This table is initialized at report set-up time through values found in the START JDE or JDL command. When a DJDE JDE or JDL is specified (with an ordered stock command), the input task will then reinitialize its internal mod/pos table with values in the DJDE JDE or JDL if it finds that the ordered stock command found in the START JDE or JDL is different from those defined in the DJDE JDE or JDL. However, if both commands are alike, the input task will not reinitialize the internal mod/pos table. In order to check if the internal mod/pos table is being initialized, you will experience tab stocks that start purging before every tab is imaged (because MOD/POS is always initialized for every tab page).

- **Paper sizes**

The PAPERSIZE parameter is not supported as a DJDE. Therefore, a JDE page-oriented DJDE that calls out the desired paper size is needed.

- When specifying paper sizes, if the JSL specifies the paper size using the keyword method (such as USLEGAL, A4, or B4), then the FSL must also use the same method. If the JSL specifies 8.5,14 and FSL specifies USLEGAL, the INPUT will display an error message.

```
OS6670 FORM PAPER SIZE TOO SMALL. FORM WILL
NOT BE PRINTED or OS6680 FORM PAPER SIZE TOO
LARGE. FORM WILL NOT BE PRINTED.
```

USLETTER and B4 are the only exceptions to this rule where the keyword and x, y method will not create any incompatibilities.

- When printing system pages, if SYSPAGE=cluster is not specified, then INPUT will default to cluster MAIN. SYSPPR defines the paper size to use for the system page. Therefore, if cluster MAIN contains a mix of USLETTER and USLEGAL, then OUTPUT will search for the appropriate tray that contains the required paper size. If SYSPPR is not defined, then INPUT will use the current job's paper size.

- **SYSPPR**

The SYSPPR parameter will take effect for the following system pages:

- DJDE error page (OPRINFO parameter of the IDEN command)
- Tape label page (PLABEL parameter of the VOLUME command)
- Report delimiter page (PRINT parameter of the RSTACK command)
- Accounting page (ACCT command)
- Graphic error summary page
- Short-Edge Feed error page

SYSPPR defines a paper size, whereas SYSPAGE defines a cluster-name reference. Both parameters must agree in order to prevent errors, similar to the way the FEED and PAPERSIZE parameters interrelate to each other.

- **Using pitch and the TMODE parameter effectively**

Paper sizes each have an associated pitch mode.

There are three processing modes available on your LPS. These are called pitch modes. A pitch is a term describing the time it takes to image a page on the printer photoreceptor belt. A pitch mode is a phrase that describes how many pitches can occur during one complete photoreceptor revolution. For example, while processing in 3-pitch mode, three images can be placed on the photoreceptor belt during one revolution of the belt.

As page size increases, fewer images can be placed on the photoreceptor belt during one revolution. Since the belt runs at a constant speed, a decrease in the number of images that can be placed on the belt causes a corresponding drop in the pages-per-minute (ppm) throughput of the LPS. Table 6-76 shows the various pitch modes, their corresponding ppms, and the default paper sizes for each pitch mode. The paper size used in a job determines the highest pitch mode in which the system will run that job.

If you change paper sizes within a job, and this change crosses a pitch boundary, the system performs a xerographic quality adjustment, which may significantly impact performance.

Specifying the pitch (also called the throughput mode) with the TMODE parameter of the OUTPUT command allows you to control the default pitch mode with the following benefits:

- Matching system throughput with finishing device restrictions: If the finishing device attached to the printer cannot accept output at rated speed, the printer inserts gaps in the paper path to ensure that pages do not arrive at the device too rapidly.
- Eliminating process adjustment cycles during jobs that cross pitch boundaries: Each time a job crosses a pitch boundary, the system performs a xerographic quality adjustment. If this is done frequently within a job, the adjustments may take a substantial amount of the overall print time. Lowering the pitch mode so that the entire job runs at the same mode may enable you to significantly increase performance.
- Estimating the time needed to run your print jobs for scheduling purposes. To better understand the potential value of specifying pitch modes in your JSLs, consider these hypothetical situations:
 - Your finishing device folds and binds your printed output but accepts pages at a considerably slower rate than the 96 pages-per-minute capability of your Xerox DocuPrint 96/DocuPrint 96MX laser printing system. You may want to slow down the rate at which your Xerox DocuPrint 96/DocuPrint 96MX LPS sends printed sheets to this device by specifying a 4-pitch mode. Assuming that your application calls for 8.5 by 11-inch paper, you could use 5-pitch mode, but by selecting 4-pitch, you reduce the pages-per-minute (ppm) from 96 to 77, which better matches your finishing device's capabilities and reduces the potential for paper jams.

- Your application uses four paper sizes, ranging from 8 by 11-inches for the first few pages to 11 by 17-inches for the last several pages. Each time your printer detects a different paper size, it adjusts to accommodate the new paper size, thus noticeably reducing throughput time. By specifying a large enough pitch mode to accommodate all the paper sizes in your application, in this case, 3-pitch mode, the entire job runs at the maximum possible throughput speed because paper size adjustments are limited. Keep in mind that you may select a lower pitch mode (allowing a greater scan value) for smaller size paper but that you would not want to select a pitch mode too small to accommodate a larger paper size.
- You want to estimate how many applications you could run on a particular shift. But, given the size and complexity of your applications, you are uncertain as to how to plan. By looking at the paper sizes, and thereby, the pitch modes required by each application, and the number of pages in each application, and by associating these with the ppms provided in tables 6-10 and 6-11, you can then roughly estimate the length of time each application requires for printing.

As you can see, there are no absolutes and no set methods for accomplishing your printing goals. The variables involved include individual finishing device requirements, the paper sizes required by your applications, the complexity of each page, the number of pages in each application, and so forth. But a general understanding of the relationship between paper size and pitch mode, and the advantages of specifying pitch modes will help you optimize your printing activities.

The Xerox DocuPrint 96/DocuPrint 96MX LPS operates in multiple pitch modes: 3, 4, and 5. Table 6-72 shows the pitch mode minimum and maximum boundary values, in inches, along with their associated print speeds, shown in pages per minute, which can be specified with the TMODE parameter of the OUTPUT command.

Table 6-72. **Pitch mode minimum and maximum boundary values and pages-per-minute throughput**

IN	CM	MM	DOTS	XDOTS	Pages per minute (ppm)
15.31 < x < 17.00	38.9 to 43.18	389 to 431.8	4595 to 5100	9190 to 10200	58
12.12 < x < 15.31	30.8 to 38.9	308 to 389	3639 to 4594	7277 to 9189	78
8.00 < x < 12.12	20.32 to 30.8	203 to 308	2400 to 3638	4800 to 7276	96

Long-edge feed mode means the sheet is fed into the paper path with its long edge first. For papers whose width ranges from 8.5 inches to 9.01 inches, the Xerox DocuPrint 96/DocuPrint 96MX LPS prints with slightly reduced productivity. The reason for this is that for papers in this width range, the normal image area extends into the inter-image zone that is normally used for xerographic process controls. In order to maintain the best productivity possible, xerographic process controls are temporarily suspended. Periodically, the Xerox DocuPrint 96/DocuPrint 96MX LPS will induce about 14 skipped "images" in order to perform a xerographic process control check and ensure that the Xerox DocuPrint 96/DocuPrint 96MX LPS maintains the highest print quality. It will then resume printing at full productivity.

For any paper size wider than 15.32 inches, 389 mm, in width, the system automatically switches to 3-pitch mode, which prints up to 58 pages per minute. For very large paper, 11 inches by 17, inches for example, paper is short edge fed, so that the 11 inch side leads through the printer.

You may also specify pitch mode boundaries in centimeters (CM), DOTS, and XDOTS. The minimum and maximum scan values with these units of measurement and the pitch mode ppm print speeds are shown in table 6-76.

- **Purging stock**

The Xerox DocuPrint 96/DocuPrint 96MX LPS can purge tab sheets whenever the tab cycle is not in sync with the formatted data. This may be caused by paper jams or running multiple jobs in succession. For example, if a typical application had the following scenario:

Two tab stock jobs to run in succession. Both jobs require tray 1.

Tray two contains the tab stock which has MOD=5.

Job 1 starts with Tab 1 (located in the upper right corner of a portrait page) and Job 2 must start with Tab 5 (located in the lowest right corner of the portrait page.)

The number of tabs Job1 will use is 2.

In order to run the job in succession, you must code the JSL with MOD=5 and POS=1 for Job 1 and MOD=5 and POS=5 for Job2. The Xerox DocuPrint 96/DocuPrint 96MX LPS will keep track of the last tab position in tray 2 at the end of Job 1. Upon starting Job 2, the Xerox DocuPrint 96/DocuPrint 96MX LPS will purge Tab 3 and Tab 4 because Job 2 starts with POS=5 and Job 1 used Tab 1 and Tab 2.

You can improve performance by ordering application-dependent tab stock. By using ordered tab stock which supports only those number of tabs required for a specific job, you will eliminate the system from purging unused tabs.

Examples

This section shows examples of using the OUTPUT command.

Example 1

```
OUTPUT
GRAPHICS=MOVE , UNITS=150 , CYCLEFORMS=NONE , FORMAT=PDE
1 , RESOLUTION=300 ;
```

Example 2

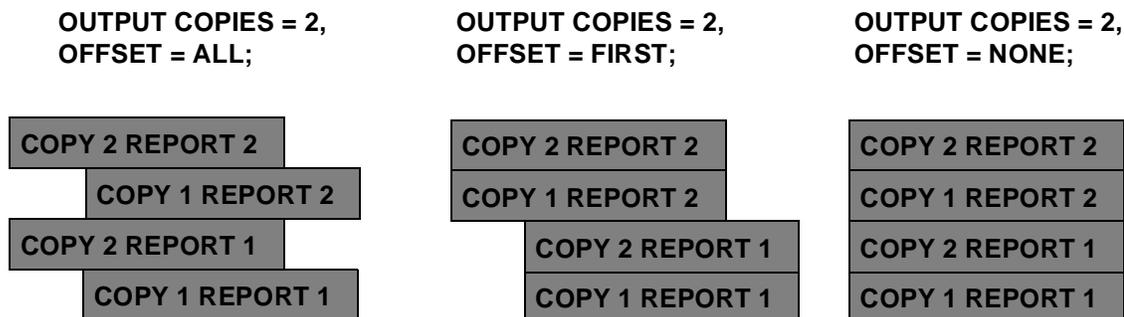
```
OUTPUT COPIES=3 , DUPLEX=YES , SHIFT=YES , OFFSET=FIRST ,
NUMBER=( 1 , 66 , 132 ) , COVER=( FRONT , SEP ) , FORMS=GBAR ;
```

Pages printed with this OUTPUT command are in duplex mode with a margin shift of 75 dots. Three copies of the report are made, the first of which is offset; the second and third copies are stacked on top of the first. Each page is numbered at the end of line 66. The form GBAR appears on each data page of the report.

Example 3

The commands in figure 6-6 illustrate use of the offsetting features of the OUTPUT command. The parameter ALL offsets all copies of all reports; FIRST offsets just the first copy of each report and NONE inhibits offsetting.

Figure 6-6. Output command offsetting features



Example 4

```
OUTPUT CYCLEFORMS=( FORM1 , NONE , FORM3 ) ;
```

or

```
OUTPUT CYCLEFORMS=( FORM1 , , FORM3 ) ;
```

If there are ten pages in the report, either of the above commands would cause FORM1 to be applied to pages 1, 4, 7, and 10; no forms to be applied to pages 2, 5, and 8; and FORM3 to be applied to pages 3, 6, and 9.

Example 5

The commands in figure 6-7 illustrate the use of cycleforms with multiple copies. Note that every copy of a particular page has the same form whether COLLATE=YES or NO. The pages in the following example are not logical pages but are one side of a logical sheet of paper.

Figure 6-7. **Cycleforms with multiple copies**

**OUTPUT COPIES = 2,
COLLATE = NO, DUPLEX = NO,
CYCLEFORMS = (FORM 1, NONE);**

**OUTPUT COPIES = 2,
COLLATE = YES
DUPLEX = YES
CYCLEFORMS = (FORM 1, NONE)**

COPY2 PAGE4
COPY1 PAGE4
FORM1 COPY2 PAGE3
COPY2 PAGE2
COPY1 PAGE2
FORM1 COPY2 PAGE1
FORM1 COPY1 PAGE3

COPY2 PAGE4
FORM1 COPY2 PAGE3
COPY2 PAGE2
FORM1 COPY2 PAGE1
COPY1 PAGE4
FORM1 COPY1 PAGE3
COPY1 PAGE2
FORM1 COPY1 PAGE1

PDE command

The PDE command specifies a page descriptor entry (PDE) which defines formatting information for each page of a report. This formatting information includes page orientation (landscape or portrait), location of the beginning print line for each logical page, and the fonts to be used.

Table 6-73 summarizes the PDE command parameters.

Table 6-73. **Summary of PDE command parameters**

Parameter	Specifies	Offline	Online	DJDE
BEGIN	Location of starting print line on page	Y	Y	Y
FONTS	Fonts for printing variable and CME data	Y	Y	Y
PMODE	Landscape or portrait printing mode	Y	Y	Y

You must specify an identifier (of type ac) when defining the PDE and reference it with the FORMAT parameter of the OUTPUT command. The PDE command must precede the OUTPUT command.

PDEs may be coded as part of the JDL or created as separate files so that they may be referenced by one or more JDLs or by DJDEs. PDEs are called out on the FORMAT parameter of the OUTPUT command or a DJDE FORMAT parameter. PDEs must be precompiled to be accessed by a DJDE. Standard PDE specifications are provided on the operating system software (OSS) system tapes as described in table 6-74.

A PDE created as a separate disk file may be used as if it were part of the JDL that references it. This is done by creating a JSL file containing only PDE commands and using the PDL processor to compile it. For each PDE command, PDL creates a control file on disk, cataloged in the PDE directory. The PDE command identifier is used as the file name. When the PDE is referenced, the system searches the PDE directory for the named PDE and, if found, loads it into memory for use in processing the report. Refer to the "Compiling the JSL" section in the chapter "Creating a job source library (JSL)" for further information on this feature.

Table 6-74. Standard LPS print format

PDE id	No. of lines	No. of columns	lpi	cpi	Approx. point size	Page size and orientation*	BEGIN values	Default font id
FMT1	66	132	8.1	13.6	9	11 by 8.5	(.18,.66)	L0112B
FMT2	66	150	8.1	15	9	11 by 8.5	(.18,.50)	L0212A
FMT3	88	132	10.7	13.6	7	11 by 8.5	(.14,.66)	L0312A
FMT4	88	150	10.7	15	7	11 by 8.5	(.14,.50)	L0412A
FMT5	49	100	6	10	12	11 by 8.5	(.17,.50)	L0512A
FMT6	80	100	8.1	13.6	9	8.5 by 11	(.57,.58)	P0612A
FMT7	60	90	6	12	12	8.5 by 11	(.50,.50)	P07TYA
FMT8	60	75	6	10	12	8.5 by 11	(.50,.50)	P0812A
FMT9	80	200	10.0	20.0	7	11 by 8.5	(.25,.25)	L0912A
FMT10	132	132	12.5	17.6	6	8.5 by 11	(.22,.51)	P1012A
FMT11	132	150	12.5	20.0	6	8.5 by 11	(.22,.50)	P1112A
FMT12	66	172	8.1	13.6	9	14 by 8.5	(.18,.66)	L0112B
FMT13	104	100	8.1	13.6	9	8.5 by 14	(.57,.58)	P0612A
FMT1A	66	132	8.3	12.5	9	11.69 by 8.27	(.18,.57)	R112BL**
FMT2A	66	150	8.3	14.3	9	11.69 by 8.27	(.18,.60)	R212BL**
FMT3A	88	132	11.1	12.5	7	11.69 by 8.27	(.18,.57)	R312BL**
FMT4A	88	150	11.1	14.3	7	11.69 by 8.27	(.18,.60)	R412BL**
FMT5A	48	100	6	10	12	11.69 by 8.27	(.22,.85)	R512BL**
FMT6A	80	100	8.1	13.6	9	8.27 by 11.69	(.91,.46)	R612BP**
FMT7A	60	90	6	12	12	8.27 by 11.69	(.85,.39)	R7TIBP**
FMT8A	60	75	6	10	12	8.27 by 11.69	(.85,.39)	R812BP**
FMT9A	80	200	10.0	20.0	7	11.69 by 8.27	(.14,.85)	R912BL**
FMT10A	132	132	12.5	17.6	6	8.27 by 11.69	(.57,.39)	RA12BP**
FMT11A	132	150	12.5	20.0	6	8.27 by 11.69	(.57,.39)	RB12BP**
*First dimension given is the horizontal dimension. Thus, 11 by 8.5 is landscape orientation, while 8.5 by 11 is portrait orientation. Dimensions are given in inches.								
**A4 European international formats.								

Multiple logical pages on a physical page

Multiple BEGINs define multiple logical pages (a user-defined page image bordered by top of form and bottom of form, left and right margins) on one physical page (one side of a sheet). Up to 63 logical pages may be defined per physical page. These logical pages may be defined in any order on the physical page and placed on the physical page in the order the BEGIN parameters appear in the PDE command. The first BEGIN specified, whatever its physical position on the page, is considered logical page one. Structure your variable data in the same order that the logical pages are defined, using either spacing or skipping printer carriage controls to move from one logical page to the next. Generally, a skip-to-channel-1 parameter is the easiest way to position to the next logical page. The following should be noted:

- Each online banner page is positioned as the first logical page of a new physical sheet (if BANNER command parameter TYPE=BANNER).
- ROFFSET causes the logical page containing the matching criteria to be the first logical page of a new physical sheet.
- Page numbering occurs on each logical page rather than each physical page.
- Commands that specify line numbers, such as CRITERIA, VFU, CME, and NUMBER, refer to the line on the current logical page (ranging from TOF to BOF).
- RTEXT is imaged on a separate physical sheet. Line and column for positioning of the text refers to the first logical page specified in the PDE.
- RAUX criteria found on any logical page on the physical page causes that sheet to be picked from the auxiliary tray.
- Accounting statistics are accumulated on the basis of physical pages.
- If any logical page overflows the physical page before encountering BOF, the next line is positioned to the first logical page of the next side, and all subsequent logical pages are repositioned on BEGIN.
- With the exception of DEPT and SHIFT, all page-oriented DJDEs are applied at a logical page boundary.
- The following DJDEs, when applied, cause the current page position to go to the first logical page of the front side of a new physical sheet:

DUPLEX
COPIES
JDL
JDE
BFORM
OTEXT
RTEXT

- The following DJDEs when applied cause the current page position to go to the first logical page of the next physical side:

FORMS
FORMAT

The following sections describe the syntax of the command parameters and explanations of the parameter options.

BEGIN

This parameter specifies the location of the starting print line of a logical page for graphics. The *vpos* parameter specifies the vertical position of the first character of the first print line on the logical page. It may be specified in inches (IN) or centimeters (CM). The default is inches.

Syntax *ac:PDE BEGIN = option(s)*

Options Table 6-75 lists the parameter option(s) and definition(s).

Table 6-75. **BEGIN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(vpos, hpos)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>vpos</i> Specifies a decimal number with up to three digits to the right of the decimal point, such as 0.563 IN and 2.35 CM, are all legal specifications). The default is .18 IN. • <i>hpos</i> Specifies the horizontal position of the first character of the first print line on the logical page (in IN or CM as for <i>vpos</i>). All specifications are rounded to the nearest dot (1/300 of an inch) for positioning of the logical page. The default is .66.

Considerations In specifying the location of the beginning of a print line on the logical page, measurement is performed by viewing the page in the mode (landscape or portrait) in which it is to be printed. There may be more than one logical page per physical page; these are defined by the use of multiple BEGIN parameters. If no BEGIN parameter is specified, then the default is FMT1.

FONTS

This parameter specifies the fonts to be used in printing variable input data and CME data.

Syntax *ac:PDE FONTS = option(s)*

Options Table 6-76 lists the parameter option(s) and definition(s).

Table 6-76. **FONTS parameter option(s) and definition(s)**

Option(s)	Definition(s)
$(f_1, f_2, \dots) $ $((f_1, s_1), (f_2, s_2), \dots) $ $((f_1, s_1, [LPI XDOTS DOTS]), (f_2, s_2, [LPI XDOTS DOTS]), \dots)$	This option has the following components: <ul style="list-style-type: none"> <i>f</i> Specifies a one-to-six alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability). <i>s</i> Specifies an optional override line-spacing value defined by either the print lines per inch (LPI) or the dots per print line (DOTS) to be associated with the font. Maximum value for si is 30 LPI; minimum value is 10 dots. If unit (LPI or DOTS) is not specified, LPI is assumed by default.

There is no default for LPI for s_1 .

Considerations If an override line spacing value is specified, lines printed using the font cause the indicated line spacing to be performed after the line using the font. If different fonts are used on the same print line, the line spacing value specified for the font of the largest character in the line is used to determine the position of the next print line.

PMODE

This parameter specifies the printing mode for each physical sheet.

Syntax *ac:PDE PMODE = option(s)*

Options Table 6-77 lists the parameter option(s) and definition(s).

Table 6-77. **PMODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
LANDSCAPE	Indicates that printing is to be parallel to the long edge of paper. The default is LANDSCAPE.
PORTRAIT	Indicates that printing is to be parallel to the narrow edge of paper.

Points to note

Note the following when using the PDE command:

- The maximum number of fonts that may be used in printing a given page or a given job varies depending on a number of factors. For both the input and output tasks, the number of fonts specified in the FONTS parameter must be at least equal to the largest number of fonts required to print any page within the job.

Particularly for the input task, the limitations of dynamically allocated memory, (that is, when input is unable to receive an allocation of dynamic memory, the job is aborted), impose additional constraints which effectively reduce the maximum number of fonts that may be used to print a page. The number of fonts specified in the FONTS parameter is used to determine the size of a dynamically allocated table where information about the fonts is cached.

If the patch to enable the use of Mergenthaler fonts is selected when the LPS is sysgened, the use of dynamic memory for fonts increases significantly since a proportionally spaced font requires an additional 120-260 bytes of dynamic memory.

The maximum block length for an offline job can be a size such that the allocation of one input buffer or two of equal size can cause dynamic memory allocation problems when used in conjunction with a large PDE FONTS parameter and with Mergenthaler fonts enabled.

- An xdot is 1/600 unit of measurement. A form specifying xdots, however, can be created, edited, and compiled on any LPS with version 3 software, but the form, which is 600 spots per inch (spi), does not print on your 4050, 4090, 4635, or Xerox DocuPrint 96/DocuPrint 96MX which accepts only 300 spi.

Examples

This section shows examples of using the PDE command.

Example 1

```
PDE1: PDE PMODE=LANDSCAPE,BEGIN=(.861,.7),
      FONTS=(L0112B, L01BOB);

PDE2: PDE PMODE=PORTRAIT,BEGIN=(1.3,.37),
      FONTS=((P08TYA,6.8),(P080AA,6.8));

L1:   JDE;
      OUTPUT FORMAT=PDE1;

P1:   JDE;
      OUTPUT FORMAT=PDE2;
```

Two PDEs are defined and referenced in separate JDEs. PDE1 specifies a landscape page and two landscape fonts; PDE2 defines a portrait page and 2 portrait fonts with override line spacing.

Example 2

```
2UP: PDE BEGIN=(.5 IN,.5 IN),BEGIN=(6.5 IN,.5 IN),
      PMODE=PORTRAIT,FONTS=P1012B;
```

Example of multiple BEGINS. Two logical pages are defined on one physical page.

Example 3

```
PDE99:PDE PMODE=PORTRAIT,BEGIN=(.10,0),
      FONTS=(P07ITA,P080AA);END;
```

Example of a PDE that may be compiled by PDL separately from a JDL. After compilation, a PDE object file is created in the PDE directory which then may be referenced from within a JDL or by a DJDE. The name of the object file in the PDE directory is PDE99 (the same as its identifier on the source command).

ROUTE command

The ROUTE command allows the user to print identifying information (text and an optional form) on the page preceding a report. The ROUTE command may be coded within a JDL or in a disk file of ROUTE commands.

Table 6-78 summarizes the ROUTE command parameters.

Table 6-78. **Summary of ROUTE command parameters**

Parameter	Specifies	Offline	Online	DJDE
RFORM	Form to be printed with RTEXT preceding a report copy.	Y	Y	Y
RTEXT	Text to be printed preceding a report copy.	Y	Y	Y

Cataloged RTEXT files

When there are a large number of RTEXT parameters, they should be precompiled and stored in a disk file instead of including them in a JDL. After they are compiled, they may be accessed (by referring to their object file name) from either a JDL or a DJDE. RTEXT parameters must be precompiled to be accessed with a DJDE.

To catalog RTEXT parameters, an identifier (of the type ac) must be included on the ROUTE command. This identifier is used to reference the RTEXT parameters with the RTEXT=rtext-id parameter of the ROUTE command. An example of cataloged RTEXT usage is illustrated below. If the cataloged RTEXT is defined within a JSL, it is also automatically applied to that JSL.

```
ROUTE1:  ROUTE  RTEXT=( 'USER 1' , 1 , 33 , 64 ) ,
          RTEXT=( 'BLDG 1' , 1 , 34 , 64 ) ,
          RTEXT=( 'USER 2' , 2 , 33 , 64 ) ,
          RTEXT=( 'BLDG 2' , 2 , 34 , 64 ) ;
```

After the above commands are compiled by PDL, a file is created under the TST file directory, such as ROUTE1.TST. This file may then be accessed by name with the ROUTE command (RTEXT=rtext-id) from within a JDL (or with a DJDE). For this example the RTEXT parameter would be RTEXT=ROUTE1. RFORM, if needed, must be specified along with the reference to the RTEXT file (RFORM is never cataloged with RTEXT). This allows the user to select different RFORMs in different JDEs with the same RTEXT information.

The following sections describe the syntax of the command parameters and explanations of the parameter options.

RFORM

This parameter specifies a form to be printed with all RTEXT pages. If RTEXT data is not specified, RFORM is not honored. However, RTEXT can be specified as one blank character (or space) to print an RFORM on the routing page without any accompanying text.

Syntax ROUTE RFORM = *option(s)*

Options Table 6-79 lists the parameter option(s) and definition(s).

Table 6-79. **RFORM parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>form-id</i>	This is the name of a file cataloged in the FRM directory. It is created by compiling a file of forms source commands, called a JSL, with the FDL compiler. (Refer to the <i>Xerox DocuPrint 96/DocuPrint 96MX LPS Forms Creation Guide</i> .) The RFORM parameter is not allowed in a ROUTE command which is being defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file. Refer to the "Cataloged RTEXT files" section in this chapter for further details.
NONE	Specifies that no form is to be printed. The default is NONE.

RTEXT

This parameter specifies text to be printed on a separate page preceding a report (or copy ply).

Syntax ROUTE RTEXT = *option(s)*

Options Table 6-80 lists the parameter option(s) and definition(s).

Table 6-80. **RTEXT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>sc</i> (<i>sc</i>, [<i>passnum</i> ALL], [<i>line</i>, [<i>col</i> [, <i>fontindex</i>]]])</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>sc</i> Specifies the message to be printed (1 to 132 characters). It is printed with the first font specified in the FONTS parameter of the PDE command (refer to fontindex). • <i>passnum</i> May be either an integer number or the keyword ALL. An integer number specifies the pass (copy ply) to which the text applies. • ALL Specifies that the text applies to all copies. If there is more than one RTEXT parameter in a DJDE packet, passnum must be in increasing numeric order. The default is passnum = 1. • <i>line</i> Specifies the line number on which the first line of a block of RTEXT message is printed. The default is line 1 for the first text string of the pass. Otherwise, the default is the next line of the page. Only one RTEXT parameter may be specified per line for a passnum. • <i>col</i> Specifies the column number at which the first character of a block of RTEXT messages is to be printed. The default is column 1. • <i>fontindex</i> Identifies the index (starting with 1) of the PDE font with which the text is printed.
<p><i>rtext-id</i></p>	<p>This option is used to reference a cataloged file of RTEXT parameters previously compiled by PDL.</p>
<p>NONE</p>	<p>Specifies that no text is to be printed on a separate page. The default is NONE.</p>

Considerations The font index is associated only with a particular string. To print an entire RTEXT page in the same font, the font index must be given with each string. Strings without a font index are printed with the first alphanumeric font (specified in the font parameter of the PDE command).

If more than one font is used to print any number of RTEXT strings on a page, the line and the character spacing values of the different fonts are used to place RTEXT on the page.

If a proportional pitch font is used, the starting column number for printing the RTEXT is determined using the character spacing value for the space character in the character font.

If RTEXT is specified along with COVER=(FRONT,SEP) or BOTH,SEP, the RTEXT pages are printed on the covers. Front covers are picked only on copies for which there is RTEXT. If RTEXT is specified in non-collate mode, only the routing page for copy one is printed.

Example

Following is an example of how to use the ROUTE command:

```
ROUTE RTEXT=( 'USER 1' , 1 , 33 , 64 ) , RTEXT=( 'USER  
2' , 2 , 33 , 64 ) ;
```

The text USER 1 and USER 2 is to be printed in the center of a page preceding the respective copies of a two-copy report. The page is 132 columns by 66 lines.

STOCKSET command

The STOCKSET command defines a set of stocks used in a report. This allows output to verify the exclusivity of all stocks in terms of trays currently assigned to them. For your LPS, a stock-name is represented by a cluster. For more information refer to the feeder tray chapter in your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operator Guide*.

Stocksets are also the means of associating stock references with stock names.

An identifier (of the type ac) is specified to name the STOCKSET command for future reference by the STOCKS parameter of an OUTPUT command.

Table 6-81 summarizes the STOCKSET command parameters.

Table 6-81. **Summary of STOCKSET command parameters**

Parameter	Specifies	Offline	Online	DJDE
ASSIGN	Lists stock names and association of stock references with stock names	Y	Y	N
INIFEED	Specifies which stock to use in the absence of any parameters of the FEED parameter	Y	Y	N
SYSPAGE	Specifies which stock to use for system-generated messages	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ASSIGN

This parameter associates stocks with the identified stockset.

Syntax *ac:STOCKSET ASSIGN = {stock-descriptor | (stock-descriptor₁ [,stock-descriptor₂][,...])}*

Function Associates stocks with the identified stockset, where:

Options *stock-descriptor = stock=name | (stock-name,stock-reference)*

Default No default

Table 6-82 lists the parameter option(s) and definition(s).

Table 6-82. **ASSIGN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>{stock-descriptor (stock-descriptor₁ [,stock-descriptor₂][,...])}</i>	

INIFEED

This parameter specifies which stock to use in the absence of any FEED parameter. If not specified, the INIFEED parameter defaults to the first stock name specified in the ASSIGN parameter.

Syntax *ac:STOCKSET INIFEED = option(s)*

Options Table 6-83 lists the parameter option(s) and definition(s).

Table 6-83. **INIFEED parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stock-name</i>	
<i>stock-reference</i>	

The default is the first *stock-name*.

SYSPAGE

This parameter specifies which stock to use for system-generated pages, the DJDE OPRINO, PLABEL, exception, and accounting pages.

Syntax *ac:STOCKSET SYSPAGE = option(s)*

Options Table 6-84 lists the parameter option(s) and definition(s).

Table 6-84. **SYSPAGE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stock-name</i>	
<i>stock-reference</i>	
MAIN	
AUX	

There is no default.

Points to note

Note the following when using the STOCKSET command:

System-generated pages, such as with the VOLUME command PLABEL parameter or the IDEN command OPRINFO parameter, are formatted using 8.5 by 11-inch paper, unless the system has been sysgened for A4 paper. These system-generated pages are printed using the STOCKSET command SYSPAGE parameter that is in effect. If no STOCKSET parameters are in effect, cluster MAIN is used, unless overridden by an operator key in.

For example, assume that you try to print this page on 8.5 by 11-inch paper, subject to the cluster representing the SYSPAGE stock, or the operator key in. If the paper size is not available, the system attempts to print the page on a different paper size chosen from the same set of trays as would normally be used. This occurs without any warning, caution, or cycle down. If a larger paper size is available, it is used. If a larger paper size is not available, a smaller paper size is used. This is the only time a page may be printed on a paper size smaller than what was specified to format the page.

When a system-generated page is not printed on the same paper size used to format it, the positioning of the data on the physical page is not guaranteed. For example, when printing on a larger paper size, the data does not fall off the physical page, but it may not appear in the correct location. When printing on a smaller paper size, the data may fall off the physical page and appear incomplete.

Example

Following is an example of how to use the ROUTE command:

The STOCKSET command is similar to PDEs and CMEs, because it can be compiled separately. For each STOCKSET encountered before an initial JDL or SYSTEM command, PDL creates a file type STK, which contains information. The .STK file can then be referenced in multiple JSL source files.

```
BILLS : STOCKSET
```

```
ASSIGN    = ('F1302', COVER),  
ASSIGN    = ('F1415', (BODY, SUMMARY)),  
ASSIGN    = (('F6204', LATE), ('F9999', DISCON)),  
INIFEED   = COVER, /* INIFEED = 'F1302' WORKS AS WELL */  
SYSPAGE   = COVER;
```

In the example above, a stockset called BILLS is created and consists of four stocks: F1302, F1415, F6204, and F9999. Stock references are created for these four stocks; COVER for F1302, BODY and SUMMARY for F1415, LATE for F6204, and DISCON for F9999. In the absence of a specific parameter of the FEED parameter at the start of the report, INIFEED directs output to feed from the stock referred to by COVER. A billing application makes use of this stockset by coding an OUTPUT STOCKS=BILLS command in its JDE, and through successive DJDE FEED records, directs feeding the cover, body and summary, late notice, and disconnect pages from the stock referenced by COVER, BODY, SUMMARY, LATE, and DISCON, respectively. Alternatively, the DJDE FEED record could refer directly to stocks by name, such as FEED='F1302'. Note that the use of stock references is preferred over stock names.

VFU

The VFU command is used to assign output line numbers to printer carriage control channels. These line-to-channel assignments perform the same function as the printer carriage control tape on a conventional line printer. The VFU command is also used to assign line numbers to the top of form and the bottom of form. Top of form indicates the number of lines from the top (as defined by the PDE BEGIN values) of an output page to the first print line on the page. TOF defines the first line on which printing appears. Bottom of form indicates the number of lines from the top of an output page to the last print line on the page. BOF defines the last line on which printing appears.

Top and bottom of form are used for prejob page alignment and for page overflow processing. For all PCCTYPES except ANSI and user-defined PCCs, the prejob page alignment is to top of form in the expectation that the first carriage control parameter of the job will be print and space one line, or something similar. Selection of ANSI causes alignment to bottom of form to handle the skip-to-channel-1-and-print parameter, which usually begins a job of that carriage control type. User-defined PCCs may set alignment at either TOF or BOF.

Page overflow occurs when spacing to the next line causes the bottom-of-form line number to be exceeded. Page transition occurs, and line spacing is continued from the top-of-form line number. Honeywell 2000 carriage control and Xerox carriage control are exceptions to this processing (refer to the *Xerox LPS Tape Formats Manual*).

You must specify an identifier of the type ac when defining the VFU table and reference it in the VFU parameter of the LINE command. The VFU command must precede the LINE command. A command identifier of the type ac may consist of one-to-six alphanumeric characters (A-Z and 0-9). One of the characters must be alpha.

Table 6-85 summarizes the VFU command parameters.

Table 6-85. **Summary of VFU command parameters**

Parameter	Specifies	Offline	Online	DJDE
ASSIGN	Assign line numbers to output channels	Y	Y	Y
BOF	Bottom-of-form line number assignment	Y	Y	Y
TOF	Top-of-form line number assignment	Y	Y	Y

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ASSIGN

This parameter specifies the output line-to-channel assignments.

Syntax *ac:VFU ASSIGN = option(s)*

Options Table 6-86 lists the parameter option(s) and definition(s).

Table 6-86. **ASSIGN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(channo,lineno)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> <p><i>channo</i> The number of the channel being assigned. It is an integer in the range 0 to 15. The user may end the VFU command with a semicolon and start another VFU command without an id field to continue specification for the same channel or a different channel.</p> <p><i>lineno</i> The number of the output print line being assigned to a particular channel. It is an integer in the range TOF to BOF. There are no default assignments for any channel, including channels 1, 9, and 12.</p>

BOF

This parameter specifies the bottom-of-form line number

Syntax *ac:VFU BOF = option(s)*

Options Table 6-87 lists the parameter option(s) and definition(s).

Table 6-87. **BOF parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies the number of lines from the top of the output page to the last print line on the page (bottom of form). The bottom-of-form specification is independent of channel assignments. Bottom of form should be greater than or equal to the largest line number assigned to a channel. The maximum value for BOF is 255.</p> <p>The default is 66 (VFU command).</p>

TOF

This parameter specifies the top-of-form line number.

Syntax *ac:VFU TOF = option(s)*

Options Table 6-88 lists the parameter option(s) and definition(s).

Table 6-88. **TOF parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the number of lines from the top of the output page to the first print line on the page (top of form). The top-of-form specification is independent of channel assignments. It should be greater than or equal to 1 and less than or equal to the smallest line number assigned to a channel. The default is 1.

Points to note

Note the following when using the VFU command:

Any channel operation encountered during printing for an unassigned channel causes a print-and-space-1-line operation. This operation occurs even though some vendor formats normally specify the default as a space-1-line-and-print operation.

Multiple line numbers may be assigned to the same channel number. This simulates the vertical tabbing feature of an impact line printer where a skip-to-channel parameter causes transition to the next punched hole in the specified channel of the paper tape. This tape, which controls the printer, facilitates spacing a fixed number of lines down the print page. There may be multiple punches in any vertical format channel on the impact printer's tape.

A skip-to-channel parameter in the LPS causes selection of the next line number in the ASSIGN list (for that channel) larger than the current line number. Page transition and alignment to the first line number in the list occurs if no line number is larger than the current line.

Example

Following is an example of how to use the ROUTE command:

In the following example, top of form is assigned to line number 5 and bottom of form is assigned to line number 55. Channels 1, 2, and 12 have been assigned line numbers.

```
V1: VFU ASSIGN=(1,5),  
      ASSIGN=(2,(10,15,20,25,30,35,40,45,50)),  
      ASSIGN=(12,55),TOF=5,BOF=55;
```

Assume the printing system is printing a report and the current line number is 11. If a skip-to-channel-1-and-print parameter were issued, a page transition occurs. Printing begins on line 5 (assigned to channel 1), which is the top of form on the new page.

If a skip-to-channel-2-and-print parameter is issued when the current line number is 11, the next line to be printed would be line 15 of the current page. Lines 10, 15, 20, and so forth, are also assigned to channel 2, but since the current line number is 11, the next consecutive line number assigned to channel 2 (greater than 11) is line 15.

If a skip-to-channel-12-and-print parameter is issued, the next line to be printed is line 55 of the current page.

7. Using logical processing

The logical processing (sometimes referred to as “special processing”) commands enable the user to specify logical functions performed on either a record, a set of records, or on a block basis. The following table summarizes the functions that may be performed and the corresponding PDL commands.

Table 7-1 summarizes the commands associated with logical processing.

Table 7-1. **Summary of commands associated with logical processing**

Command	Function	Offline	Online	DJDE
BANNER	Banner page detection test (online only)	N	Y	N
BDELETE	Block deletion test	Y	N	N
BSELECT	Block selection test	Y	N	N
CRITERIA	Logical processing test specifications	Y	Y	N
RAUX	Select page from auxiliary tray test	Y	Y	N
RDELETE	Record deletion test	Y	N	N
RFEED	Changes cluster names on page-by-page basis without using DJDEs	Y	Y	N
ROFFSET	Page offset test	Y	Y	N
RPAGE	Logical page reposition test	Y	Y	Y
RRESUME	Resume printing test	Y	N	N
RSELECT	Record selection test	Y	N	N
RSTACK	End of report test	Y	Y	N
RSUSPEND	Suspend printing test	Y	N	N
TABLE	Table of constants for logical processing	Y	Y	N

Logical processing commands

This section explains the logical processing command format and lists the commands with TEST parameters.

Logical processing command format

To define a logical processing command fully, you must specify one or two fields in the record or block to be tested. In general, a logical processing command has the following format:

***command* TEST = *test-exp*;**

The logical processing command tests the value of the specified *test-exp* and directs the flow of processing based on the result of the test.

The *text-exp* portion of the command defines a test to be performed on either one or two specified fields and their associated constants for a true or false value. The fields in the record or block are compared with their associated set of constants using either an equal (EQ) or a not equal (NE) operator. The basic element used to describe a test for a logical function is the CRITERIA command. Refer to the CRITERIA section in this chapter for more detailed information.

Logical processing commands with TEST parameters

The available logical processing commands with TEST parameters are listed below and are described separately in the following sections. Each individual command can be specified only once per job descriptor entry. If any one command is specified more than once, the last occurrence is used without notification of any error.

Table 7-2 lists the logical processing commands with TEST parameters

Table 7-2. **Logical processing commands with TEST**

Command	Definition
BANNER	Banner page detection
BDELETE	Block selection
BSELECT	Block selection
RDELETE	Record deletion
RFEED	Change cluster names
ROFFSET	Page offsetting
RSELECT	Record selection
RSUSPEND	Printing suspension
RRESUME	Printing resumption
RSTACK	End of report
RAUX	Auxiliary tray page selection
RPAGE	Logical page repositioning

All of the above commands except BANNER are available for offline processing. BANNER, RAUX, RPAGE, ROFFSET, and RSTACK are available for online.

CRITERIA command

The basic element used to describe a test for a logical function is the CRITERIA command. Each CRITERIA command describes a field in either a record or block and the specific test to be performed. The CRITERIA command requires an identifier that can be specified in any of the logical processing commands, as a TEST parameter. The definition of the TEST parameters for logical processing commands is described in the "Test expressions" section.

Specific TEST parameters are described in the individual logical processing command sections.

A CRITERIA command may use any or all of its three parameters to specify the conditions which must be met in order for a given operation to be completed. A JSL may contain any number of CRITERIA commands which you can specify for a particular print job by calling out the appropriate CRITERIA identifier. For example, the following criteria are specified at the system level:

```
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);
```

Criteria C1 and C2 are then specified at the job level:

```
HDRP:  JOB;
BANNER TEST=C1, HCOUNT=2, TCOUNT=0;

TRLP:  JOB;
BANNER TEST=C2, HCOUNT=0, TCOUNT=3;

BOTH:  JOB;
BANNER TEST=(C1 OR C2), HCOUNT=2, TCOUNT=3;
```

Refer to "Test expressions" section for additional information.

CHANGE

This parameter defines test specifications for a logical processing function with change mode criteria.

Syntax CRITERIA CHANGE = *option(s)*

Options Table 7-3 lists the parameter option(s) and definition(s).

Table 7-3. **CHANGE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>offset</i>	The offset in bytes (relative to zero) from the start of the user's portion of the record to the control field within the record or from the start of the physical tape block to the control field within the block.
<i>length</i>	The length in bytes of the control field. Its range is 1 to 255.
NE	Indicates not equal to.
LAST	Indicates that the control field of the current record (or block) is being compared to the control field of the previous (last encountered) record (or block). If a control field of the current record (or block) is less than the specified length, the comparison is not done and the test fails. (For online and HIP jobs, where trailing blanks are truncated by the host, control fields which are less than the specified length may be padded with blanks so that a comparison can be done. If you are using a value less than the length specified, or if the number is the same but less than zero, the values are not internally the same and thus, the test is met.)

There is no default.

CONSTANT

This parameter defines test specifications for a logical processing function with constant mode criteria in block processing.

Syntax CRITERIA CONSTANT = *option(s)*

Options Table 7-4 lists the parameter option(s) and definition(s).

Table 7-4. **CONSTANT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>offset</i>	The offset in bytes (relative to zero) from the start of the physical tape block to a field within the tape block being compared to a table or string constant. In the case of record processing, it is the offset from the start of the user's portion of the record to the field in the record being compared.
<i>length</i>	The length in bytes of the test field. Its range is 1 to 255.
EQ NE	Keywords that specify the operation of equal to or not equal to.
<i>tab-id</i>	The identifier of a TABLE command.

There is no default.

LINENUM

This parameter specifies the range of consecutive line numbers for which the CRITERIA command is fully evaluated. If not specified, the default range is all lines.

Syntax CRITERIA LINENUM = *option(s)*

Options Table 7-5 lists the parameter option(s) and definition(s).

Table 7-5. **LINENUM parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>init</i>	An integer specifying the number, on each page, of the beginning line for which the CRITERIA command is evaluated.
<i>count</i>	An integer specifying the consecutive number of lines (starting with <i>init</i>) for which the CRITERIA command is evaluated.

The default is all lines.

Test expressions

The activation of testing is done by coding the TEST parameter of any logical processing command described in this chapter. Either one or two CRITERIA commands can be specified.

The following is a definition of the test expression for a logical processing command. The test definition is referred to in the syntax of each of the logical processing commands that follow:

```
TEST=cri-id;  
TEST=(cri-id1,AND,cri-id2);  
TEST=(cri-id1,OR,cri-id2);
```

The *cri-id*₁ and *cri-id*₂ are identifiers for either the change mode or constant mode CRITERIA commands discussed in this section. If only *cri-id* is present, the test is satisfied if the criteria in *cri-id* is satisfied.

If two CRITERIA commands are specified for a test, the testing of two is linked by either an AND or an OR logical operator. If AND is coded, both CRITERIA commands must be true for the same data record in order for the TEST to be successful. If OR is coded, then either CRITERIA command is satisfied for a data record, the TEST is successful. The second and third CRITERIA commands are optional; but if either is specified, the other must also be present. If *cri-id*₁ and *cri-id*₂ are both present and the keyword AND is coded, the test is true only if criteria in both *cri-id*₁ and *cri-id*₂ are satisfied. If the keyword OR is coded, the test is true if the criteria in either *cri-id*₁ or *cri-id*₂ are true. If the test is satisfied, the logical processing function is performed.

For example, the following CRITERIA command describes a test for a subfield equal to a specific constant table.

```
cri-id: CRITERIA CONSTANT = (offset,length,EQ,tab-id);
```

The CONSTANT parameter specifies that the content of a record is to be compared to the values in a table command with identifier *tab-id* (refer to the "TABLE command" section later in this chapter). The subfield to be compared is located *offset* bytes from the start of the record or block, with *length* in bytes. When the subfield in the record or block matches a constant in the table, the CRITERIA command is true and the processing called for in the parameter is initially completed or acted upon. The particular logical processing function is performed when the test expression is true as specified by the parameter.

Specifying one CRITERIA command

To complete the description of the entire test for a logical function, the TEST parameter requires a parameter that specifies either one or two CRITERIA commands. If only one test is to be performed to determine the value of a particular logical processing function, the form of the TEST parameter is as follows:

```
TEST = cri-id;
```

The *cri-id* is the identifier for the particular CRITERIA command. Parentheses in this format are optional.

Specifying two CRITERIA commands

If two CRITERIA commands are needed to determine the true or false value for a logical processing function, they may be linked by

either an AND or an OR. The formats of these TEST parameters may be as follows:

```
TEST = (cri,AND,cri-id-2);
```

or

```
TEST = (cri,OR,cri-id-2);
```

The *cri-id* and *cri-id-2* are the identifiers for two CRITERIA commands. The parentheses in this format are required.

For example, the following three commands specify that a record should be selected (RSELECT parameter) if a particular field is equal to a given string constant (sc):

```
T1: TABLE CONSTANT=(sc);
C1: CRITERIA CONSTANT=(offset,length,EQ,T1);
    RSELECT TEST=(C1);
```

There are two formats for the CRITERIA commands: constant mode and change mode.

Each CRITERIA command may be either constant mode or change mode, but not both.

Constant mode

In the constant mode, the user must specify the location, length, and contents of a fixed field within a user's record or block. Each user's record or block is examined at the specified location to determine if the constant is present (the identifier *tab-id* defines the table containing the constant). If present, the CRITERIA command is true; if not, the command is false. A constant mode CRITERIA command can be coded as follows:

```
cri-id:CRITERIA CONSTANT=(offset,length,EQ | NE,tab-id
    LINENUM=(init,count);
```

The *tab-id* is the name (or identifier) of a value TABLE.

Change mode

In change mode, the user must specify the length and location of a control field in each record or block. When the content of the control field of one record (or block) differs from the content of the control field of the previous record (or block), the CRITERIA command is true.

A change mode CRITERIA command can be coded as follows:

```
cri-id: CRITERIA
CHANGE=(offset,length,NE,LAST),
LINENUM=(init,count);
```

LINENUM parameter

The LINENUM parameter of a CRITERIA command specifies that only a subset of a report's data records are to be tested for the user-defined criteria. This subset is specified as a set of contiguous line numbers for a page. If the carriage control specification for a given record is such that the record is placed within the set of lines specified by LINENUM, the record is examined according to the CRITERIA command. Otherwise, it is not processed.

Combining change and constant modes

The CRITERIA tables may specify either change mode or constant mode functions; there are no restrictions on their usage or combination.

There are two special cases: when the record or block is too short to include the field being tested and when the line being tested contains no record, that is, it has been skipped. If the test specifies a constant mode function, the CRITERIA fails.



Note: If the test specifies a change mode function, the CRITERIA fails because no change has occurred, but the value for LAST option of the CHANGE parameter is unchanged for comparison with the next record. (See additional information under the “Criteria parameter functions” section.)

A compiler error occurs if you attempt an AND test on two different criteria which appear on two non-overlapping line ranges. For example, a logic problem occurs with ROFFSET where line 3 contains Criteria 1 and Line 2 contains Criteria 2, that is, Criteria 1 and Criteria 2 are not equal to LAST.

Points to note

Note the following when using the CRITERIA command:

When change mode CRITERIA commands are evaluated, a string from the current line is compared with a string saved from the corresponding part of a previous line.

- If the comparison is not equal, the string from the current line becomes the saved comparison string for subsequent lines and the criteria is considered true.
- If two records are overprinted on the same line but with different data in the field defined by the CRITERIA command, the string from the last overprint record becomes the saved comparison string and the CHANGE CRITERIA is satisfied.
- If the LINENUM parameter is used, lines within the specified range are processed normally, and lines outside the range are not evaluated. Thus, change mode comparison strings are not saved from lines outside of a LINENUM range.

No print records (carriage control specifies no printing but only skipping or spacing) are evaluated for logical processing in the following manner:

- If change mode is specified, no print records are evaluated.
- If constant mode is specified, no print records are evaluated.

Examples

This section shows two examples of using the CRITERIA command.

Example 1

The following are examples of CRITERIA commands including the TABLE commands they reference. The identifiers C1 and C2 are referenced by a logical processing TEST parameter.

```
T1: TABLE CONSTANT=( '$$/' );  
C1: CRITERIA CONSTANT=( 4, 4, EQ, T1 );  
T2: TABLE CONSTANT=( '//JOB ', '//EXEC' );  
C2: CRITERIA CONSTANT=( 1, 6, EQ, T2 );
```

Example 2

```
T1: TABLE CONSTANT=( 'EFGH' );  
C1: CRITERIA  
CONSTANT=( 104, 4, NE, T1 ), LINENUM=( 10, 6 );  
RDELETE TEST=( C1 );
```

The LINENUM specification (10,6) means that lines 10 through 15 on each page are being examined for the RDELETE test. Any line not in that range is not examined.

String comparison concepts

This section focuses on string comparisons, character types, and masked comparisons using default and non-default type assignments.

String comparisons

String comparisons for logical processing are specified using the CRITERIA and TABLE commands. These comparisons test for one of the following conditions:

- Absolute equality—tested for by either a change mode CRITERIA command or by a constant mode CRITERIA command which references a TABLE command that does not have a MASK parameter coded (refer to processing sequence 1 in figure 7-1).
- Equality under a mask—tested for when the referenced TABLE command has a MASK parameter coded (refer to processing sequence 2 in figure 7-1).

Comparing strings for equality under a mask means that the user has specified, for each character position of the input data string, that the test performed is one of the following:

Compare the character in that position for absolute equality.

Ignore the character in that position (consider the character, whatever it is, to compare as equal).

Compare the character in that position for type, for example, alphabetic, numeric, or some specially defined type.

Character types

An attribute called type may be associated with any character(s) in a character set. The possible types are identified by the integers 1 through 7. Any single character in a set, for example (), will either have a type (type 1, type 2, type 3, and so forth.) or not. If it has no type, it is referred to as untyped. It is possible for any character to be untyped, have one type, or have multiple types, for example, be type 2 and type 3.

The defining of types, that is, the associating of type numbers with any group of characters in a character set, is described in the "Character type assignments—TCODE command" section of the "Specifying input parameters" chapter.

For every standard character set, there is a set of standard default type assignments which, for most applications, is sufficient without modification. These assignments define the numeric characters 0-9 to be type 1 characters and the uppercase and lowercase alphabetic characters A-Z and a-z to be type 2 characters. Use of these default type assignments can be invoked by the TCODE parameter of the VOLUME command.

Masked comparisons using default type assignments

Performing masked comparisons using unmodified standard default type assignments requires coding the following:

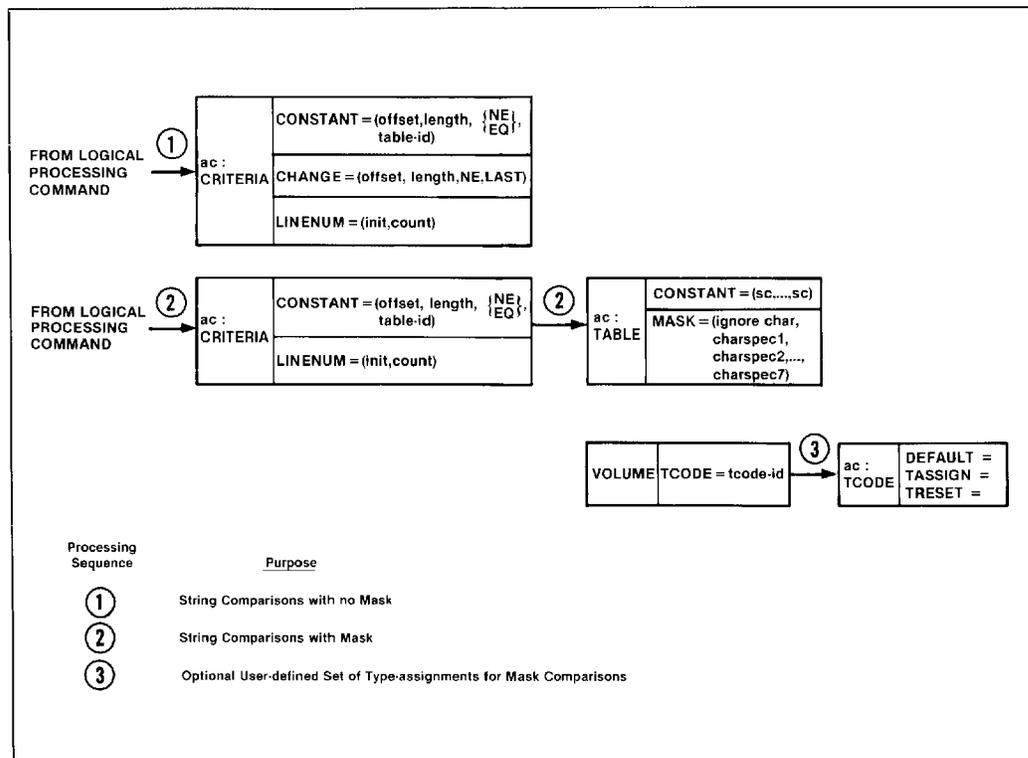
- **TCODE command.** Used either to modify a set of standard default type assignments or to define a completely new set (refer to processing sequence 3 in figure 7-1). Refer to the “Character type assignments” section in the chapter “Specifying input parameters,” for detailed information on the TCODE command.
- **TCODE parameter of the VOLUME command.** Allows you to select a set of standard default type assignments.
- **MASK and CONSTANT parameters of one or more occurrences of the TABLE command.** Together, these parameters define exactly how the comparisons for that TABLE command are made using the set of type assignments specified by the TCODE parameter of the VOLUME command.

Masked comparisons using non-default type assignments

Performing masked comparisons using either a nonstandard character set or a standard set for which the default type assignments are not suitable requires coding the following:

- TCODE command.** This command is used either to modify a set of standard default type assignments or to define a completely new set (refer to processing sequence 3 in figure 7-1). Refer to the "Character type assignments" section in chapter "Specifying input parameters," for detailed information on the TCODE command.
- TCODE parameter of the VOLUME command.** This parameter is used to specify the set of assignments defined by the TCODE command as the one used for masked comparison.
- The MASK and CONSTANT parameters of one or more occurrences of the TABLE command.** The use of these parameters is the same as for comparisons using unmodified defaults.

Figure 7-1. String comparison parameters



BANNER command

The stacked reports feature enables the online and offline user to define a series of reports in a single file. This is accomplished by specifying an end-of-report condition in the coded logical processing commands RSTACK (online and offline) and BANNER (online use only). End-of-report is that point in processing a report when all of the pages of a copy of a report have been formatted to disk and processing has begun on the next report.

Reports are stacked in a file if more than one report is included in a single file and separated from each other logically but not physically (with tapemarks, operating system labels, and so forth). In processing stacked reports, the system checks each record for the logical end-of-report specification as defined by the TEST parameter of the RSTACK and BANNER commands. When BANNER is coded, the user specifies the number of consecutive banner pages which must satisfy the test criteria before end-of-report occurs.

Report separation based on banner page detection is performed with the BANNER command. This includes a test expression to detect a banner page and a count of the header or trailer pages used to separate reports. Table 7-3 lists the function and use of each BANNER parameter.

Table 7-6 summarizes the BANNER command parameters.

Table 7-6. **Summary of BANNER command parameters**

Parameter	Specifies	Offline	Online	DJDE
HCOUNT	The total number of consecutive header banner pages.	N	Y	N
HJOBNO	Subfield of the first record to satisfy banner selection criteria be displayed as a CUST JOB number.	N	Y	N
HRPTNA	Subfield of the first record to satisfy banner selection criteria to be displayed as a REPORT NAME.	N	Y	N
TCOUNT	Total number of consecutive trailer banner pages needed for banner page detection.	N	Y	N
TEST	Test expression for banner page detection.	N	Y	N
TYPE	Use of selected data pages as banner pages when reports do not contain this information.	N	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

HCOUNT

This parameter specifies the total number of consecutive header banner pages which must be detected to satisfy the test expression.

Syntax BANNER HCOUNT = *option(s)*

Options Table 7-7 lists the parameter option(s) and definition(s).

Table 7-7. **HCOUNT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the total number of consecutive header banner pages which must be detected to satisfy the test expression. The default is 0.

HJOBNO

This parameter specifies that for each report a subfield of the first record satisfying the banner selection criteria is to be displayed as CUST JOB # on the LPS system controller in response to the JOBS command or STATUS function key.

Syntax BANNER HJOBNO = *option(s)*

Options Table 7-8 lists the parameter option(s) and definition(s).

Table 7-8. **HJOBNO parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(offset, length)</i>	This option has the following components: <ul style="list-style-type: none"> • <i>offset</i> Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record. • <i>length</i> The number of bytes in the subfield (0 to 6).
NONE	Specifies that no HOST job number is to be selected. The default is NONE.

HRPTNA

This parameter specifies that for each report a subfield of the first record satisfying the banner selection criteria is to be displayed as the REPORT NAME on the LPS system controller in response to the JOBS command or STATUS function key. (For jobs with HCOUNT=0, that is, trailers only, the report name is associated with the report that precedes the trailer banner page.

Syntax BANNER HRPTNA = *option(s)*

Options Table 7-9 lists the parameter option(s) and definition(s).

Table 7-9. **HRPTNA parameter option(s) and definition(s)**

<i>(offset, length)</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>offset</i> Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record. • <i>length</i> The number of bytes in the subfield (1 to 16).
NONE	<p>Specifies that no report name is to be selected. The default is NONE.</p>

TCOUNT

This parameter specifies the total number of consecutive trailer banner pages which must be detected to satisfy the test expression.

Syntax BANNER TCOUNT = *option(s)*

Options Table 7-10 lists the parameter option(s) and definition(s).

Table 7-10. **TCOUNT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies the total number of consecutive trailer banner pages which must be detected to satisfy the test expression. The default is 0.</p>

TEST

For online jobs, this parameter defines the test expression for detection of a banner page for either change mode or constant mode criteria.

Syntax BANNER TEST = *option(s)*

Options Table 7-11 lists the parameter option(s) and definition(s).

Table 7-11. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. If a test-exp is satisfied, the page containing the record tested is considered a banner page. There is no default.

TYPE

This parameter specifies the use of selected data pages as banner pages when reports do not contain this information

Syntax BANNER TYPE = *option(s)*

Options Table 7-12 lists the parameter option(s) and definition(s).

Table 7-12. **TYPE parameter option(s) and definition(s)**

Option(s)	Definition(s)
DATA	For online jobs, use when reports do not contain user-specified banner pages. This parameter allows the user to treat selected data page(s) as banner pages for report separation. This type of banner page is printed with a form (if specified). In duplex mode, this type of banner page is printed as a duplex page with data on the back. If multiple PDE begins are specified and TYPE=DATA, only the first header page of the report is repositioned to the first logical page of a new physical sheet.
BANNER	Should be coded when reports contain user-specified banner pages. These banner pages are printed without forms. In duplex mode, the banner pages are printed with blank backs. The default is BANNER.

Points to note

Table 7-13 shows the effect of the TCOUNT and HCOUNT parameters of the BANNER command under the conditions listed and the results of each.

Table 7-13. Summary of BANNER command parameter conditions and results

Condition	Parameter	Result
LPS positions to TOF after end-of-report processing	If TCOUNT = 0 If TCOUNT not 0	The page after the first header is positioned at TOF. The page after the last trailer is positioned at TOF.
Running online with header pages only as the offset criteria	TCOUNT = 0	Job parameters such as DJDEs and page numbering for a given report are applied to the first header page of the following report.
The system encounters a non-banner page	TCOUNT = 0	The next banner page is treated as the first header of a new report, even if HCOUNT is not satisfied.
The system encounters a non-banner page before HCOUNT is satisfied	HCOUNT not 0 and TCOUNT not 0	LPS assumes that the report has fewer header pages than were specified and begins looking for trailer pages.

Examples

This section shows examples of using the BANNER command.

Example 1

The following commands define the BANNER criteria for a job stream with two trailer pages and one header page. A character string of 120 asterisks beginning in print position one of line 66 occurs on header and trailer pages.

```
T1: TABLE CONSTANT=(120)'*';
C1: CRITERIA CONSTANT=(0,120,EQ,T1),
    LINENUM=(66,1);
    BANNER TEST=C1, HCOUNT=1, TCOUNT=2,
    HJOBNO=(121,6);
```

Example 2

If a job stream has two trailer pages with the character string DATE=mm/dd/yy appearing on line 10 or 11 and beginning in print position 20, the BANNER criteria could be coded as follows:

```
T2: TABLE CONSTANT=('DATE=::/::/::'), MASK=('::');
C2: CRITERIA CONSTANT=(19,13,EQ,T2),
    LINENUM=(10,2);
    BANNER TEST=C2, HCOUNT=0, TCOUNT=2;
```

The MASK parameter specifies a masking character. The CONSTANT parameter specifies both the constant data being compared and the character positions which contain variable data which are not being compared. In this example, positions 1 through 5 (DATE=) as well as positions 8 and 11 (the /'s) of the specified character string contain fixed data. Positions 6, 7, 9, 10, 12, and 13, which are identified by the mask character (:), contain variable data and are not compared. Header page count has been set to zero because criteria defining header pages have not been specified.

BSELECT and BDELETE commands

Interspersed blocks within one offline report or file may be either selected for or deleted from printing by use of the BDELETE and BSELECT commands.

TEST—BSELECT and BDELETE commands

These commands can also be used selectively to delete specialized blocks, for example, control blocks and unsupported labels, that are on the data tape but are not to be printed. Syntax for the TEST parameter of BSELECT and BDELETE commands is as follows.

TEST

For online jobs, this parameter defines test expression for selecting blocks for printing.

Syntax TEST = *option(s)*

Options Table 7-14 lists the parameter option(s) and definition(s).

Table 7-14. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. The block is selected (BSELECT) for printing or deleted (BDELETE) if a <i>test-exp</i> is satisfied. There is no default.

Points to note

Note the following when using the BSELECT and BDELETE commands:

- It is important to note that block selection or block deletion is performed before the extraction of the records from the block. If a block is deleted from or not selected for printing, none of the records contained within that block are processed or are available for any other logical processing functions.
- If a block does not match the same format as the normal blocks, it can be deleted and thus not cause a processing error. For example, a control block in a fixed blocked file may cause a processing error unless it is first deleted.
- When defining TABLE CONSTANTS or CHANGE parameters, offsets to subfields of a block specify in bytes, relative to zero from the start of block to the beginning of subfield.

Examples

This section shows examples of using the BSELECT and BDELETE commands.

Example 1

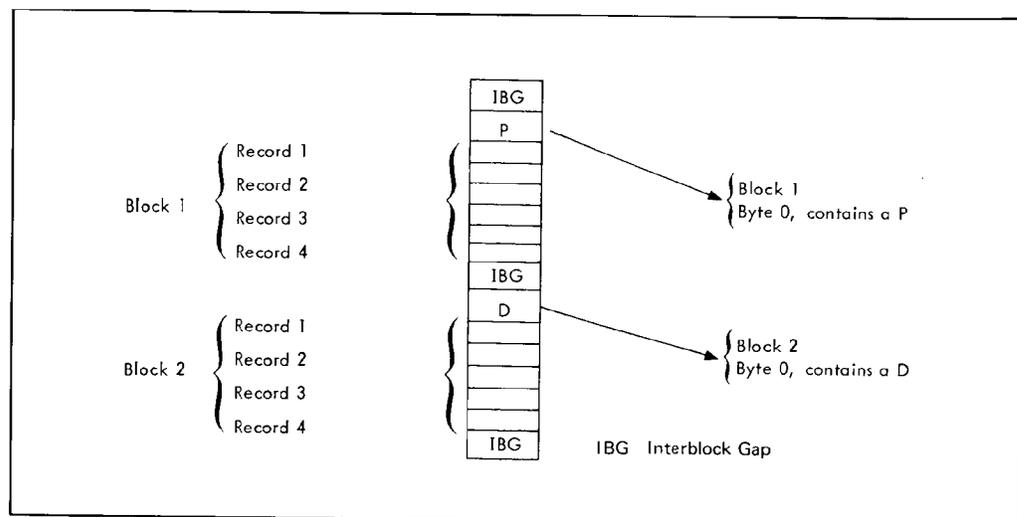
The following commands illustrate the use of BSELECT to process interspersed reports (refer to the following figure) on a block basis.

```
T1: TABLE CONSTANT=('P');
C1: CRITERIA CONSTANT=(0,1,EQ,T1);
    BSELECT TEST=(C1);
```

The contents of the first byte of each tape block (offset=0, length=1) is examined for the character constant 'P'. When a 'P' is detected, the entire block is selected (BSELECT) for printing. When the first byte of any block does not contain a 'P', that block is bypassed and not printed. In this example, only block 1 is printed.

Figure 7-2 shows a sample of how BSELECT and BDELETE commands are used.

Figure 7-2. Sample BSELECT and BDELETE command usage



Example 2

The BDELETE command to delete the blocks on the tape that have \$\$11 is shown below:

```
T1: TABLE CONSTANT=('$$11');
C1: CRITERIA CONSTANT=(4,4,EQ,T1);
    BDELETE TEST=C1;
```

RAUX command

Feeding a single sheet of paper from the auxiliary tray may be controlled from within the input data stream. If a data record satisfying the RAUX test criteria is found, the page on which the record is found is printed on a sheet of paper picked from the auxiliary tray. For simplex printing, the next page is printed on a sheet of paper selected from the main tray, unless it also contains a record satisfying the RAUX test criterion. For duplex printing, the need to pick a sheet from the auxiliary tray (as the result of detecting the specified RAUX criterion on either the front or back side of a page) is ascertained on a page-pair basis before the first side is printed. The FEED MAIN operator command (refer to chapters 5 and 6 of this manual) should be used with this parameter to prevent the auxiliary tray from being automatically selected when the main tray is emptied.

This parameter can also be used in combination with the PDL parameter FEED of the OUTPUT command (refer to the chapter "Print format commands") and the FEED DJDE command (refer to the chapter "Specifying dynamic job descriptor entries (DJDEs)").

The following section describes the syntax of the command parameters and explanations of the parameter options.

TEST

This parameter specifies the test expression for selection of paper from the auxiliary tray.

Syntax RAUX TEST = *option(s)*

Options Table 7-15 lists the parameter option(s) and definition(s).

Table 7-15. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. If a <i>test-exp</i> is satisfied, a page is selected from the auxiliary tray. There is no default.

RFEED command

The logical processing RFEED command allows you to change cluster names on a page-by-page basis without using DJDEs. The RFEED command allows you to specify paper fed from different clusters if certain criteria are met on a record basis. RFEED is essentially an extension of the RAUX command in that it allows you to specify virtually any cluster-name or cluster-reference without being limited to the AUX cluster.

The following section describes the syntax of the command parameters and explanations of the parameter options.

TEST

This parameter allows you to specify paper fed from different clusters if certain criteria are met on a record basis.

Syntax RFEED TEST = *option(s)*

Options Table 7-16 lists the parameter option(s) and definition(s).

Table 7-16. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>((test-exp, clu-def)[,...])</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> <i>test-exp</i> References a criteria table to be used. <i>clu-def</i> Defines the cluster to use for the page, (either a cluster-name or a cluster-reference).

There is no default.

Points to note

Note the following when using the RFEED command:

- Satisfaction of the RFEED criteria will cause the current page to feed from the cluster defined in the RFEED command.
- If a criteria is met, the cluster name will remain in effect until the next RFEED criteria is met or a new report is processed.
- RFEED is not available as a DJDE.
- In a multiple RFEED criteria command, the order in which the criteria are specified makes a difference in the job application. Once a criterion is satisfied, the system ignores criteria checking for the rest of the page. The criteria checking resumes when a page transition occurs. Consider this example:

```
PAPERSTK:
    STOCKSET ASSIGN = ('RED', red-ref),
              ASSIGN = ('BLUE', blue-ref),
              ASSIGN = ('GREEN', green-ref);
T1:  TABLE  CONSTANT = ('ABC');
T2:  TABLE  CONSTANT = ('DEF');
T3:  TABLE  CONSTANT = ('GHI');
C1:  CRITERIA CONSTANT = (4,3,EQ,T1);
C2:  CRITERIA CONSTANT = (4,3,EQ,T2);
C3:  CRITERIA CONSTANT = (15,3,EQ,T3);
RFEED TEST = ((C1,'RED'), (C2,'BLUE'),
              (C3,'GREEN'));
```

In this example, the input task checks C1, C2, and then C3. If C2 and C3 satisfy the specified criteria, the C2 cluster has priority over the C3 cluster due to its position in the JSL.

RSELECT and RDELETE commands

Interspersed records within one report or file may either be deleted from or selected for printing by use of the RDELETE and RSELECT commands. These commands can also selectively delete specialized records, for example, control records and offset records that are on the data tape but are not to be printed.

The following section describes the syntax of the command parameters and explanations of the parameter options.

TEST

This parameter defines test expression for selection (RSELECT) or deletion (RDELETE) or records for printing.

Syntax RFEED TEST = *option(s)*

Options Table 7-17 lists the parameter option(s) and definition(s).

Table 7-17. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. If a test-exp is satisfied, the record is selected for printing. If the RDELETE test-exp is satisfied, the record is deleted from the printed output.

There is no default.

Points to note

Note the following when using the RSELECT and RDELETE commands:

- When defining TABLE CONSTANTS or CHANGE parameters, offsets to the subfields of the records are specified in bytes, relative to zero from the start of the user's portion of the record to the beginning of the subfield.
- Record selection or deletion is performed prior to RAUX and suspend or resume, offline. If a record satisfying either the RAUX suspend or resume test criteria was previously not selected for, or deleted from printing, neither the RAUX nor suspend nor resume functions.

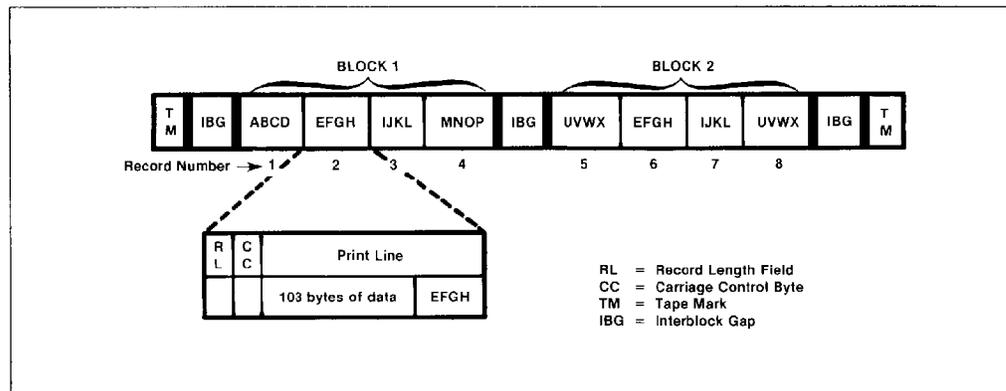
Example

The following commands illustrate the use of RDELETE to process interspersed reports on a record basis (refer to figure 7-4).

```
T1: TABLE CONSTANT=('EFGH');
C1: CRITERIA CONSTANT=(104,4,EQ,T1);
RDELETE TEST=C1;
```

If the contents of a deletion control field located 104 bytes from the start of the user portion of the record are equal to the constant 'EFGH', the record is not printed. In the following example, records 2 and 6 would not be printed.

Figure 7-4. Sample RDELETE command usage



ROFFSET command

The logical processing function ROFFSET provides the capability for online and offline users to initiate a page offset in the stacker tray under control of the input data, that is, data and DJDE records. These special user-controlled offsets can be used to simplify job distribution by creating separate stacks for each distribution entity. No other special processing occurs as a result of the ROFFSET test being satisfied; for instance, the report is not terminated nor are multiple copies produced at the offset juncture.

In duplex mode, ROFFSET forces the logical page on which the criteria are satisfied to be the first logical page of a new sheet.

The page on which the offset occurs can be determined by the following criteria:

- If the record which satisfies the ROFFSET test is printed on a page of the output, that page is the offset sheet.
- If the record which satisfies the ROFFSET test is not printed on a page of the output, that is, it is deleted by RDELETE but not BDELETE, the ROFFSET function is performed for the next printable record. If the next printable record causes a transition to the next page, the next page is offset.

ROFFSET also provides the control to force an offset on either all copies of the report or only the first copy. In conjunction with job offset control (OFFSET parameter of the OUTPUT command), the user can exert extensive control over the offsetting function so as to build tailored, job-controlled stacks of output in the stacker tray of the LPS.

The following section describes the syntax of the command parameters and explanations of the parameter options.

PASSES

This parameter specifies a page offset for reports.

Syntax ROFFSET PASSES = *option(s)*

Options Table 7-18 lists the parameter option(s) and definition(s).

Table 7-18. **PASSES parameter option(s) and definition(s)**

Option(s)	Definition(s)
FIRST	Specifies that the satisfied criteria causes an offset only on the first pass of a collated print run.
ALL	Specifies that the satisfied criteria causes an offset on all passes of a collated print run. The default is ALL.

TEST

This parameter defines test expression for offsetting pages to the stacker tray.

Syntax ROFFSET TEST = *option(s)*

Options Table 7-19 lists the parameter option(s) and definition(s).

Table 7-19. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. If a test-exp is satisfied, the record causes an offset in the stacker tray. If a <i>test-exp</i> is satisfied in duplex mode, the logical page on which the record occurs is the first logical page of a new sheet.

There is no default.

Points to note

Note the following when using the ROFFSET command:

- The ROFFSET feature prints the record satisfying the test expression according to the normal job parameters.
- If ROFFSET is specified for an uncollated job, an offset occurs on only the first copy of the offset page.
- If a record satisfying the ROFFSET test criteria is not selected for or deleted from printing, the offset indication is maintained and used on the next record printed.
- If OFFSET=FIRST (or ALL) and the ROFFSET criteria is satisfied on the first page of a report, the normal offset from the preceding report is nullified.
- If segments are specified, the ROFFSET command is ignored.

Example

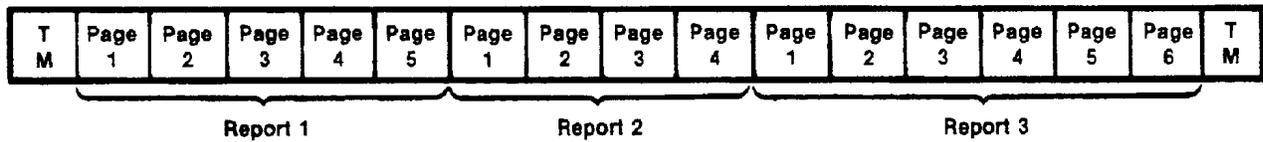
Following is an example of using the ROFFSET command.

A file has multiple reports without any delimiters separating the reports (as illustrated below). Each page of the report has a page number as part of the heading. Each report causes renumbering of the pages starting with Page...1. With the ROFFSET command coded below, an offset occurs for all passes of the reports.

```
T1: TABLE CONSTANT=('PAGE...1');
C1: CRITERIA CONSTANT=(105,8,EQ,T1);
    ROFFSET TEST=C1, PASSES=ALL
```

Figure 7-5 shows a sample ROFFSET command usage.

Figure 7-5. **Sample ROFFSET command usage**



RPAGE command

The RPAGE command allows for the repositioning of the current (or next) logical page. Refer to figures 7-5 and 7-6 for a pictorial description of the effect of the RPAGE parameters. Table 7-5 summarizes RPAGE command parameters.

Table 7-20 summarizes the RPAGE command parameters.

Table 7-20. **Summary of RPAGE command parameters**

Parameter	Specifies	Offline	Online	DJDE
SIDE	Side of physical page on which to reposition the logical page.	Y	Y	Y
TEST	Test expression for detecting an RPAGE record.	Y	Y	N
WHEN	Portion of the current logical page to be positioned.	Y	Y	N

The following sections describe the syntax of the command parameters and explanations of the parameter options.

SIDE

This parameter defines the side of the physical sheet on which the repositioned logical page is to appear.

Syntax RPAGE SIDE = *option(s)*

Options Table 7-21 lists the parameter option(s) and definition(s).

Table 7-21. **SIDE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>side-opt</i>	<p>Available side-opt parameters are: NUFRONT (the default), BACK, NUBACK, NEXT.</p> <p>In duplex (with no BFORM), side-opt produces the following results. If NUFRONT is coded, the specified logical page is positioned as the first logical page on a new sheet. If BACK is coded, the specified logical page is positioned as the first logical page of the next available back. If NUBACK is coded, the specified logical page is positioned as the first logical page on the back of a new sheet. If NEXT is coded, the specified logical page is positioned as the first logical page on the next available side, the back of the current sheet or the front of the next sheet.</p> <p>The logical page is moved unless it is already properly positioned as the first logical page on the specified side.</p> <p>In simplex, or in duplex with BFORM, the SIDE parameter defaults to NUFRONT and the specified page is positioned as the first logical page. If the logical page is already properly positioned, a blank sheet is not created.</p>
<i>offside-opt</i>	<p>Available offset-opt parameters are: NOFFSET (the default), OFFSET. If OFFSET is coded, the sheet on which the repositioned logical page occurs is offset in the output stacker. If NOFFSET is coded, no offset occurs.</p>

The default is (NUFRONT,NOFFSET).

TEST

This parameter defines the test expression for detection of an RPAGE record.

Syntax RPAGE TEST = *option(s)*

Options Table 7-22 lists the parameter option(s) and definition(s).

Table 7-22. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. The logical page on which this record normally prints is called the "current logical page."

WHEN

This parameter defines the portion of the current logical page to be positioned to the first logical page of a new location.

Syntax RPAGE WHEN = *option(s)*

Options Table 7-23 lists the parameter option(s) and definition(s).

Table 7-23. **WHEN parameter option(s) and definition(s)**

Option(s)	Definition(s)
BOTTOM	The logical page following the current logical page is repositioned.
NOW	The current record is repositioned. This record prints at TOF. Any spacing or skipping defined to occur before printing the current record is ignored.
TOP	The current logical page is repositioned. The default is TOP.

Points to note

Note the following when using the RPAGE command. These points apply to offline jobs only:

- If a record intended to satisfy the criteria for RPAGE is suspended by RSUSPEND, that record is not checked for the RPAGE criteria. Note that the RPAGE criteria takes effect when printing is resumed.
- Record selection/deletion is performed prior to RPAGE processing. If a record satisfying the RPAGE test criteria was previously not selected for or deleted from printing, the RPAGE does not function.
- Since DJDE processing occurs prior to RPAGE processing, DJDE records are not checked for satisfying the RPAGE criteria.

Figures 7-6 and 7-7 show the effects of RPAGE command SIDE and WHEN parameters, respectively.

Figure 7-6. Effects of RPAGE command SIDE parameter

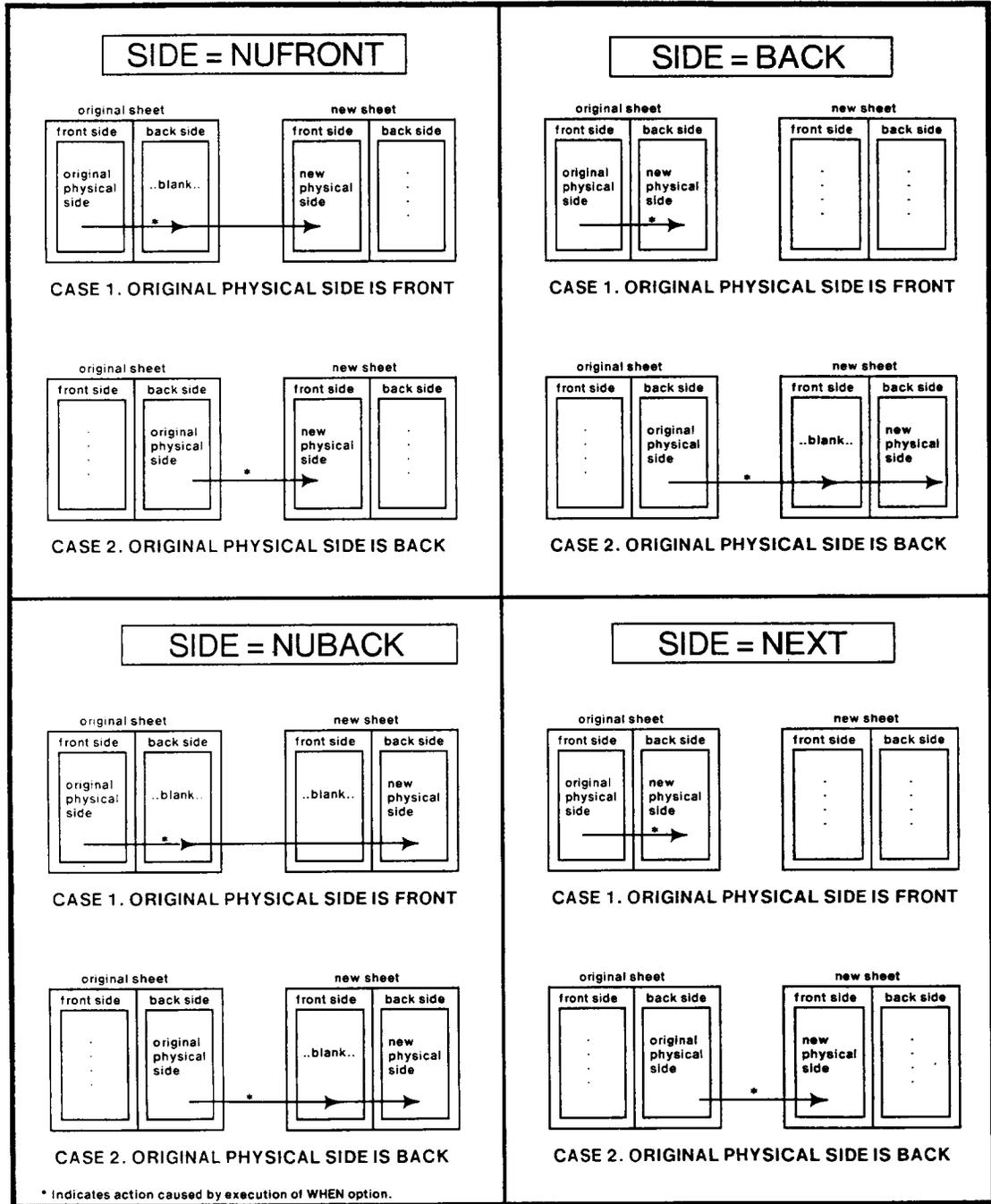
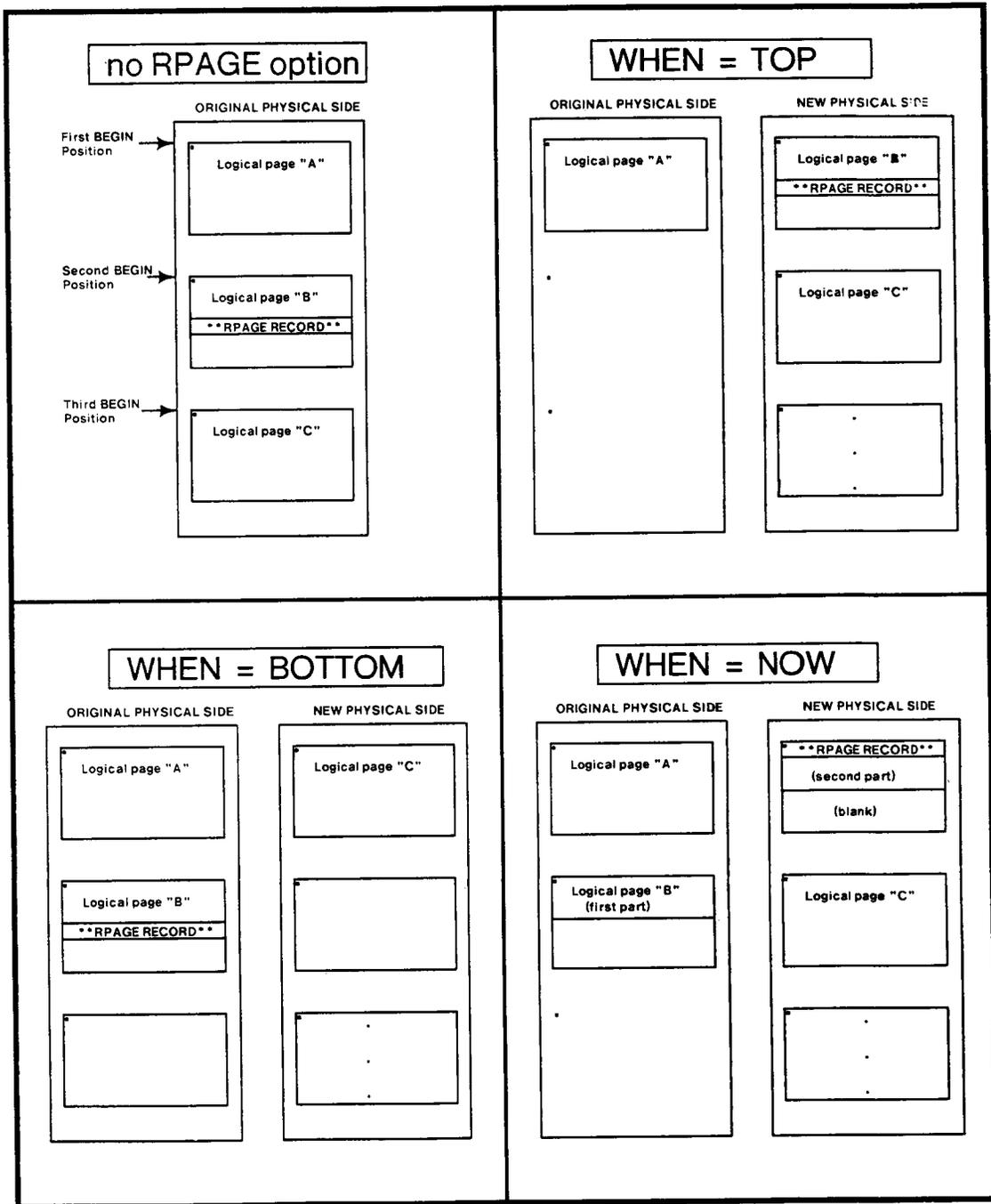


Figure 7-7. RPAGE command parameters



RSTACK command

The stacked reports feature enables the online and offline user to define a series of reports in a single file. This is accomplished by specifying an end-of-report condition in the coded logical processing commands RSTACK (online and offline) and BANNER (online use only). End-of-report is that point in processing a report when all of the pages of a copy of a report have been formatted to disk and processing has begun on the next report.

Reports are stacked in a file if more than one report is included in a single file and separated from each other logically but not physically (with tapemarks, operating system labels, and so forth). In processing stacked reports, the system checks each record for the logical end-of-report specification as defined by the TEST parameter of the RSTACK and BANNER commands. When RSTACK is used for report separation, end-of-report occurs as soon as the test criteria is satisfied. However, when BANNER is coded, the user specifies the number of consecutive banner pages which must satisfy the test criteria before end-of-report occurs.

Report separation based on banner page detection is performed with the BANNER command. This includes a test expression to detect a banner page and a count of the header or trailer pages used to separate reports. The syntax and format of the BANNER command are described in the "BANNER command" section.

RSTACK delimiter modes

There are two modes of stacked reports: delimiter and non-delimiter. In the delimiter mode, the record satisfying the TEST criteria is not part of the report following the delimiter but simply serves to separate or delimit one report from another. In the non-delimiter mode, reports are stacked one on top of each other without any special records separating the reports. The two modes are specified by the DELIMITER parameter of the RSTACK command.

If DELIMITER=YES is coded, the user may actually separate each report with multiple successive records, each of which satisfies the TEST expression on the RSTACK command. In this case, all consecutive delimiters are treated as a delimiter packet. In delimiter mode, the user has the parameter to print the delimiter (or the delimiter packet) and to select the output destination of this delimiter page—BIN, TRAY, or BOTH. The parameter BOTH delivers the page to the sample print tray and the output tray. The delimiter page, when printed, is output as part of the subsequent report. All of the printing parameters selected in the JDL remain in effect during report delimiter printing except carriage control, which is ignored; the carriage control is replaced by a print and space 1 line control. If DJDE records exist within the delimiter packet or immediately following, they are included on the delimiter page.

In non-delimiter mode (DELIMITER=NO), a single record separates one report from the next. After satisfying the TEST criteria, this record is considered part of the subsequent report. In this mode the delimiter record cannot be printed to the tray, that is, PRINT parameter not available when DELIMITER=NO. It also does not appear on the operator's display.

Delimiter display

Display of the delimiter (performed only when DELIMITER=YES) is dependent on the mode (Single/Multi) in which the job is run. Single or multiple report mode is a selectable parameter on the offline START command.

If the print job is run in single-report mode, the first delimiter of the next report is displayed. This occurs after an end-of-report is encountered. To begin the next single-report job, the operator must key in another START command. If the report mode is Multi or the system is online, the delimiter is not displayed.

Delimiter on accounting page

The user has the option to include part of the first record of a report on the accounting page. This command is selected by coding the ACCTINFO parameter, which is normally used to print part of the first delimiter.

Status display

The user has the parameter to display a subfield of the first record in a report in response to the JOBS command or STATUS function key. This command is selected by coding the HRPTNA or HJOBNO parameter.

Note that HJOBNO is not an RSTACK parameter.

Online RSTACK usage

When using RSTACK online, consider the following:

- Detection of RSTACK criteria within a not yet recognized banner page (RSTACK record occurs prior to BANNER criteria line) results in subsequent incorrect report separation.
- Detection of RSTACK criteria in a recognized but incomplete banner page (RSTACK record occurs after BANNER criteria line) is ignored.
- Detection of RSTACK immediately following report separation is ignored. This prevents null reports. If RSTACK is specified in a selected JDE or JDL, that is, a JDE or JDL invoked in a DJDE), the following restrictions to online systems apply:
 - If RSTACK is not specified in the JDE or JDL of the START command, the DELIMITER=NO parameter should be specified in the selected JDE or JDL. Specification of DELIMITER=YES is overridden.
 - If RSTACK is specified in the JDE or JDL of the START command, the DELIMITER parameter cannot be changed in a selected JDE or JDL.

- If DELIMITER=YES is specified, a record that satisfies the TEST criteria of a selected JDE or JDL results in an end of report normally. However, it prints at the start of the subsequent report unless it also satisfies the TEST (or DJDE) criteria of the starting JDE or JDL.
- If RSTACK is specified in the JDE or JDL of the START command but not in the selected JDE or JDL, RSTACK processing is suspended until end of report is triggered by BANNER page detection.
- If ACCTINFO is specified in the JDE or JDL of the START command, the ACCTINFO parameter cannot be changed or deleted in a selected JDE or JDL (even if RSTACK is suspended).
- If ACCTINFO is not specified in the JDE or JDL of the START command, it can be invoked in a selected JDE or JDL. The first record after the selected JDE takes effect is then printed as the ACCTINFO field.

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ACCTINFO

This parameter specifies that a subfield of the first record is being printed on the accounting page at the end of the report. If DELIMITER=YES is also coded, the subfield is from the first delimiter record of the report. For DELIMITER=NO the subfield is from the first data record.

Syntax RSTACK ACCTINFO = *option(s)*

Options Table 7-24 lists the parameter option(s) and definition(s).

Table 7-24. **ACCTINFO parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>offset, length</i>)	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>offset</i> Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record. • <i>length</i> The number of bytes in the subfield (1 to 64).

There is no default.

DELIMITER

This parameter specifies whether or not records that are separated from other records are part of the subsequent report.

Syntax RSTACK DELIMITER = *option(s)*

Options Table 7-25 lists the parameter option(s) and definition(s).

Table 7-25. **DELIMITER parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that all consecutive records satisfying the TEST criteria separate one report from another but are not part of either report.
NO	Specifies that this single record separates one report from another and is actually part of the subsequent report. The default is NO.

HRPTNA

This parameter specifies that a subfield of the first record is being displayed as REPORT NAME on the LPS console in response to the JOBS command or STATUS function key. If DELIMITER = YES is also coded, the subfield is from the first delimiter record of the report. For DELIMITER = NO, the subfield is from the first data record.

Syntax RSTACK HRPTNA = *option(s)*

Options Table 7-26 lists the parameter option(s) and definition(s).

Table 7-26. **HRPTNA parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(offset, length)</i>	This option has the following components: <ul style="list-style-type: none"> <i>offset</i> Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record. <i>length</i> The number of bytes in the subfield (1 to 16).
NONE	Specifies no report name is selected. The default is NONE.

PRINT

If DELIMITER=YES is coded, the user may specify if the report delimiters are to be printed, and if so, the output destination of the printed delimiters.

Syntax RSTACK PRINT = *option(s)*

Options Table 7-27 lists the parameter option(s) and definition(s).

Table 7-27. **PRINT parameter option(s) and definition(s)**

Option(s)	Definition(s)
BIN	Specifies that report delimiters are to be printed and the output delivered to the output stacker tray.
BOTH	Specifies that report delimiters are to be printed, and the output delivered to both the sample print tray and output stacker tray.
TRAY	Specifies that report delimiters are to be printed, and the output delivered to the sample print tray.
NONE	Specifies that report delimiters are not to be printed. For DELIMITER=NO, no page is printed. If the report is being printed in duplex mode, report delimiters are printed on a single-sided duplex output page. The parameter BOTH in duplex is treated as BIN. The default is NONE.

TEST

This parameter defines test expression for end-of-report conditions for either change mode or constant criteria.

Syntax RSTACK TEST = *option(s)*

Options Table 7-28 lists the parameter option(s) and definition(s).

Table 7-28. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	Defines test expression for end-of-report conditions for either change mode or constant criteria. The format and syntax of a test-exp are defined in the "Test expression definition" section. If a test-exp is satisfied, the record specifies an end-of-report condition. There is no default.

Points to note

Note the following when using the RSTACK command:

- If the TEST expression on the RSTACK command consists solely of a change mode CRITERIA command, DELIMITER=NO must be coded.
- An RSTACK command containing a TEST expression specifying a constant mode CRITERIA command and DELIMITER=NO can be used to detect a heading of a report as a report boundary.
- A record which is an RSTACK delimiter, that is, it satisfies the TEST expression, cannot be deleted from or not selected for printing by the RSELECT or RDELETE logical processing. If the record satisfies the RSTACK test criteria but is not a delimiter, it can be deleted from or not selected for printing but still causes report separation.

Example

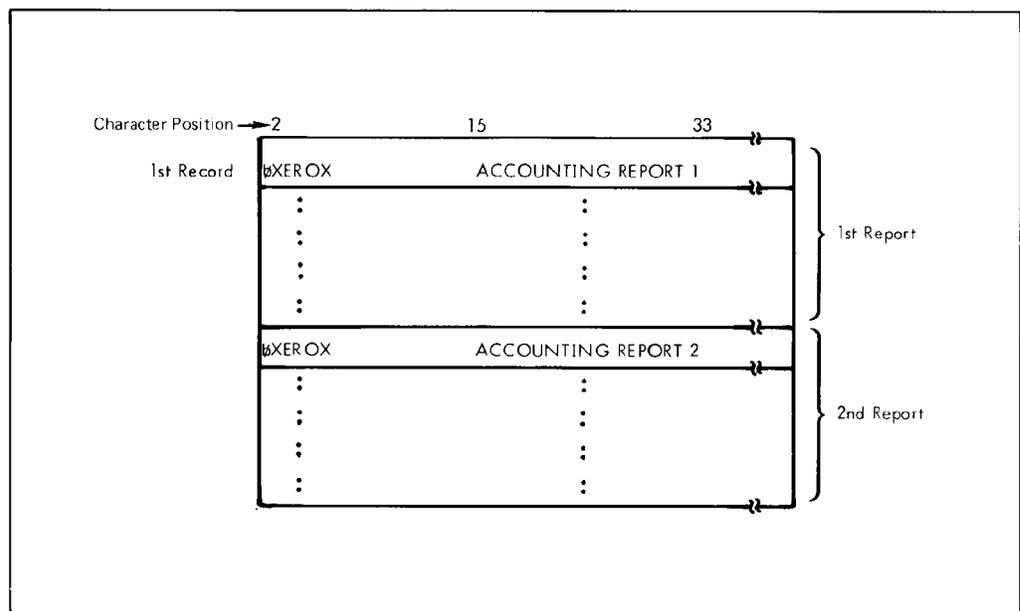
Following is an example of using the RSTACK command:

```
T1: TABLE CONSTANT=( 'XEROX' );
C1: CRITERIA CONSTANT=( 1, 5, EQ, T1 );
RSTACK
TEST=C1, DELIMITER=YES, ACCTINFO=( 14, 19 ), HRPTNA=( 20, 10 );
```

Reports are separated by a record with the characters XEROX in bytes 1 to 5 (relative to 0). Two reports are created as illustrated in the following figure. The delimiter record is not printed with the report.

Refer to figure 7-8 for a sample RSTACK command usage.

Figure 7-8. **Sample RSTACK command usage**



RSUSPEND and RRESUME commands

The print suppression logical processing function permits the user to delete from printing groups of records that are distinguishable at the start and end, but whose intermediate records may not be unique or distinguishable. (Print suppression or resumption is invoked by the use of two separate commands—RSUSPEND and RRESUME.) The tests for each command are independent and must be described separately. Each of the commands can specify the full range of tests as described previously for the other logical processing commands.

When specifying either the RSUSPEND or RRESUME command, the user can also specify whether suspension or resumption of printing occurs on the current or next record. This is controlled by the BEGIN parameter. This additional control provides the necessary flexibility to cope with the variability of requirements for print suppression. The BEGIN parameter is independently specifiable in both RSUSPEND and RRESUME commands.

Upon encountering a record which satisfies the test criteria specified on the RSUSPEND command, printing is suspended. If BEGIN=CURRENT is coded on the RSUSPEND command, this record is not printed. If BEGIN=NEXT is coded, the record satisfying the test criteria is printed, and records are discarded beginning with the following record.

Printing is resumed when a record satisfying the TEST in the RRESUME command is encountered. If BEGIN=CURRENT is coded in the RRESUME command, the record satisfying the TEST is printed. If BEGIN=NEXT is coded, printing resumes with the next record.

The following section describes the syntax of the command parameters and explanations of the parameter options.

BEGIN

This parameter specifies the BEGIN options.

Syntax RSUSPEND BEGIN = *option(s)*

Options Table 7-29 lists the parameter option(s) and definition(s).

Table 7-29. **BEGIN parameter option(s) and definition(s)**

Option(s)	Definition(s)
CURRENT	Specifies whether printing stops or resumes on the current or next record for offline jobs. For RSUSPEND, if CURRENT is coded, the record satisfying test-exp does not print. For RRESUME, if CURRENT is coded, the record is printed.
NEXT	If NEXT is coded, the record satisfying test-exp is not printed, and printing begins with the next record. If NEXT is coded, the record satisfying test-exp is printed, and printing is suppressed beginning with the next record. The default is NEXT.

TEST

This parameter defines test expressions for the beginning of print suppression (RSUSPEND).

Syntax RSUSPEND TEST = *option(s)*

Options Table 7-30 lists the parameter option(s) and definition(s).

Table 7-30. **TEST parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>test-exp</i>	The format and syntax of a test-exp are defined in the "Test expression definition" section. If the test-exp is satisfied, the record is used to suspend printing. Defines test expression for the resumption of printing (RRESUME) following print suppression. If the test-exp is satisfied, the record is used to resume printing following print suppression. There is no default.

Points to note

Note the following when using the RSUSPEND and RRESUME commands:

- **RSUSPEND**

- Make sure that if an RSUSPEND command is coded, an RRESUME command must also be present for the job. A warning is issued by the PDL compiler if one command, but not both, is invoked for a job. However, the JDE is compiled as programmed.
- The RSUSPEND command is intended to suspend records within a report.

- **RSTACK**

- A record satisfying the RSTACK test can still be found and terminates the report, even if the printing of records is suspended at the time. Also, DJDE records are not processed even if record printing has been suspended. Refer to chapter 8, "Specifying dynamic job descriptor entries (DJDEs)," for further information.
 - A record satisfying the RSTACK criteria is still detected and terminates the report and record suspension, even if the printing of records was suspended at the time.
 - If the criteria for RSTACK and RSUSPEND are satisfied on the same record, that record delimits the report, and the record is suspended.
- If a data record satisfying the test expression in the RSUSPEND is encountered, printing is suspended. If no record satisfying the test expression in the RRESUME command is encountered (or no RRESUME command is present for the job), there is no output generated for records that occur after the point of suspension.
 - Record selection or deletion is performed before suspend/resume processing. If a record satisfying either the suspend or resume test criteria was not selected for or deleted from printing previously, it does not cause either the suspend or resume function.
 - The records just before print suspension and after resumption should have compatible printer carriage control (PCC) characters. No additional carriage control characters are inserted by the system during the print suppression.
 - Use of the LINENUM parameter in the CRITERIA command is not recommended when using RRESUME. Since the carriage control characters are not processed during the print suppression, the line number used by the system is that which existed when the suppression started. This could result in the test criteria being either unexpectedly satisfied or never satisfied as a function of the line number where the print suppression started.
 - DJDE records are processed, not ignored, even when detected after an RSUSPEND record and before the corresponding RRESUME record.

TABLE command

The TABLE command is used to build a table of constants for use by the logical processing commands. Each constant included in a TABLE command is examined by the system to see if it is equal in value to the input data field specified in the CONSTANT parameter of the CRITERIA command. The TABLE command must precede its reference in a CRITERIA command.

The MASK parameter of the TABLE command permits character positions of a string to be ignored or tested for type attributes.

In an unmasked comparison (using only the CONSTANT parameter of the TABLE command) characters in the corresponding positions of the constant string and the input data string are tested for absolute equality one character at a time. In a masked comparison (using the MASK parameter), before the comparison is done for each character position, that position in the constant string is checked for a mask character as specified by the user. If one is found, the corresponding data character is tested only for the type indicated by the mask character.

Any valid character in the data character set, such as ASCII and EBCDIC, may be used in the MASK parameter as a mask character. Any character that is not explicitly tested for absolute equality by that TABLE command may be defined by the MASK parameter as a mask character. MASK parameter character definitions apply only to the TABLE command in which the definition occurs and, as in unmasked comparisons, it is possible to specify multiple strings in the CONSTANT parameter.

The following section describes the syntax of the command parameters and explanations of the parameter options.

CONSTANT

This parameter specifies the content of one or more string constant(s).

Syntax *ac:TABLE CONSTANT = option(s)*

Options Table 7-31 lists the parameter option(s) and definition(s).

Table 7-31. **CONSTANT parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>sc</i> ₁ [, <i>sc</i> ₂][,...])	<i>sc</i> is a string of constants composed of literal characters and masking characters (as defined by the MASK parameter of this command). All string constants must be of the same length in bytes. The number of bytes for all constants in the table (after conversion of any hexadecimal or octal constants) is limited to 255 bytes. There is no default.

MASK

This parameter defines mask characters used within string constants (specified by CONSTANT parameter) and indicates which character positions have special type testing and what that testing is.

Syntax *ac:TABLE MASK = option(s)*

Options Table 7-32 lists the parameter option(s) and definition(s).

Table 7-32. **MASK parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>(ignore-char [, charspec₁ [, charspec₂][, ...])</i></p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>ignore-char</i> Specifies a character to appear in those positions within a string constant which are not to be compared. • <i>charspec_i</i> (<i>i</i> = index number) specifies a character(s) to appear in those positions within a string constant where a test for type <i>i</i> characters is to be performed. A charspec is of the form: '<i>char</i>' or ('<i>char</i>', '<i>char</i>', ... '<i>char</i>'). Mask characters should be defined using the same string type as in the string specified in the CONSTANT parameter. <p>There is no default.</p>

Examples

This section shows examples of using the TABLE command.

Example 1

The following are sample TABLE commands. The identifiers T1, T2, and T3 are each referenced by a CRITERIA command.

```
T1: TABLE CONSTANT=('$$//');
T2: TABLE CONSTANT=('//JOB ','//EXEC');
T3: TABLE CONSTANT=(120)'*';
```

Example 2

The following example illustrates the use of the MASK parameter to determine whether a particular field in a data record contains one of several different 6-character serial number formats. Possible formats are: xx999x, x999x9, and 99x999. For this example, x represents any alphabetic character and 9 represents any numeric character.

If the input code type is EBCDIC, the VOLUME command includes TCODE=EBCDIC. The TABLE command to find the format x999x9 within the input data stream would look like:

```
T1: TABLE MASK=('?', '%', '@'), CONSTANT=('@ % % % @ %');
```

The characters used in the MASK parameter occupy numbered positions, beginning with 0, separated by commas. These position numbers in the MASK parameter are used as type numbers. Therefore, the mask character to type associations listed in table 7-33 are made for the above TABLE command.

Table 7-33. **Mask character to type associations for the TABLE command**

Character	Type	Meaning
?	None	Make no comparison
%	1	Standard default, any numeric (0-9)
@	2	Standard default, any alphabetic (A-Z, a-z)

The data string characters in the positions corresponding to those occupied by \ in the CONSTANT parameter are tested for type 2 (in the example, alphabetic A-Z or a-z). Characters corresponding positionally with a % are tested for type 1 (numeric, 0-9). If the string specified in the CONSTANT parameter included a ?, the corresponding data character from the input stream is considered equal without any comparison being made, because it is an ignore type mask character.

Example 3

If the TABLE command in example 2 were changed to:

```
T2: TABLE MASK=('?', '%', '@'), CONSTANT=('A7%%@%');
```

Then, when the first two character positions of the CONSTANT parameter are checked for a mask character (as specified in the MASK parameter), none is found. For those two positions, exact character matches between the input data string characters and the CONSTANT parameter characters (in this case A7) are required. Therefore, in this example, only data strings that begin with A7 can possibly pass the entire test.

Example 4

```
T2: TABLE MASK('*', ':', '!'), CONSTANT=('DATE :/:/:: ID!!!');
```

The TABLE command above sets up the character-to-type associations shown in table 7-34.

Table 7-34. **Character-to-type associations**

Mask position	Mask character	Character type	Meaning
0	*	None	Make no comparison
1	:	1	Any numeric (0-9)
2	!	2	Any alphabetic (A-Z, a-z)

A logical test using this TABLE command examines a field of an input data record to determine if the word DATE is present, if the / is where it is supposed to be, if the date itself is numeric (type 1), if the string ID is present, and if the first three characters after ID are alphabetic (type 2).

The basic element used to describe a test for a logical function is the CRITERIA. Each CRITERIA command describes a field in either a record or block and the specific test to be performed.

8. Specifying dynamic job descriptor entries (DJDEs)

Dynamic job descriptor entries (DJDEs) are parameters embedded within the input data stream and are used to modify the printing environment established by a job descriptor entry (JDE) “on the fly.” Dynamic job descriptor entry processing enables certain JDE parameters to be changed on a page-to-page or record-to-record basis. The IDEN command coded in a JDE notifies the system that DJDE records are included in the input data stream. This chapter contains information on these topics:

- Benefits of using DJDEs
- Page- and record-oriented DJDEs
- JDL specifications: the IDEN command
- DJDE format
- DJDE record specification
- Application of DJDEs
- DJDE operator information pages
- Job parameter modification restrictions
- Duplex DJDE page printing
- Effect on multiple logical pages
- Online DJDE restrictions
- DJDE processing optimization
- DJDE parameter definitions
- PDL parameters for graphics
- DJDE FILE parameters.

Table 8-1 lists the page and record oriented DJDEs. Table 8-2 contains a detailed description of each of the DJDE parameters.

Benefits of using DJDEs

Some of the benefits derived from changing these job parameters with DJDEs are as follows:

- The printing system does not stop between reports nor does it require operator intervention. The operator starts up a job on the printing system and typically returns to it only when minor operational activity is required.
- Forms may be changed on a page-to-page basis.
- Many variations on VFU channel, margin, and top- and bottom-of-form assignments may be applied to reports as they are created instead of being stored in the printing system through JDEs. Thus, the number of JDEs required for job processing is reduced.
- Unusual processing requirements may be satisfied through the use of DJDEs.
- Varying numbers of copies can be generated automatically with routing or distribution notification sent to the operator.
- Nonprint data can be downloaded for storage on the LPS system disks through the print stream.

In the following sections of this chapter, note that some DJDE parameters show a default value. This is the default value used if the parameter is not specified in either a DJDE or the current JDE or JDL. If the parameter is not specified in a DJDE but is specified in the current JDE or JDL, the JDE or JDL value remains in effect.

Page- and record-oriented DJDEs

There are two types of DJDEs: page-oriented and record-oriented.

Page-oriented

Page-oriented DJDEs change specific pages within a report and can change these pages differently in different copies. Such parameters may be placed within the report itself and take effect at the next page boundary. They may also appear at report boundaries to effect changes on all pages of a report (or report-ply).

Table 8-1 lists the page-oriented DJDEs.

Table 8-1. Page-oriented DJDEs

DJDE	Function
BEGIN	Specifies the location of the starting print line of a logical page for graphics.
BFORM	Specifies that a form be printed on the back side of a printed page.
COLLATE	Specifies whether the pages of a job are to be collated (placed into sets) or uncollated (like pages placed together).
COPIES	Specifies the number of copies or sets of the pages to produce.
DEPT	Specifies the accumulation of accounting statics for reports on a name basis.
DESTINATION	Specifies the output destination of printed pages.
DUPLEX	Specifies whether both sides of a piece of paper are printed (duplex) or only to top side (simplex).
FEED	Controls the stock (type of paper) on which the page is printed.
FONTINDEX	Specifies a particular location in the input record where an index to the specified font is stored.
FONTS	Specifies the fonts used in input data or variable (CME) data.
FORMAT	Specifies that a new page descriptor entry (PDE) is used for formatting control.
FORMS	Specifies the form to be merged on the printed pages.
INVERT	Specifies that an image is to be inverted.
ITEXT	Specifies a text message to be displayed to operators during the input process.
JDE	Specifies the JDE to be used within the selected JDL at the next page boundary.
JDL	Specifies the name of the JDL to be invoked at the next page boundary.
MAP	References a previously created font mapping file.
MARGIN	Specifies the left printing margin within each logical page.
MODIFY	Specifies the CME to be used on the page to override an existing CME.
NUMBER	Specifies page numbering control.
OTEXT	Provides a text message to be displayed to the operator during job printing.
PMODE	Specifies the printing mode for each page (either landscape or portrait).
RFORM	Specifies whether a form is printed on all RTEXT pages.

Table 8-1. **Page-oriented DJDEs** (continued)

DJDE	Function
RTEXT	Specifies text to be printed on a separate page preceding a report.
SEFMAP	Specifies a font mapping table different than the one specified in the JDL.
SEPARATORS	Specifies that the next data page in a report be used to generate segment separator sheets.
SHIFT	Specifies a shift of image on the page for binding purposes.
SF1FUNCTION	Specifies sheet finisher function number 1 through the DFA channel C6.
SF2FUNCTION	Specifies sheet finisher function number 2 through the DFA channel C7.
SIDE	Specifies the repositioning of the new logical page to the first logical page of the given side of a physical sheet of paper.
SNUMBER	Prints the segment number on generated segment separator sheets.
SPLIT	Defines the conditions for starting and ending segments.
SRECOVER	Specifies the recovery mode for sheets of paper in the paper path.
STIMING	Specifies timing parameters. Performs same function as the Finishing Configuration Utility (FCG) but is ignored by the system.
STOCKS	Redefines the parameters of a STOCKSET command (the types of stocks to use) for the page.
TMODE	Specifies process direction measurement for determining default pitch modes.
TRANS	Specifies whether transparencies are used in the print job.
XSHIFT	Allows shifting in the x-direction for short-edge binding.

Record-oriented

Record-oriented DJDEs take effect immediately at the next record following the last record of the set of DJDE records of which they are a part; that is, after an END parameter, a compiled record-oriented DJDE takes effect immediately.

The exception is the OVERPRINT DJDE, which takes effect at the next logical page boundary if FONTINDEX has been invoked in a JDE or DJDE. These DJDEs also may appear at report boundaries to change all pages in a report (or report-ply).

Table 8-2 lists the record-oriented DJDEs.

Table 8-2. **Record-oriented DJDEs**

DJDE	Function
ASSIGN	Assigns a VFU channel to a page line number or set of line number.
BOF	Specifies the bottom-of-form line number.
<i>C text</i>	Allows comment text in the DJDE record.
DATA	Specifies the location and length of printable data within an input record.
END	Specifies the end of DJDE information.
FILE	Enables files to load to the system disks while a print job is in progress.
OVERPRINT	Provides instructions to the system when overprint lines (print lines whose carriage control specifies printing with no line spacing since the last printed line).
TOF	Specifies the top-of-form line number.

Table 8-3 lists graphic DJDEs that are record-oriented.

Table 8-3. **Record-oriented DJDEs**

DJDE	Function
ALTER	Specifies the new imaging parameters for graphics.
BATCH	Permits normal processing on online banner pages for batch mode jobs.
CANCEL	Cancels automatic reimaging (hold) of IMAGE or GRAPHICS parameters.
GRAPHIC	Specifies that the DJDE is a graphic sentinel.
IMAGE	In batch mode, defines the new imaging parameters for the graphic.
SAVE	Specifies IMG files updated the current report which are not to be purged when the report's output processing is completed.

IDEN command

To invoke DJDE processing, an IDEN command must be coded in the JDL. This command notifies the system that DJDE records may be part of the input data stream. It also describes the search criteria for locating and identifying DJDE records. The DJDE record (or records) are interspersed among the data records in the input data stream. Each DJDE record contains an identification field (which matches the search criteria specified in the active JDE) and a series of parameters that describe the actual JDE changes to be applied to the report.

The IDEN command (coded in a user's JDE) notifies the system that a DJDE record (or records) may be part of the input data stream. The command also describes the characteristics of a DJDE record so the system can identify and locate any DJDE records. The IDEN command syntax and interpretation are described in table 8-3. An example of a coded IDEN command is:

```
IDEN PREFIX='RTEST', SKIP=9,OFFSET=3,OPRINFO=YES;
```

The PREFIX parameter defines the character string which must appear in the identification part of each DJDE record in the input data stream. When the system recognizes this character string (in the example, the EBCDIC string 'RTEST'), it examines the rest of the record for DJDE parameters. The OFFSET and SKIP parameters define the starting positions of the prefix string and DJDE parameters within the record.

Example:

```
IDEN PREFIX=E'DJDE', SKIP=7, OFFSET=2, OPRINFO=YES;
```

OFFSET

This parameter specifies the number of bytes (beginning at 0) from the beginning of the user portion of the record to the beginning of the prefix string constant of the DJDE record. It may be a negative number.

Syntax OFFSET = *option(s)*

Options Table 8-4 lists the parameter option(s) and definition(s).

Table 8-4. **OFFSET parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	A <i>value</i> specifies the number of bytes (beginning at 0) from the beginning of the user portion of the record to the beginning of the prefix string constant of the DJDE record. It may be a negative number. The default is 0.

OPRINFO

This parameter specifies whether the DJDE record is to be printed to the tray or not.

Syntax OPRINFO = *option(s)*

Options Table 8-5 lists the parameter option(s) and definition(s).

Table 8-5. **OPRINFO parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that the DJDE record is to be printed and delivered to the tray.
NO	Specifies that the DJDE record will not be printed. The default is NO.

All non-Interpress system-generated pages, such as accounting sheet, error summary, OPRINFO, recovery marker, and report separator are printed in the same xerographic mode as the previous page to prevent unnecessary xerographic mode switching (XMS). The exception to the rule is when the default ink is a primary color and the previous page's xerographic mode is BLACK. Accounting sheets always print using the system default 'BLACK' ink. The ink used to print all other non-Interpress system generated pages depends on the xerographic mode of the previous page and the default ink.

PREFIX

This parameter specifies an *sc* identifier, which is the search criterion for recognizing DJDE records. It is a byte string of up to 255 characters represented as a hexadecimal, octal, BCD, ASCII, or EBCDIC character constant (as described in the "Right-part constants" section of the "Overview" chapter).

Syntax PREFIX = *option(s)*

Options Table 8-6 lists the parameter option(s) and definition(s).

Table 8-6. **PREFIX parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>sc</i>	Any records within the data stream which contain an identification field equal to the specified prefix string <i>sc</i> are recognized and processed as DJDE records. There is no default.

SKIP

This parameter offsets to the starting column of the DJDE parameters.

Syntax SKIP = *option(s)*

Options Table 8-7 lists the parameter option(s) and definition(s).

Table 8-7. **SKIP parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies the number of bytes (beginning at 0) from the beginning of the user portion of the record to the beginning of the DJDE parameters. It may be a negative number.</p> <p>The default is 1.</p>

Points to note

Note the following when using the IDEN command.

When OPRINFO=YES is selected, the length of the DJDE record that prints on the sheet is whatever amount will fit on the width of the paper. With portrait orientation, using OPRINFO=YES may cause data or page exceeds page size errors if the DJDE record is too long.

DJDE record specification

DJDE records are created by the user as part of the data stream. The system looks for them only if a DJDE prefix has been specified by an IDEN command within the JDL used to process the job. DJDE information is contained in one or more records, each of which may be up to the maximum record length specified for the job input data. For each DJDE record, the prefix (the identification field) must begin in the same location in the record. DJDEs are terminated by an END parameter in the last DJDE record. All specified DJDE information is applied at the next record or page boundary after the END parameter is encountered. There may be multiple DJDE sequences in a job. Each set modifies only the specific parameters mentioned within the DJDE.

The following should be considered when preparing DJDE records as part of the input data stream. Examples of DJDE records are illustrated in figures 8-1 and 8-2.

- The DJDE record may contain more than one parameter. Each parameter within a record must be separated from the next parameter by a comma.
- Lowercase alphabetic characters are not accepted in the DJDE parameter line. If they are used, a DJDE syntax error message is displayed.
- The end of a DJDE record parameter that is split and continued on the next DJDE record should be terminated with a ,; (comma, semicolon).
- The prefix in the DJDE record may appear after the DJDE parameter(s) as long as it is consistently located in all DJDE records.
- DJDE records need not be consecutive since none of the parameters are applied until END; is encountered. However, it is recommended that DJDE records be consecutive, especially when used with delimiter records. To optimize input processing speed, DJDE parameters should be grouped into as few records as possible. A group of DJDE records terminated by an END parameter is called a packet.
- Only one page-oriented packet can be deferred to the next page. If a page-oriented DJDE packet is received after the start of page A so that it is deferred to page A+1, any separate DJDE packets following it on page A may cause unpredictable results and may possibly be printed as data.
- A page-oriented DJDE positioned prior to any print parameters on a page takes effect on the current page. A page-oriented DJDE positioned after the first print parameter on a page takes effect on the next page, unless end-of-report is encountered.
- For delimiter mode stacked report processing, DJDE parameters may be placed in the report body. They may also be placed within or after (but not before) the delimiter record packet.
- For change mode stacked reports, the DJDE parameters may be placed within the body of the stacked report, where the DJDE records contain the same change field contents as the report to which the DJDE applies.

- After a set of DJDEs is terminated by an END parameter, there must be one or more data records before the next DJDE packet. DJDE records directly following a previously terminated DJDE parameter set are ignored.
- If the file containing the DJDE is variable-blocked, the program that blocks the file may strip off trailing blanks. Thus, if the comment record contains no actual comments, the blank following the C may be eliminated. If the C is the last character of the record, the parameter record is accepted. However, any character other than a blank following the C causes the record to be processed as a legitimate DJDE record and not as a comment. Refer to the C text parameter in table 8-2.
- Print-and-skip carriage control parameters associated with online DJDE records are treated as skip immediate carriage control parameters. If the LPS is not at the channel specified, a skip occurs; but if it is already there, no skip occurs unless the last carriage control parameter received was a print without spacing parameter. Print-and-space carriage control parameters (as well as the print-without-spacing carriage control parameter) associated with DJDE records are ignored.
- The printer carriage control (PCC) byte associated with offline DJDE records is ignored.
- If a DJDE is always created because of coding procedures, but there is no need to set any of the DJDE parameters for a particular report, a null DJDE may be created with only an END parameter and no other parameters specified, which will not modify any of the existing job setup parameters.
- FORMS, FORMAT, and BFORM DJDE parameters can appear only once in the DJDE packet.

Application of DJDEs

The following occurrences should be noted when using DJDE records:

The specific parameters included in a DJDE packet are the only ones modified when the DJDE is applied. The only exceptions are the JDE=DJDE and JDL=DJDE, which cause all processing parameters contained in the JDE to be updated except for those listed in the "Job parameter modification restrictions" section. The modified parameter remains in effect until that parameter is encountered in another DJDE packet or until the end-of-report is reached. At the start of the next report, either of the following occurs:

- If in multireport mode or operating online, the next report begins with all of the parameters specified in the original JDE invoked by the operator's START command.
- If in single-report mode, the system stops, waits for the operator to enter a new START command and then processes the next report with all processing parameters defined in the JDE invoked by the new START command.

Figures 8-1 and 8-2 show the usage of a single- and multiple-record DJDE and IDEN command

Figure 8-1. Single record DJDE and IDEN command

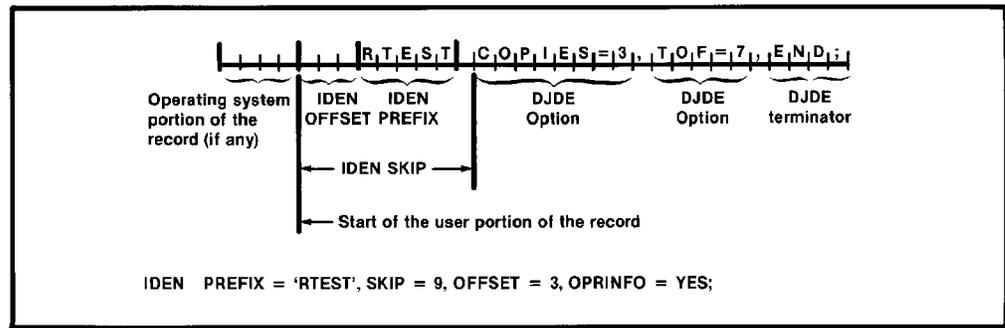


Figure 8-2. Single record DJDE and IDEN command

```

*MTEST C MULTI RECORD DJDE EXAMPLE;
*MTEST FORMS=(XEROX1,1,3),FORMAT=XPDE12, FONTINDEX=1,NUMBER=(3,15,55),;
*MTEST COPIES=20,COLLATE=YES,ASSIGN=(1,5),ASSIGN=(5,32),;
*MTEST FONTS=((P0612A),(P0812A)),;
*MTEST ASSIGN=(12,63),TOF=5,BOF=66,END;

  IDEN    PREFIX='MTEST',SKIP=7,OFFSET=1,OPRINFO=YES;

* = printer carriage control byte

```

Using the JDE=DJDE to reassign the top-of-page channel assignment (generally channel 1), when the DJDE is a data record, may incorrectly position the first page after the page transition occurs. Page-oriented DJDEs, like the JDE=DJDE, are implemented when the LPS detects the switch from one physical page to the next.

The normal page transition is caused by a skip from the bottom of a page to the top of the next, generally channel 1. The LPS knows a page transition has occurred only because the channel 1 assignment is on a line number less than the present line number of the current page, and therefore a page transition must have been created. Unfortunately, positioning to the old channel 1 line number has already occurred. The new JDE parameters are implemented after the page transition. There are several possible solutions to this problem.

- You may add an ASSIGN=DJDE to the DJDE packet to change the applicable channel, such as channel 1, to the new line number. Since the ASSIGN DJDE is a line-oriented DJDE, it is implemented on the next line and thus affects the next channel skip.
- You may insert a record after the DJDE packet to cause the page transition and then allow a second skip to channel 1 to position to the correct line. This causes a blank page to be printed if the new channel 1 is less than the old but will be totally transparent if the new channel 1 assignment is greater (further down the page) than the old.
- In the online environment only, the carriage control on the DJDE record itself is honored if it is a SKIP parameter. If the DJDE uses the ASSIGN parameter to specify a new channel 1, a skip-to-channel-1 carriage control on the DJDE record causes a skip to the new channel 1. But if the DJDE uses JDE= to specify a new channel 1, the skip-to-channel-1 carriage control on the DJDE record causes a skip to the old channel 1, until a physical page break occurs.

DJDE operator information pages

The changes to the JDE, specified in the DJDE, are incorporated when END parameter is encountered. The changes begin on the next record or page following the last DJDE record.

The specification OPRINFO=YES in the JDE ensures that the DJDE records are printed and sent to the tray at the next page transition after an END parameter. The DJDE records are printed on a separate page from the report data. However, no page formatting occurs and DJDEs on one long record are truncated at the end of the physical page boundary. They are printed and delivered to the tray regardless of the OPRINFO parameter if the DJDE contains an error. Comments in the DJDE may be used for operator notification or output routing instructions in conjunction with the OPRINFO parameter.

In duplex processing, DJDE records are printed as duplex pages with a blank page on the back (and are counted on the accounting sheet). If a DJDE is applied at the transition to the back side of a duplex page and OPRINFO=YES, the DJDEs are printed following the back side page. A row of asterisks (*****) appears on the DJDE page following the parameters whose printing is so delayed.

The message:

***MISSING END COMMAND OR MISSING PAGE BOUNDARY

is automatically printed on an OPRINFO page at the end of a report if:

- No END parameter is found in the DJDE.
- No complete page boundary was found before the end of the report, that is, at the end of report there were some DJDEs that had not yet been applied.

Job parameter modification restrictions

To process a DJDE, the system has already been required to process data up to the recognition and interpretation of the DJDE itself. This requires that the basic description of the input source must have been correct prior to the application of the DJDE. Thus, the description of the input source cannot be changed through DJDE, either by accident or by intent. The system does not allow these basic parameters to be changed through the DJDE parameter JDE (as defined in table 8-4). The JDE specified by JDE=DJDE is referred to as a selected JDE. The parameters which cannot be changed through a selected JDE are listed in table 8-8.

Table 8-8. **Parameters which cannot be changed through a JDE**

Command	Parameter(s)
VOLUME	HOST LABEL UNPACK
OUTPUT	OFFSET
IDEN	All parameters
BLOCK	All parameters
RECORD	All parameters
BANNER	All parameters

Changing the online BANNER offset criteria or VOLUME command OPTIMIZE parameter in a selected JDE causes unpredictable results and should not be attempted.

The selected JDE should specify all JDE parameters which stay the same as the original JDE as well as the changes. The parameters which can be changed through a selected JDE are listed in table 8-9.

Table 8-9. **Parameters which can be changed through a JDE**

Command	Parameter(s)
LINE	DATA FONTINDEX MARGIN OVERPRINT PCCTYPE VFU
LINE	All parameters except OFFSET
VOLUME	TCODE CODE
ACCT	All parameters
EXPORT	All parameters
MESSAGE	All parameters
PDE	All parameters
ROUTE	All parameters
ROFFSET	All parameters
RAUX	All parameters
RPAGE	All parameters
RSTACK	TEST (online only)

If a front cover is invoked in a selected JDE which takes effect after the first page of the report, a cover is pulled from the AUX tray at the point where the DJDE is applied.



Note: CODE changes to data may occur and will take effect on the page boundary following the DJDE record. If a translation code change is involved in a selected JDE, the parameter portion of any subsequent DJDE must be in the new code. The prefix used by the IDEN must, however, retain the original hexadecimal value. Thus, if the CODE changes from EBCDIC to ASCII, the prefix must remain in EBCDIC even though the parameter portion changes to ASCII.

Duplex DJDE page printing

When certain DJDEs (listed below) are invoked between an odd-numbered page (front side) and an even-numbered page (back side), the following printing sequence occurs. The odd-numbered page is printed as usual, but with a blank back side. What would have been printed on the back side (the even-numbered page) is printed on the front of the next sheet. This occurs for the following DJDEs:

- COPIES
- BFORM
- OTEXT
- RTEXT
- JDE
- JDL
- DUPLEX=NO
- FEED

Effect of multiple logical pages

If multiple BEGINs are being used to define multiple logical pages on one physical page (as described in the "Multiple logical pages on physical page" section in the chapter "Print format commands") the following points should be considered:

- With the exception of DEPT and SHIFT, all page-oriented DJDEs are applied at a logical page boundary.
- The following DJDEs, when applied, cause the current page position to be set to the first logical page of the front side of a new physical sheet:
 - DUPLEX
 - COPIES
 - JDL
 - JDE
 - BFORM
 - OTEXT
 - RTEXT
 - FEED
- The following DJDEs, when applied, cause the current page position to be set to the first logical page of the next physical side:
 - FORMS
 - FORMAT
 - BEGIN
 - PMODE
 - FONTS

COPIES=processing

If the system is approaching disk saturation, the output task may have already released disk space for a report that increases its copy count from an initial count of one to a higher number. To avoid this, do one of the following:

- Set the number of copies of the first page to 2 or higher.
- Include a copy-sensitive CME.
- Use the COPIES parameter in the START command.

Online DJDE restrictions

Online DJDE restrictions If an online job has header pages only (no trailers), DJDEs in the first header page of a report are recognized but not applied. Also, if headers only are defined, DJDEs invoked within the current report are applied to the first header page of the following report. These conditions result from the fact that the end-of-report processing for the first report is not detected until the first header page of the following report is received and processed.

DJDE processing optimization

A slight optimization in the processing of the FORMS, FORMAT, MODIFY, ALTER, CANCEL, GRAPHIC, and IMAGE DJDEs may be realized if the following conditions are met:

- The FORMS, FORMAT, and MODIFY DJDE parameters are contained within a single record with no other parameters.
- The ALTER, CANCEL, GRAPHIC, and IMAGE DJDE parameters each are contained within a single record with no other parameters.
- The first DJDE starts at the SKIP position, and there are no blanks until the END parameter.
- Each DJDE parameter is followed by a comma, and the last comma is followed by the END parameter.
- The FORMS, FORMAT, and MODIFY DJDE parameters are limited to the following syntax:

```
FORMS=form-id  
FORMAT=pde-id  
MODIFY=cme-id
```

DJDE parameter definitions

The following sections describe the syntax of the command parameters and explanations of the parameter options.

ALTER

This parameter specifies new imaging parameters for graphics previously referenced using the hold (automatic reimaging) parameter.

Syntax ALTER = *option(s)*

Options Table 8-10 lists the parameter option(s) and definition(s).

Table 8-10. ALTER parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>name</i>	Identifies a previously called out graphic (refer to DJDE IMAGE parameter) which is currently subject to the hold parameter of its calling command. ALTER maintains the hold parameter in effect but also specifies a new set of imaging parameters for the current and subsequent pages.
(<i>vpos, hpos</i>)	This option has the following components: <ul style="list-style-type: none"> <i>vpos</i> Specifies the vertical position of the top edge of the graphic, relative to 0,0 on the current physical page in centimeters (cm), inches (in), (specified as a decimal number with up to three digits to the right of the decimal point), dots, xdots, or user-defined units (un). If un is specified, the user-defined positioning unit must previously be defined by the UNITS parameter of the OUTPUT command of the current JDE or JDL (refer to the chapter "Print format commands"). If no units are specified, inches are assumed. <i>hpos</i> Specifies the horizontal position of the left edge of the graphic relative to 0,0 on the current physical page. The form of the specification (un, cm, in, dots, or xdots) is the same as for <i>vpos</i>.
(<i>n/d</i>)	Specifies the reference scale factor. Each parameter, <i>n</i> and <i>d</i> , must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied.

Examples ALTER=(BUS , 5 , 5 , 1) ;
ALTER=(IMG 1 , 3 , 5) , END ;

ASSIGN

This parameter specifies an assignment of a VFU channel number to a page line number or set of line numbers. It is possible to have multiple ASSIGN parameters within a DJDE. They take effect at the next record following a DJDE END parameter. DJDE assignments affect only those channel assignments specified. Other assignments remain the same. The ASSIGN parameters are the same as for the VFU command ASSIGN parameter described in the chapter "Print format commands."

Syntax ASSIGN = *option(s)*

Options Table 8-11 lists the parameter option(s) and definition(s).

Table 8-11. **ASSIGN parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>channo</i> , <i>lineno</i> [, <i>lineno</i>] [,...])	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>channo</i> The number of the channel being assigned. It is an integer in the range 0 to 15. The user may end the VFU command with a semicolon and start another VFU command without an id field to continue specification for the same channel or a different channel. • <i>lineno</i> The number of the output print line being assigned to a particular channel. It is an integer in the range TOF to BOF. There are no default assignments for any channel, including channels 1, 9, and 12.

BATCH

This parameter is supported for online batch mode jobs only. It permits normal processing of online banner pages for batch mode jobs by delimiting the batch mode graphic data. It must be used precisely to mark the beginning and ending of online batch mode data. The first graphic (or IMAGE DJDE) must be preceded immediately by a BATCH=START DJDE, and the last graphic must be followed immediately by a BATCH=END DJDE. This parameter, when specified, must be the only parameter in the DJDE packet.

Syntax BATCH = *option(s)*

Options Table 8-12 lists the parameter option(s) and definition(s).

Table 8-12. **BATCH parameter option(s) and definition(s)**

Option(s)	Definition(s)
START	Marks the beginning of online batch mode data.
END	Marks the end of online batch mode data.

BEGIN

This parameter specifies the location of the starting print line of a logical page for graphics. The *vpos* parameter specifies the vertical position of the first character of the first print line on the logical page. It may be specified in inches (IN) or centimeters (CM). The default is inches.

Syntax BEGIN = *option(s)*

Options Table 8-13 lists the parameter option(s) and definition(s).

Table 8-13. **BEGIN parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>vpos</i>	Specifies a decimal number with up to three digits to the right of the decimal point, for example, 0.563IN and 2.35CM are all legal specifications.
<i>hpos</i>	Specifies the horizontal position of the first character of the first print line on the logical page (in IN or CM as for <i>vpos</i>). All specifications are rounded to the nearest dot (1/300 of an inch) for positioning of the logical page.

In specifying the location of the beginning of a print line on the logical page, measurement is performed by viewing the page in the mode (landscape or portrait) in which it is to be printed. There may be more than one logical page per physical page; these are defined by the use of multiple BEGIN parameters. If no BEGIN parameter is specified, then the default is FMT1.

BFORM

This parameter specifies that a form be printed on the back side of a user page (must have DUPLEX=YES). It takes effect as soon as the data is read into the system. BFORM parameters are the same as for the OUTPUT command.

Syntax BFORM = *option(s)*

Options Table 8-14 lists the parameter option(s) and definition(s).

Table 8-14. **BFORM parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(form-id,[init],[copies])</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>form-id</i> Specifies a file name which exists on disk. This file is created by compiling an FSL source file with the FDL system task. form-id may be the keyword NONE, which means that no form is to be added to the associated report page of variable data. • <i>init</i> Specifies the beginning ply number to which a specified form applies. This defaults to the first or next copy. • <i>copies</i> Specifies the number of plies (passes) to which a specified form applies.
NONE	Specifies that no form is to be added to the associated report page of variable data.

BOF

This parameter specifies the bottom-of-form line number. It takes effect at the next record following a DJDE END parameter. A value is defined the same as for VFU BOF in the chapter "Print format commands."

Syntax BOF = *option(s)*

Options Table 8-15 lists the parameter option(s) and definition(s).

Table 8-15. **BOF parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	Specifies the number of lines from the top of the output page to the last print line on the page (bottom of form).

Considerations If bottom-of-form (BOF) is altered by DJDE to a line number smaller than the current line from which the DJDE was processed, input effects an immediate page transition. If carriage control action to perform at BOF is specified as OVR (the default), input spaces from TOF a number of lines equal to the difference between its location at DJDE and the new BOF. Refer to the PCC ASSIGN command in chapter 5 regarding alternate BOF parameters.

Parameter rules

Note the following rules:

- All parameters have the form
keyword = value
- *values* must be specified. A *value* may be a keyword described as listed in table 8-16.

Table 8-16. **Value(s) and definition(s)**

value(s)	Definition(s)
<i>xxx</i>	Alphanumeric text, including spaces. Case sensitivity is retained.
<i>nnn</i>	Numeric text, including one optional decimal point. Maximum of two digits after the decimal point.
<i>sss</i>	Numeric text, no decimal points.
<i>yy-mm-dd</i>	Numeric text describing year, month, and d day. Each pair of characters is separated by a dash. Eight characters in all are required.

- Parameters are separated by a comma.
- *values* cannot have commas embedded.
- White space within a parameter is ignored if it immediately precedes or follows the "=" sign.

- White space following a value is ignored.
- White space preceding or following a comma is ignored.
- A comma following the last parameter is ignored.
- Parameter keyword text is case insensitive.

C

This parameter allows for comment text to be included in the DJDE record. The C must be followed by a space (not by an equal sign), and must begin at the skip position specified in the IDEN command. Any text up to a semicolon, comma (unless semicolon or comma is enclosed in parentheses), or end-of-record is treated as commentary.

Syntax C = *option(s)*

Options Table 8-17 lists the parameter option(s) and definition(s).

Table 8-17. **C parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>text</i>	Specifies the comment text.

CANCEL

Cancels, effective on the current page, the hold (automatic reimaging) specified by the H parameter of an IMAGE or GRAPHIC parameter.

Syntax CANCEL = *option(s)*

Options Table 8-18 lists the parameter option(s) and definition(s).

Table 8-18. **CANCEL parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(name₁</i> <i>[,name₂][,...])</i>	<i>name</i> identifies the graphic(s) being held.
ALL	Specifies that all graphics being held are to be cancelled.

COLLATE

This parameter specifies whether collated or uncollated mode is to be used. It takes effect at the next page boundary but must occur prior to the first data record of the report. In duplex processing COLLATE=NO is ignored.

Syntax COLLATE = *option(s)*

Options Table 8-19 lists the parameter option(s) and definition(s).

Table 8-19. **COLLATE parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that report copies are to be printed in collated mode.
NO	Specifies that report copies are to be printed in uncollated mode.

COPIES

This parameter specifies the number of copies of a report to produce at the next page boundary. In duplex processing, if COPIES appears before the back side of a duplex page, a blank back sheet is output, and the print data continues on the next front page. If number is set to zero, this copy count remains in effect until a new copy count or end of report is encountered. DJDEs encountered while number is zero is in effect are processed normally.

The number of copies specified in the DJDE overrides the number of copies in the OUTPUT command. You may encounter problems when the copy count at the start of the report is 1 (either by virtue of the JDE or the last DJDE encountered at the start of report), and the copy count is later increased. There is no problem if the COPIES=DJDE is in the delimiter at the start of report and the value of the COPIES=DJDE is at least 2. The problem is caused when output processing releases space as soon as it believes that it no longer needs it.

The following procedures are suggested to avoid the problem:

- Set the number of copies of the first page to 2 or higher.
- Include a copy-sensitive CME.
- The COPIES parameter in the START command can be used. COPIES parameter in the START command overrides the DJDE parameter, COPIES=; in this case, the DJDE COPIES= parameter is not applied. Therefore, it is possible to run the same job with the COPIES parameter in the START command and one without the COPIES parameter in the START command and get two different outputs.

Syntax COPIES = *option(s)*

Options Table 8-20 lists the parameter option(s) and definition(s).

Table 8-20. **COPIES parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>number</i>	Specifies the number of copies of a report to produce at the next page boundary.

DATA

This parameter specifies the location and length of printable data within an input record. It takes effect at the next record following a DJDE END parameter.

Syntax DATA = *option(s)*

Options Table 8-21 lists the parameter option(s) and definition(s).

Table 8-21. **DATA parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>pdo</i> , <i>length</i>)	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>pdo</i> (Print data offset) is the number of bytes between the start of the user portion of the logical record and the first character of the record to be printed. • <i>length</i> Specifies the maximum length of printable data within each logical record. For offline systems the default is (1,132). For online systems the default is (0,150). For HOST=RSX11(FLX), the DATA parameter must be included in the user's JSL since the default pdo value causes the first data byte to be skipped. <p>The default is (1,132) offline. The default is (0,150) online.</p>

DESTINATION

This parameter determines the delivery destination of paper.

Syntax DESTINATION = *option(s)*

Options Table 8-22 lists the parameter option(s) and definition(s).

Table 8-22. **DESTINATION parameter option(s) and definition(s)**

Option(s)	Definition(s)
BIN	Specifies delivery to the currently active bin.
TRAY	Specifies delivery to the sample tray. Transparencies should only be sent to the sample tray. Use caution when selecting the sample tray because of its limited capacity.
EXPORT	Specifies that sheets be sent to the bypass transport. Export is honored only if SELECT=AUTO is specified by the operator.

DUPLEX

This parameter specifies whether printing is to occur on both sides of a page (duplex mode) or a single side (simplex mode). It takes effect on the next page boundary.

Syntax DUPLEX = *option(s)*

Options Table 8-23 lists the parameter option(s) and definition(s).

Table 8-23. **DUPLEX parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Selects duplex printing.
NO	Selects simplex printing.

END

This parameter specifies the end of DJDE information. When an END parameter is encountered, the system applies all DJDE information specified to the current printing environment at the next page or record boundary. Note that after an END parameter, a DJDE is compiled and record-oriented DJDEs take effect immediately.

Syntax END

FEED

This parameter controls the stock on which the page is printed. FEED=stock-reference references the stock assigned to a stock-name by the STOCKSET command in effect at the time the page is printed.

Syntax FEED = *option(s)*

Options Table 8-24 lists the parameter option(s) and definition(s).

Table 8-24. **FEED parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stock-reference</i>	Allows users to change stocks associated with a job without altering the stock references in the data application itself.
<i>stock-name</i>	Bypasses the reference feature, but still requires that the stock-name is specified as present in the current STOCKSET. (FEED=cluster- reference should always be used.)
(MAIN,AUX ,OPR)	Provide compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN. Note that the terms <i>cluster-name</i> and <i>stock-name</i> are synonymous. The default is OPR.

FILE

This parameter provides a capability of loading files to the LPS disks while a print job is in progress. The print job may be invoked solely for the purpose of downloading permanent files. Although intended primarily for use with online systems, this parameter is not restricted to online use.

Syntax FILE = *option(s)*

Options Table 8-25 lists the parameter option(s) and definition(s).

Table 8-25. **FILE parameter option(s) and definition(s)**

Option(s)	Definition(s)
([<i>file-name</i>] [, <i>file-type</i>] [, <i>f</i>] [, <i>s</i>] [, <i>n</i>])	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>file-name</i> This is a 1 to 6 character name used to identify the disk file. • <i>file-type</i> This is a 3 character file type symbol. • <i>f</i> This is one of the following file input format parameters: <ul style="list-style-type: none"> — C = card-image format — L = LPS-labeled tape format (default) • <i>s</i> This is one of the following file storage parameters: <ul style="list-style-type: none"> — D = delete after report printed — P = permanent (default) • <i>n</i> This is the maximum number of card-images (default=120).

For LPS-labeled files, the DJDE FILE parameter may be specified as follows:

FILE=()

In this case, the file name and file type are obtained from the label record that precedes the file data.

Considerations

The input task handles the creation and deletion of files which are transmitted from a host. The output task marks files for deletion at the end of the report which created them. The input task checks for files to be deleted at the start of the job, at report setup, and before exiting. Note that when the output task marks the files for deletion, the input task has long since finished with the report that created the files and is either waiting for data or actively processing another report. If the input task is processing a report, the files are deleted after end-of-report processing for the current report. After the files for the previous report have been deleted, the input task then completes report setup for the report that follows.

(Refer to the "DJDE FILE command" later in this chapter.)

FONTINDEX

This parameter allows the user to specify a certain location in the input record where an index to the font to be used is stored. It takes effect on the next logical page boundary. The FONTINDEX parameters are the same as in the LINE command in the chapter "Print format commands."

Syntax FONTINDEX = *option(s)*

Options Table 8-26 lists the parameter option(s) and definition(s).

Table 8-26. **FONTINDEX parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(offset [,init-val] [,bit-opt])</i>	This option has the following components: <ul style="list-style-type: none"> • offset • init-val • bit-opt
NONE	

FONTS

This parameter specifies the fonts to be used in printing variable input data or CME data. The FONTS parameter within a DJDE takes effect on the next page boundary.

Syntax FONTS = *option(s)*

Options Table 8-27 lists the parameter option(s) and definition(s).

Table 8-27. **FONTS parameter option(s) and definition(s)**

Option(s)	Definition(s)
$((f_1, s_1 [LPI XDOTS DOTS]) [, (f_2, s_2 [LPI XDOTS DOTS])] [, \dots])$	<p>This option has the following components:</p> <ul style="list-style-type: none"> • f_1 Each font index (f_1) specifies a 1 to 6 alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability and up to 94 fonts on a single page). If the number of fonts to be invoked exceeds the size of one DJDE record, multiple FONTS parameters must be used. • s_1 Each spacing index (s_1) value specifies an optional override line spacing value specifying either lines per inch or dots per line to be associated with the font; maximum value for s_1 is 30 LPI; minimum value is 10 XDOTS or dots. If a unit (LPI, XDOTS, or DOTS) is not specified, LPI is used as the default.

If an override line spacing value is specified, lines printed using the font cause the indicated line spacing to be performed after the line using the font. If different fonts are used on the same print line, the line spacing value specified for the font of the largest character in the line is used to determine the position of the next print line. (Refer to the "Points to note" section.)

Points to note

Note the following when using the FONTINDEX and FONTS commands:

- An xdot is 1/600 unit of measurement. A form specifying xdots, however, can be created, edited, and compiled on any LPS with version 3 software, but the form, which is 600 spots per inch (spi), will not print on your 4850, 4890, 4050, 4090, 4635, or DocuPrint 96/DocuPrint 96MX which accepts only 300 spi jobs.
- Since there is ambiguity between the FONTINDEX and FONTS parameters, if either is abbreviated to the first three letters, the parameter defaults to FONTINDEX.
- In DJDE processing, if the input task (DJD) encounters more than 127 fonts in a packet, the remaining fonts in the packet are ignored or truncated. Because the DJDE packet in this case is truncated, the outcome of the DJDE is unpredictable. Thus, the DJD task sends the error message:

OS6700 SYNTAX ERROR IN DJDE.

FORMAT

This parameter specifies an entirely new page descriptor entry (PDE) to be used for formatting control. It takes effect on the next page boundary.

Syntax FORMAT = *option(s)*

Options Table 8-28 lists the parameter option(s) and definition(s).

Table 8-28. **FORMAT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>pde-id</i>	Refers to a separately cataloged file in the PDE library on disk. Parameters of the selected PDE, such as BEGIN, FONTS, and PMODE, may also be modified on an individual basis. Further information is contained in the "Page descriptors—PDE command" section in the chapter "Print format commands."

FORMS

This parameter specifies the form to be merged on the printed pages. It takes effect on the next page boundary. The form specified is invoked beginning with the first page of the specified starting copy number. Multiple FORMS parameters may be used to associate different forms with different copy plies.

Syntax FORMS = *option(s)*

Options Table 8-29 lists the parameter option(s) and definition(s).

Table 8-29. **FORMS parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>form-id</i> <i>form-id</i> [, <i>init</i> [, <i>copies</i>]])	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>form-id</i> Specifies a file name which exists on disk. This file is created by compiling an FSL source file with the FDL system task. <i>form-id</i> may be the keyword NONE, which means that no form is to be added to the associated report page of variable data. • <i>init</i> Specifies the beginning ply number to which a specified form applies. This defaults to the first or next copy. • <i>copies</i> Specifies the number of plies (passes) to which a specified form applies.
NONE	

Example

```
FORMS=(FORM3, 3, 2), END;
```

Points to note

Note the following when using the FORMAT and FORMS commands:

- Since there is ambiguity between the FORMAT and FORMS parameters, if either is abbreviated to the first three letters, the parameter defaults to FORMAT.
- Paper size can only be changed in a job stream by using a JDE to call out the desired paper size, that is, through the PAPERSIZE parameter of the OUTPUT command. It is important to change the image size when the paper size is smaller than the current image so that all data is printed on the page.

GRAPHICS

This parameter specifies that the DJDE is a graphic sentinel. That is, it immediately precedes and identifies a graphic in the report data stream. This parameter, when specified, must be the only parameter in a DJDE packet.

If no textual data has been previously encountered, the graphic is document interleaved and copied to an IMG type disk file.

If textual data has been previously encountered, the graphic is page interleaved (block mode) and the last previous text record should have caused a page transition. If not, the occurrence of this DJDE parameter forces an immediate page termination, and the next text record encountered is processed relative to TOF on the next physical page.

For online processing, an interleaved graphic which immediately follows a header banner page is treated as document interleaved. The header banner page must be defined by a BANNER command. Skip carriage control on a graphics sentinel DJDE is ignored for online processing.

Syntax GRAPHICS = *option(s)*

Options Table 8-30 lists the parameter option(s) and definition(s).

Table 8-30. **GRAPHICS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p>(<i>name</i>,<i>vpos</i>{UN CM IN DOTS XDOTS} ,<i>hpos</i>{UN CM IN DOTS XDOTS} ,[<i>H</i>],[<i>n</i>/<i>d</i>])</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>name</i> Identifies the graphic. If document interleaved, it becomes the file name with which the IMG file is written. If page interleaved, it is used to associate the graphic with other DJDE references. (If no other parameters are specified, the parentheses may be omitted.) <p>None of the remaining parameters may be used if the graphic is document interleaved nor should it normally be used if the graphic is referenced by an IMAGE DJDE. If page interleaved, the remaining parameters may be used exactly as on an IMAGE DJDE (except for the T parameter). If these parameters are specified in the GRAPHIC DJDE and also on one or more IMAGE DJDEs which reference the same graphic, the graphic is imaged on the page once for each reference.</p> <ul style="list-style-type: none"> • <i>vpos</i> Specifies the vertical position in units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS. An XDOT is 1/600 unit of measurement, which is used for 600 spots per inch (spi) processing. Jobs with XDOTS can be created, edited, and compiled on any LPS with version 3 software, but will not print on your 4050, 4090, 4635, DocuPrint 96/DocuPrint 96MX, or 4850 which is 300 spi. • <i>hpos</i> Specifies the horizontal position in units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS. • <i>H</i> Specifies that the same graphic is to be held and imaged without further callouts at the same position and with the same scale factor at all subsequent pages until changed by a DJDE CANCEL or ALTER parameter. • <i>n/d</i> Specifies the reference scale factor. Each parameter, <i>n</i> and <i>d</i>, must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied.

IMAGE

In batch mode, this OVERPRINT = parameter defines new imaging parameters for subsequent graphics. In all other modes, it defines imaging parameters for the named graphic.

Syntax IMAGE = *option(s)*

Options Table 8-31 lists the parameter option(s) and definition(s).

Table 8-31. **IMAGE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>(img-name,vpos{UN CM IN DOTS},hpos{UN CM IN DOTS},[,T][,H][,n/d])</i></p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>img-name</i> The name of an IMG file or the name on the DJDE sentinel record, such as GRAPHIC=name) of a page interleaved graphic immediately following the current page. The name parameter may be omitted in batch mode; however, the comma before vpos must be coded, since it is a positional parameter. • <i>vpos</i> Specifies the vertical position of the top edge of the graphic as an offset relative to 0,0 on the current physical page (the position that would be specified by a PDE BEGIN of 0,0). This parameter is specified as a decimal number with up to three digits to the right of the decimal point and may be in DOTS, centimeter (CM), inches (IN), or user-defined units (UN). If UN is specified, the user-defined positioning unit must previously be defined by the UNITS parameter of the OUTPUT command of the current JDE or JDL (refer to the chapter "Using logical processing"). If no units are specified, inches are assumed. • <i>hpos</i> Specifies the horizontal position of the left edge of the graphic as an offset, relative to 0,0 on the current physical page. The form of the specification is the same as for <i>vpos</i>. • <i>T</i> Specifies that the referenced graphic is page interleaved and follows the current page data in the input stream. • <i>H</i> Specifies that the same graphic is to be held and imaged without further callouts, at the same position and with the same scale factor on all subsequent pages until changed by a DJDE CANCEL or ALTER. • <i>n/d</i> Specifies the reference scale factor. Refer to the "Scaling" section in your <i>Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference</i> for more information.

The parameters T, H, and n/d are not order dependent. They may be coded as needed without extra position holding commas.

If the UN parameter is used to define position, the UNITS parameter must precede this command in the OUTPUT command.

Example `IMAGE=(IMG1, 3, 2), END;`

INVERT

This parameter specifies that the following pages are to be inverted. When specified in a DJDE data stream (page oriented), this feature allows the Xerox DocuPrint 96/DocuPrint 96MX LPS to invert the image on a physical page by 180 degrees.

Syntax `INVERT = option(s)`

Options Table 8-32 lists the parameter option(s) and definition(s).

Table 8-32. **INVERT parameter option(s) and definition(s)**

Option(s)	Definition(s)
FRONT	Specifies page inversion on front sides.
BACK	Specifies page inversion on back sides.
BOTH	Specifies page inversion on both front and back sides.
NONE	Specifies turning page inversion off.

ITEXT

This parameter specifies a text message to be output during input processing. It takes effect on the next page boundary. The parameters are the same as for the MESSAGE ITEXT parameter defined in the "Print format commands" chapter."

Syntax ITEXT = *option(s)*

Options Table 8-33 lists the parameter option(s) and definition(s).

Table 8-33. ITEXT parameter option(s) and definition(s)

Option(s)	Definition(s)
NONE	Specifies that no text message is output to the operator during input processing. The default is NONE.
<i>sc</i>	Specifies a text message of up to 80 characters.
<i>passnum</i>	Specifies the pass (copy ply) to which the message text applies. The message is output to the operator just before processing of the indicated pass (copy ply) is begun. If no pass number is specified, the indicated message is output at the beginning of the first pass.

JDE

The *jde-id* specifies the JDE to be used within the selected JDL at the next page boundary. In a DJDE packet containing a JDE, other record- or page-oriented DJDE parameters included in the packet override parameters specified by this selected JDE. (Refer to the "Job parameter modification restrictions" section in this chapter.)

Syntax JDE = *option(s)*

Options Table 8-34 lists the parameter option(s) and definition(s).

Table 8-34. JDE parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>jde-id</i>	Specifies the JDE to be used within the selected JDL at the next page boundary.

JDL

The *jdl-id* specifies the name of the JDL to be invoked at the next page boundary. The *jdl-id* must exist on disk in the JDL directory. If only JDL is specified (not JDE) in the DJDE, the JDE to be used is the one named in the START command. In a DJDE packet containing a JDL parameter, other record- or page-oriented DJDE parameters override parameters specified by the JDL DJDE.

Syntax JDL = *option(s)*

Options Table 8-35 lists the parameter option(s) and definition(s).

Table 8-35. **JDL parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>jdl-id</i>	The <i>jdl-id</i> specifies the name of the JDL to be invoked at the next page boundary.

MAP

This parameter references a previously created font mapping file (file-name) from a [label:] SEFFNT command

Syntax MAP = *option(s)*

Options Table 8-36 lists the parameter option(s) and definition(s).

Table 8-36. **MAP parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>file-name</i>	References a previously created font mapping file (file-name) from a [label:] SEFFNT command

MARGIN

This parameter specifies the left printing margin within each logical page. It takes effect at the logical page following a DJDE END parameter. The parameters are the same as in the LINE command defined in the chapter "Print format commands."

Syntax MARGIN = *option(s)*

Options Table 8-37 lists the parameter option(s) and definition(s).

Table 8-37. **MARGIN parameter option(s) and definition(s)**

Option(s)	Definition(s)
{ <i>value</i> (<i>value</i> , <i>value-type</i>)}	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>value</i> The form <i>nnn.mm</i> (a positive decimal number with up to 2 digits to the right of the decimal point) which is the distance from the left margin. • <i>value-type</i> Indicates whether a value is specified in inches (IN), centimeters (CM), or character positions (POS). A value must be specified as an integer (<i>nnn</i>), if a value-type is character positions. <p>The default is (1,POS).</p> <p>Refer to the "Points to note" for the VOLUME command in the chapter "Specifying input parameters" for additional information.</p>

MODIFY

This parameter specifies the copy modification entry (CME) to be used in report processing. It takes effect at the next logical physical page boundary. A *cme-id* refers to a file which is separately cataloged in the CME directory. Refer to the "CME" section in the chapter "Print format commands" for further information. The parameters are the same as those in the OUTPUT command defined in the chapter "Print format commands."

Syntax MODIFY = *option(s)*

Options Table 8-38 lists the parameter option(s) and definition(s).

Table 8-38. **MODIFY parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>(cme-id,[init][,copies])NONE</i></p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>cme-id</i> The command identifier of a CME. The CME may be coded within a JSL or coded and cataloged separately for use by several JSLs. Any CME invoked by a DJDE must be cataloged separately (refer to the "Cataloged CMEs" section in this chapter for further information). If the CME is within the JSL, it must precede a reference to its identifier by the MODIFY parameter. Details on CME usage and syntax are discussed in the "CME" section of the "Print format commands" chapter. • <i>init</i> Specifies the initial ply (pass) to which the associated CME is to be applied. • <i>copies</i> Specifies the number of plies (passes) on which to apply the CME. If copies is not specified, then the CME applies to all copies beginning with the copy number specified by <i>init</i>. • NONE Specifies that data are processed without modification. <p>The default is NONE.</p>

NUMBER

This parameter specifies page numbering control. It takes effect on the next logical page boundary. The parameters are the same as those for OUTPUT command.

Syntax NUMBER = *option(s)*

Options Table 8-39 lists the parameter option(s) and definition(s).

Table 8-39. **NUMBER parameter option(s) and definition(s)**

Option(s)	Definition(s)
<p><i>(pnum, lnum, cnum [, findex])</i> NO</p>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>pnum</i> Page number: specifies the starting number (an integer) for page numbering. The beginning page number may be non-positive. The number is incremented at page transitions but not printed until it goes positive. The maximum page number that will be printed is 2,147,483,647. • <i>lnum</i> Line number: specifies an integer line number on which the page number is to be placed. • <i>cnum</i> Column number: specifies an integer ending column number for the page number sequence. • <i>findex</i> Font index: the list of fonts specified in the FONTS parameter of the PDE command. If not specified, the first font in the list is used. • NO Specifies that no page numbering is to be performed. <p>The default is NO.</p>

Example NUMBER=(1 , 1 , 40 , 1) , END ;

OTEXT

This parameter specifies a text message to be output to the operator during job printing. It takes effect on the next page boundary. The parameters are the same as for the MESSAGE command defined in the chapter "Print format commands."

Syntax OTEXT = *option(s)*

Options Table 8-40 lists the parameter option(s) and definition(s).

Table 8-40. **OTEXT parameter option(s) and definition(s)**

Option(s)	Definition(s)
NONE	Specifies that no text message is output to the operator during job printing. The default is NONE.
<i>sc</i>	Specifies a text message of up to 80 characters (maximum of 400 characters per report).
<i>passnum</i>	Specifies the pass (copy ply) to which the text applies. Multiple <i>sc</i> messages, one per <i>passnum</i> , may be specified in a JSL. The message is output to the operator prior to the beginning of printing the specified report ply. If no pass number is specified, the text is output once at the beginning of printing the entire report.
WAIT	Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.
END	Specifies that the text is displayed after the last copy of the report is printed.

OVERPRINT

This parameter specifies overprint control. Overprint lines are print lines whose carriage control specifies printing with no line spacing after the last printed line. It takes effect on the next record following a DJDE END; parameter. If FONTINDEX has been invoked in a JDE or DJDE, OVERPRINT takes effect at the next logical page boundary. OVERPRINT parameters are the same as for the LINE command.

Syntax OVERPRINT = *option(s)*

Options Table 8-41 lists the parameter option(s) and definition(s).

Table 8-41. **OVERPRINT parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>over-opt</i>	Options are as follows: <ul style="list-style-type: none"> • PRINT Specifies that all overprint lines are printed as they would be on an impact printer, with the second line printed over the top of the first line with no regard to previous data, including character spacing, which may vary between the two lines of data. • IGNORE Specifies that all overprint lines are ignored. • MERGE Specifies the same instructions as the PRINT option, except when used with FONTINDEX or CME processing. (See the "Points to note" section which follows.) • PRINT2 Specifies that up to two consecutive lines are printed per line—one line and one overprint. Other overprints for the line are ignored.
<i>disp</i>	Options are DISP and NODISP. These are nonfunctional parameter options which are preserved for compatibility with other Xerox laser printing systems. The number of overprint lines is always printed on the accounting page.

PMODE

This parameter specifies the printing mode for each page. LANDSCAPE indicates that printing is to be parallel to the long edge of paper. PORTRAIT indicates that printing is to be parallel to the narrow edge of paper. If PMODE is changed through a DJDE command, the BEGIN command should be respecified (even if the same horizontal and vertical position has been specified in the previous BEGIN parameter).

Syntax PMODE = *option(s)*

Options Table 8-42 lists the parameter option(s) and definition(s).

Table 8-42. **PMODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
LANDSCAPE	Indicates that printing is to be parallel to the long edge of paper.
PORTRAIT	Indicates that printing is to be parallel to the narrow edge of paper.

RFORM

This parameter specifies a form to be printed with all RTEXT pages. It takes effect on the next RTEXT page. The parameters are the same as for the ROUTE command defined in the chapter "Print format commands."

Syntax RFORM = *option(s)*

Options Table 8-43 lists the parameter option(s) and definition(s).

Table 8-43. **RFORM parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>form-id</i>	This is the name of a file cataloged in the FRM directory. It is created by compiling a file of forms source commands, called a JSL, with the FDL compiler. (Refer to the <i>Xerox DocuPrint 96/DocuPrint 96MX LPS Forms Creation Guide</i> .) The RFORM parameter is not allowed in a ROUTE command which is being defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file. Refer to the "Cataloged RTEXT files" section in this chapter for further details.
NONE	Specifies that no form is to be printed. The default is NONE.

RTEXT

This parameter specifies text to be printed on a separate page preceding a report. It takes effect on the next page boundary. The parameters are the same as for the ROUTE command defined in the chapter "Print format commands." RTEXT parameters must be in ascending passnum sequence.

Syntax RTEXT = *option(s)*

Options Table 8-44 lists the parameter option(s) and definition(s).

Table 8-44. RTEXT parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>(sc,[passnum ALL],[line,[col[, fontindex]]])</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>sc</i> Specifies the message to be printed (1 to 132 characters). It is printed with the first font specified in the FONTS parameter of the PDE command (refer to fontindex). • <i>passnum</i> May be either an integer number or the keyword ALL. An integer number specifies the pass (copy ply) to which the text applies. • ALL Specifies that the text applies to all copies. If there is more than one RTEXT parameter in a DJDE packet, passnum must be in increasing numeric order. The default is passnum = 1. • <i>line</i> Specifies the line number on which the first line of a block of RTEXT message is printed. The default is line 1 for the first text string of the pass. Otherwise, the default is the next line of the page. Only one RTEXT parameter may be specified per line for a passnum. • <i>col</i> Specifies the column number at which the first character of a block of RTEXT messages is to be printed. The default is column 1. • <i>fontindex</i> Identifies the index (starting with 1) of the PDE font with which the text is printed.
<i>rtext-id</i>	This option is used to reference a cataloged file of RTEXT parameters previously compiled by PDL.
NONE	Specifies that no text is to be printed on a separate page. The default is NONE.

Example RTEXT=('TEST' , ALL , 15 , 5 , 2) ;
END ;

SAVE

This parameter must occur prior to or at the end of the report and specifies IMG files updated during the current report which are not to be purged when the report's output processing is completed. This DJDE parameter is used to override, for specific files, automatic file deletions specified by PURGE=YES in the JDE's OUTPUT command.

Syntax SAVE = *option(s)*

Options Table 8-45 lists the parameter option(s) and definition(s).

Table 8-45. **SAVE parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>file-name</i> [,...])	The <i>file-name</i> is either the name of an IMG file or a file name mask whose form and interpretation are as described for the operator SAMPLE parameter (refer to "SAMPLE parameter").
ALL	Specifies that all document interleaved files created by this report's processing are to be saved, overriding an explicit (or default) specification in the JDE's OUTPUT command.

SEFMAP

This parameter specifies a mapping table different than the one defined in the PDL command on a page-by-page basis.

Syntax SEFMAP = *option(s)*

Options Table 8-46 lists the parameter option(s) and definition(s).

Table 8-46. **SEFMAP parameter option(s) and definition(s)**

Option(s)	Definition(s)
$((font_1, font_2)$ $, [font_m, font_n])$ $[, UPD REP]$	<p>This option has the following components:</p> <ul style="list-style-type: none"> • $font_1, font_2$ Font pair in which $font_2$ is the SEF mapping for $font_1$. • $font_m, font_n$ Font pair in which $font_n$ is the SEF mapping for $font_m$. • UPD (Update) indicates that the specified font pairs change existing mappings or append new mappings to the current font mapping table. • REP (Replace) indicates that the specified font pairs create a new mapping table to replace the current one. REP is the default. <p>With UPD or REP, only memory-resident font mapping tables (not disk-resident tables) are affected.</p>
NONE	Disables short-edge feed (SEF) and font mapping (stops a previous SEFFNT).

Considerations Refer to the SEFMAP examples at the end of this section.

SEPARATORS

This parameter specifies that the printer software save an internal copy of the next data page in the report and use it to generate segment separator sheets. If separators are already in effect when this DJDE is encountered, it replaces the prior specification and subsequent separators are generated using the new specification.

Syntax SEPARATORS = *option(s)*

Options Table 8-47 lists the parameter option(s) and definition(s).

Table 8-47. **SEPARATORS parameter option(s) and definition(s)**

Option(s)	Definition(s)
FIRST	Specifies that subsequent segments should have a copy of the saved data page inserted as the first sheet of the segment (on a separate sheet of paper) as a hardcopy segment separator.
(LAST[,D])	Specifies that at least one copy of the saved data page should be generated (on a separate sheet of paper) as a hardcopy segment separator and counted as the final sheet in the segment. Note: In an on-line environment, a true banner trailer page, if encountered, will be used to image any LAST separators of the final segment in place of the saved data page.
(BOTH[,D])	Specifies both FIRST and LAST, as above.
NONE	Specifies that no segment separator sheets are generated. However, the system continually updates segment numbering even if no separator sheets are printed. If SEPARATORS or SNUMBER are specified again in a DJDE, the updated number is used.

SF1FUNCTION

This parameter specifies to the finisher whether (YES) or not (NO) to begin a particular operation on the current sheet. SF1FUNCTION invokes the C6 signal of the electronic interface between the LPS and a third party finishing device. Refer to your vendor's third party device documentation for detailed information.

Syntax SF1FUNCTION = *option(s)*

Options Table 8-48 lists the parameter option(s) and definition(s).

Table 8-48. **SF1FUNCTION parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies to the finisher to begin a particular operation on the current sheet.
NO	Specifies to the finisher not to begin a particular operation on the current sheet.

SF2FUNCTION

This parameter specifies to the finisher whether (YES) or not (NO) to begin a particular operation on the current sheet. SF2FUNCTION invokes the C7 signal of the electronic interface between the LPS and a third party finishing device. Refer to your vendor's third party device documentation for detailed information.

Syntax SF2FUNCTION = *option(s)*

Options Table 8-49 lists the parameter option(s) and definition(s).

Table 8-49. **SF2FUNCTION parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies to the finisher to begin a particular operation on the current sheet.
NO	Specifies to the finisher not to begin a particular operation on the current sheet.

SHIFT

This parameter specifies a shift of the image on the page for purposes of binding. It takes effect on a physical page boundary.

Syntax SHIFT = *option(s)*

Options Table 8-50 lists the parameter option(s) and definition(s).

Table 8-50. **SHIFT parameter option(s) and definition(s)**

Option(s)	Definition(s)
v_1	An integer value in dots for the amount of shift on the simplex page or the odd (front) side of the duplex page. (Each dot is 1/300 of an inch.) Range for v_1 is -75 to 75.
v_2	A value in dots for the amount of shift on the even (back) side of a duplex page. Range for v_2 is -75 to 75. A standard size shift occurs if YES is specified ($v_1=75$ and $v_2=-75$).
YES	Specifies that a shift occurs. A standard size shift occurs if YES is specified ($v_1=75$ and $v_2=-75$).
NO	Specifies that no shift occurs ($v_1=0$ and $v_2=0$). The default is NO.

SIDE

This parameter specifies repositioning of the new logical page to the first logical page of the given side of a physical sheet. It takes effect at a logical page boundary. The parameters are the same as for the RPAGE command. The SIDE DJDE triggers repositioning for only a single logical page and does not otherwise override the SIDE parameter of the RPAGE command. It is intended to be used instead of the RPAGE command, not with the RPAGE command. The SIDE DJDE overrides both the RPAGE SIDE= parameter and the ROFFSET parameter.

Syntax SIDE = *option(s)*

Options Table 8-51 lists the parameter option(s) and definition(s).

Table 8-51. **SIDE parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(side-opt, offset-opt)</i>	This option has the following components: <ul style="list-style-type: none"> • <i>side-opt</i> • <i>offset-opt</i>

Considerations It is important to ensure that the data does not pass the margin and go off the page.

SNUMBER

This parameter specifies that the printer merge the current segment sequence number into any subsequent segment separator sheets produced. SNUMBER can be modified as a DJDE under the following conditions:

- Both SEPARATORS and SNUMBER are previously defined in either a JDE or JDL or as DJDEs.
- If the SEPARATORS parameter is not specified in the EXPORT command, it must be specified as a DJDE, along with the SNUMBER DJDE.

Syntax SNUMBER = *option(s)*

Options Table 8-52 lists the parameter option(s) and definition(s).

Table 8-52. **SNUMBER parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>(Inum, cnum, [findex])</i>	<p>This option has the following components:</p> <ul style="list-style-type: none"> • <i>Inum</i> Specifies the integer line number on which the segment sequence number string is to start. • <i>cnum</i> Specifies the integer column number on which the segment sequence number string is to end. • <i>findex</i> Specifies fontindex of the font to be used for printing the segment sequence number string. If SNUMBER is specified without SEPARATORS (under allowable conditions), the font index refers to the font list used at the time SEPARATORS was specified.

Considerations *Inum* and *cnum* are interpreted using the metrics of the font specified by *findex* (the same as with page numbering). Refer to the "Points to note" section for additional information.

SPLIT

This parameter specifies the use of segment management. When segment management is in effect, an end of segment condition occurs whenever max sheets (including separators) are generated or an end-of-copy or end-of-report condition occurs.

Syntax SPLIT = *option(s)*

Options Table 8-53 lists the parameter option(s) and definition(s).

Table 8-53. **SPLIT parameter option(s) and definition(s)**

Option(s)	Definition(s)
NOW	Specifies that the current or upcoming end-of-page (backside if duplex) will mark the last data page of the segment.
(<i>min, max</i>)	Specifies the minimum and maximum numbers of sheets allowed in a segment. The minimum number of sheets allowed is 2 and the maximum is 65,535. If a segment end has not occurred by the time max sheets are generated, then the printer declares end-of-segment. If end-of-segment occurs before min sheets are generated, then the printer generates padding of either blank or multiple LAST sheets (if the LAST option of the SEPARATORS parameter has been specified). If specified with segment management already in effect, then the current segment is tested against the new max and if greater, an end-of-segment is immediately declared.
OFF	Specifies that printer segment management not be provided, and if currently in effect, be immediately terminated.

Considerations When SPLIT=OFF is specified, the segment number (*nnn* of part *nnn* on separator sheets) is reset from its current value and begins at 001 again when segment management is turned on again.

SRECOVER

This parameter specifies the recovery mode. When used as a DJDE, SRECOVER must be specified prior to the first data record in the report or it is ignored.

Syntax SRECOVER = *option(s)*

Options Table 8-54 lists the parameter option(s) and definition(s).

Table 8-54. **SRECOVER parameter option(s) and definition(s)**

Option(s)	Definition(s)
PAGE	Specifies standard jam recovery. The printer will resume with the first sheet not delivered.
SEGMENT	Specifies the reprinting of the entire segment that was being printed at the time of the jam.
ASK	Specifies that the operator be requested through the user interface to select PAGE or SEGMENT recovery.
DEVICE	DEVICE - Places recovery mode selection responsibility with a support third party finishing device through connection with the Xerox DocuPrint 96/DocuPrint 96MX LPS bypass transport option.

STIMING

This parameter is provided for possible compatibility with other Xerox software supporting third party finishing devices and is ignored by the Xerox DocuPrint 96/DocuPrint 96MX LPS. The STIMING DJDE, although ignored by the Xerox DocuPrint 96/DocuPrint 96MX LPS, specifies that the printer accommodate timing constraints needed by finishing equipment interfaced with the Programmable Bypass Transport.

Syntax STIMING = *option(s)*

Options Table 8-55 lists the parameter option(s) and definition(s).

Table 8-55. **STIMING parameter option(s) and definition(s)**

Option(s)	Definition(s)
(INTERVAL, sec)	INTERNAL specifies the minimum allowable time between end-of segment signals (by way of the hardware interface at the bypass transport) for consecutive segments. If additional time is required, it is inserted after the last sheet is delivered and before the end-of-segment signal is emitted.
(DELAY, sec)	DELAY specifies the minimum allowable time between end-of-segment signal for one segment and delivery of the initial sheet of the next segment through the bypass transport,



Note: The sec specifies the time interval as a decimal number to a tenth of a second.

STOCKS

This parameter defines the STOCKSET and its associated stock(s) to be used in a report. If the stockset-name does not refer to a STOCKSET command coded earlier in the same JDL, it is assumed that the STOCKSET exists globally as an STK file, which is read at print time. In this case, PDL inserts a message in the JSL file listing, indicating that an STK file will be used at print time.

Whenever a new stockset is chosen, that is, at start of report or through a DJDE JDE or JDL switch, each stock is checked to determine that the stock exists and can be made active. This provides an automatic method of changing stocks in the LPS as required by the data stream.

Syntax STOCKS = *option(s)*

Options Table 8-56 lists the parameter option(s) and definition(s).

Table 8-56. **STOCKS parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>stockset-name</i>	Defines the STOCKSET used in a report.

TMODE

This parameter specifies process direction measurement for determining default pitch mode.

Syntax TMODE = *option(s)*

Options Table 8-57 lists the parameter option(s) and definition(s).

Table 8-57. **TMODE parameter option(s) and definition(s)**

Option(s)	Definition(s)
(<i>width</i> [, <i>width unit</i>])	<p>This option has the following components:</p> <ul style="list-style-type: none"> <i>width</i> The process direction measurement that determines the default pitch mode. <i>width unit</i> Can be IN, CM, DOTS, or XDOTS. The default is IN. Width values can range from the 7-inch equivalent to the 17-inch equivalent.

TOF

This parameter specifies the top-of-form line number. It takes effect at the next record following a DJDE END. The value parameter is defined the same as in the VFU command as defined in the chapter "Print format commands."

Syntax TOF = *option(s)*

Options Table 8-58 lists the parameter option(s) and definition(s).

Table 8-58. **TOF parameter option(s) and definition(s)**

Option(s)	Definition(s)
<i>value</i>	<p>Specifies the number of lines from the top of the output page to the first print line on the page (top of form). The top-of-form specification is independent of channel assignments. It should be greater than or equal to 1 and less than or equal to the smallest line number assigned to a channel.</p> <p>The default is 1.</p>

TRANS

This parameter specifies whether or not transparencies are used in the print job.

Syntax TRANS = *option(s)*

Options Table 8-59 lists the parameter option(s) and definition(s).

Table 8-59. **TRANS parameter option(s) and definition(s)**

Option(s)	Definition(s)
YES	Specifies that transparencies are used in the print job.
NO	Specifies that transparencies are not used in the print job.

XSHIFT

This parameter specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data for two-hole paper, binding, finishing, and edgemarking. XSHIFT allows shifting in the x-direction for short-edge binding.

Syntax XSHIFT = *option(s)*

Options Table 8-60 lists the parameter option(s) and definition(s).

Table 8-60. **XSHIFT parameter option(s) and definition(s)**

Option(s)	Definition(s)
NO	Specifies that no shift occurs.
<i>value</i>	An integer that specifies a shift amount on the front and back side. The shift is in the x-direction and each dot is 1/300 of an inch. The allowable range is 75 to -75.

Points to note

Note the following when using DJDE parameters:

- **TOF**

The DJDE top-of-form (TOF) command affects the first data record following the DJDE record. Depending on the time of a page transition, this may not take effect on the desired page. In other words, even though a DJDE TOF is processed during the processing of the next record, a page transition may have occurred before the DJDE TOF value could take effect. For example, if the initial TOF and BOF values are 2 for every job, every time a data record is processed, a page transition occurs since the TOF and BOF have the same value. If a DJDE TOF with value 1 is processed and the page transition has already occurred, this DJDE TOF value is too late and does not take effect until the next page transition.

- **Short-edge feed (SEF)**

If you want to use 11 x 17 inch paper (or any size that requires SEF) be aware that FSLs/FRMs, LGOs, and IMGs are not automatically scaled or rotated. Refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference* for detailed information on scaling.

- **FONTINDEX**

When multiple fonts of various sizes are used on a single page, the line spacing of any particular line is determined by the largest font in use in the previous line. The current line can also be adjusted downward by the difference in height between the first and the largest font in the current line. The line spacing of the first line in a report is determined by the first font in the PDE, provided the FONTINDEX parameter has not been specified. When FONTINDEX is specified, the line spacing of the first line of a report is determined by the font indicated by the font index value in the first line of the report. If there is no font index value in the record and FONTINDEX is specified, the line spacing for the first line of the report is determined by the first font in the PDE. When the FONTINDEX or CME parameters are specified, the line spacing of the first line of subsequent pages is determined by the line spacing of the font called out by the last font index of the previous page into the current PDL font list. If a PDE DJDE has occurred prior to the page transition, the font used for the line spacing comes from this list.

- **FONTINDEX, OVERPRINT=MERGE**

FONTINDEX, used in conjunction with overprinting, is handled as follows: for OVERPRINT=PRINT, the LPS overprints records analogously to an impact printer if the fonts are the same size. If the fonts differ, records are overprinted without regard to character spacing. For OVERPRINT=MERGE, the LPS replaces characters in the previous record which are blank. Character spacing values are adjusted; thus proportionally spaced or different size fonts may be used, and the LPS performs the character placement.

If FONTINDEX and OVERPRINT=MERGE are specified and proportional fonts are being used, data lines are not properly aligned if printing in landscape mode, and the data switches to a larger font in the middle of the line.

- **SNUMBER**

If the system encounters a valid SNUMBER dynamic job descriptor entry after an incorrect EXPORT command SNUMBER parameter had been specified, unpredictable results may occur.

SNUMBER as a DJDE may only be used to change current line or column numbers and fontindex values that are specified in the SNUMBER parameter of the EXPORT command used in the JDL. The DJDE SNUMBER may only enable segment sequence numbering if it is invoked with the SEPARATORS DJDE.

- **Ending a segment**

A segment may be ended by a DJDE under any of these conditions:

- SEPARATORS is specified in a DJDE
- The system encounters a SPLIT=NOW dynamic job descriptor entry
- The system encounters a SPLIT=OFF dynamic job descriptor entry.

- **Using a new separator page**

You may use a new separator page by specifying it as a DJDE.

- **Resetting the segment number**

You may set a new segment number after the current one is terminated. This is accomplished when the system encounters a series of two DJDEs: SPLIT=OFF followed by SPLIT=(min,max).

Examples

Following are examples of how to use the SEFMAP command.

Example 1

Before the DJDE short-edge feed parameter, the contents of the memory font mapping table where:

((font₁,font₂),(font₃,font₄))

DJDE SEFMAP parameter:

SEFMAP=*((font₁,font₇),(font₅,font₆),UPD)*;

After this DJDE SEFMAP parameter is processed, the contents of the memory font mapping table are:

((font₁,font₇),(font₃,font₄),(font₅,font₆))

Example 2

Before the DJDE short-edge feed parameter, the contents of the memory font mapping table where:

((font₁,font₂),(font₃,font₄))

DJDE SEFMAP parameter:

SEFMAP=*((font₁,font₇),(font₅,font₆),REP)*;

After this DJDE SEFMAP parameter is processed, the contents of the memory font mapping table are:

((font₁,font₇),(font₅,font₆))

DJDE FILE parameter

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 may be invoked solely for the purpose of downloading files; in other situations, the files may 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) that is to be copied to the LPS system disk. The DJDE packet must include a FILE parameter used to signal the start of the file data.

The DJDE FILE parameter is record oriented and is therefore applied immediately. It may be included with other record-oriented or page-oriented DJDE parameters in a DJDE packet. The FILE parameter is specified as follows:

FILE = ([*file-name*],[*file-type*],[*f*],[*s*],[*n*]])

Table 8-61 lists the parameter option(s) and definition(s).

Table 8-61. FILE parameter option(s) and definition(s)

Option(s)	Definition(s)
<i>file-name</i>	This is a 1 to 6 character name used to identify the disk file.
<i>file-type</i>	This is a 3 character file type symbol.
<i>f</i>	This is one of the following file input format parameters: C = card-image format L = LPS-labeled tape format (default)
<i>s</i>	This is one of the following file storage parameters: D = delete after report printed P = permanent (default)
<i>n</i>	This is the maximum number of card-images (default=120).

For LPS-labeled files, the DJDE FILE parameter may be specified as follows:

FILE=()

In this case, the file name and file type are obtained from the label record that precedes the file data.

Points to note

Note the following when using the DJDE FILE parameter:

- If the file name or file type specified in the DJDE FILE parameter differ from those in the label record, the file is renamed as specified in the label record. This renaming feature for LPS-labeled files allows five restricted types of files to be downloaded as different file types.
- If a disk file already exists with the specified file name, it is replaced. Refer to the "File replacement and deletion" section for details and restrictions.
- For card-image data, the FILE parameter may be specified as follows to indicate the end of card-image data:
FILE=(,C) or FILE=(END,,C)
- The single characters C, L, D, and P are file storage parameter keywords for the DJDE FILE=parameter and cannot be specified as the file-name in a FILE DJDE.
- The DJDE DELETE parameter, DJDE FILE, allows the removal of the following types of files only: IMG, FRM, LGO, FNT, CME, JDL, PDE, TST, and XCS.

File effectiveness

Files may be used by input or output as soon as they are written to the system disk. However, they become effective only when they are referenced or invoked by a DJDE. For example, a font or form file becomes effective at the next page boundary when a DJDE changes the FORMAT or FORMS parameter to invoke the new font or form file. If a file on the disk is being replaced, the new file is not to be used until it is reinvoked by an appropriate DJDE.

File replacement and deletion

Files that may be used by input or output processing during the course of a print job can be replaced or deleted only when they are no longer required. If a file can be used by output processing, for example, FNT, FRM, IMG, and LGO, the file replacement transaction entails the temporary retention of the replaced file until the current report is printed. A file that is used only by input processing, such as CME and PDE, is replaced immediately. Thus, the files marked for deletion after the current report is printed include those that the user had specified for deletion and the superseded files that may have been required by output. Files which are transferred with a delete option are available for deletion as soon as the report is printed and therefore must not be referenced in a subsequent report.



Note: Unlike the hostcopy procedure, the DJDE FILE parameter does not allow the file replacement or deletion transaction to occur if the file to be replaced or deleted is secured and the operator's logon level is not 5.

Card-image file processing

The FILE parameter 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 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 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 are predicted by the parameter specifying the maximum number of card images, the extra records are read and discarded, and 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 due to insufficient space on the disk, the preceding DJDE packet is printed on an OPRINFO page 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 in effect.

Sequence numbers are not inserted into the records prior to storing, so the file must be retrieved by MERGE prior to editing in order to insert valid sequence numbers.

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

and 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 parameter.

LPS-labeled file processing

The FILE parameter 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 together with the message:

```
FILE PROCESSING ERROR - NO LABEL RECORD.
```

Input then proceeds to process the data as variable text data.

- If the destination file type is OSD, SAF, SYS, \$\$\$, or TSK, or if the file type specification in the DJDE parameter is not one of the file types included in the LPS file directory, the preceding DJDE packet is printed on an OPRINFO page 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 cannot be created due to 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 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 parameter 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 with a brief indication of whether the file was newly created or replaced 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 non-blank character at the end of each record. The same non-blank character must be used with all records processed in the file processing mode.

If the non-blank 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 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.

The Advanced Image 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:

- Magnetic tapes with fixed or variable record/block structures, that is, 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 software package or equivalent.

This chapter contains information on the following topics:

- Graphics considerations
- Input for graphics
- Processing modes
- Tape formats
- Online formats
- PDL command and DJDE options for graphics
- Performance
- Restrictions

Graphics considerations

There are several PDL parameters available for graphics handling. These include four JDE OUTPUT command parameters and six DJDEs.

Tables 9-1 and 9-2 summarize these parameters. The OUTPUT command parameters are described in the "OUTPUT command" section in the chapter "Print format commands." The graphics DJDE parameters were discussed previously in the chapter "Specifying dynamic job descriptor entries (DJDEs)."

Graphics DJDEs are record oriented because they take effect immediately. For example, ALTER, CANCEL, and IMAGE may appear anywhere on the page and are applied to the page on which they occur. Positioning GRAPHIC, BATCH, and SAVE DJDEs is discussed in table 9-2, "DJDE parameter definitions." These also take effect immediately.

Page-oriented DJDEs to be applied to the next page must not precede batch mode or page interleaved graphics to be applied to the current page.

Note that your Xerox DocuPrint 96/DocuPrint 96MX LPS can handle single-color graphics from a Xerox 4850 or 4890 HighLight Color LPS, but cannot process 2-color graphics.

Input for graphics

Named graphics may be input to the LPS through magnetic tape, the online channel, or Ethernet. 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:

- Batch
- Block
- Move
- Random

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 command. (Refer to the chapter "Print format commands," section "Specifying output parameters.")



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 parameter of the OUTPUT command.

Tape formats

This section covers the following topics:

- Noninterleaved
- Document interleaved
- Page interleaved
- Batch mode

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.

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:

- The tape format may be any legal fixed or variable length block, or record format which is otherwise supported by the LPS, that is, RECORD STRUCTURE = F, FB, V or VB. It may not be an undefined block or record format, that is, RECORD STRUCTURE = U or UB. Packed data formats are not supported.
- Use of block and record constants in an appropriately structured file (type F, FB, V, or VB) containing interleaved graphics is supported. However, the actual delimiter searching is suspended while an IMG file is being read.
 - 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.
 - 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.
- Each graphic must be preceded immediately by a GRAPHIC DJDE.
- 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, that is, 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:

- The graphics must occur on the tape immediately following the last text to appear on the same physical page.
- 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: 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 section covers the following topics:

- Noninterleaved
- Document and page interleaved
- Batch mode
- Document interleaved graphic file transfers
- Management of image files

Noninterleaved

HOSTCOPY is the mechanism for transferring noninterleaved graphics through 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:

- Each graphic must be preceded immediately by a DJDE GRAPHIC record.
- There is no change in the maximum record length when processing interleaved graphic records. However, 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 is unprintable and the results may be unpredictable.
- 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 magnetic tape.

The input data stream format must comply with the following rules:

- 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 followed immediately by a DJDE specifying BATCH=END. These batch stream delimiters are necessary to distinguish banner page data (header and trailer) from graphic data.
- 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.
- There is no change in the maximum record length when processing batch mode graphic records. However, 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 command 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 image data, the LPS file id must be saved in ASCII in the LPS 128-byte file label.

PDL command and DJDE options for graphics

Several PDL commands are available for graphic handling. These include four job descriptor entry (JDE) OUTPUT commands and six dynamic job descriptor entries (DJDEs).

Tables 9-1 and 9-2 summarize these commands.

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 lists the OUTPUT command parameters and functions for graphics

Table 9-1. **OUTPUT command parameters and functions for graphics**

Option(s)	Definition(s)
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 lists the DJDE definitions for graphics

Table 9-2. **OUTPUT command parameters and functions for graphics**

Option(s)	Definition(s)
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 option of the GRAPHICS DJDE.
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 or purged.

Refer to the chapter "Specifying dynamic job descriptor entries (DJDEs)" for more information on DJDEs.

Performance

This section covers the following topics:

- Random mode
- Online
- Document interleaved file creation

Random mode

Random mode requires a separate disk access by the output processing task for every graphic on a page. The more graphics on a page and the larger the graphics, the more likely it is that paper path holes are created.

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.



Note: The restrictions which follow are applicable without Interpress.

Graphic feature restrictions

Note the following graphic feature restrictions:

- Tape formats supported by interleaved graphic processing are restricted to LPS compatible fixed or variable record structures, that is, 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.
- There is no allowance or “tolerance” available for the paper size in a JDE for a graphic to be off the page.
- No online banner page (neither header nor trailer) may include graphic references.
- A JDE which is selected through 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.

A. PDL command and DJDE summary

This appendix may be used as a quick reference for information about PDL commands and command parameters. The following definitions describe the conventions used in this appendix.

Conventions

Symbols used in the command syntax are listed in table A-1.

Table A-1. **Symbols used in command syntax**

Symbol	Meaning
<i>ac</i>	Command identifier. Consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be alpha.
<i>dd</i>	Command identifier. Same as <i>ac</i> except restriction of having one alpha character is removed. This identifier applies only to SYSTEM (or JDL) and JOB (or JDE) command sets.
<i>id</i>	Reference to a command identified by an <i>ac</i> command identifier.
<i>sc</i>	String constant. A hexadecimal, octal, ASCII, EBCDIC, or character constant.
<i>value</i>	A decimal constant.
<i>s</i>	The "s" at the end of some PDL commands and DJDEs is optional.

The LPS column indicates the laser printing system to which the command or parameter applies. ALL means that the command is applicable to all laser printing systems operating with version 3 software or higher.

The "Online" and "Offline" columns specify the input data stream modes of operation for the LPS. A Y for a particular parameter means the parameter is available for that processing mode. An N means the parameter is not available for that mode.

The "DJDE orientation" column indicates whether a command or parameter has an equivalent DJDE and, if so, specifies whether the DJDE is page- or record-oriented.

Table A-2. PDL commands and DJDEs

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
ABNORMAL	ERROR	STOP	All	Y	Y	—
	IMISMATCH	STOP	4850, 4890	Y	Y	—
	ISUBSTITUTE	ANY	4850, 4890	Y	Y	—
	OTEXT	NOWAIT	All	Y	Y	—
	SECURITY	NO	All	Y	Y	—
ACCT	DEPT	<i>jdl-name</i>	All	Y	Y	Page
	USER	BIN	All	Y	Y	—
BANNER	HCOUNT	0	All	N	Y	—
	HJOBNO	NONE	All	N	Y	—
	HRPTNA	NONE	All	Y	Y	—
	TCOUNT	0	All	N	Y	—
	TEST	—	All	N	Y	—
	TYPE	BANNER	All	N	Y	—
BDELETE	TEST	—	All	Y	N	—
BLOCK	ADJUST	0	All	Y	N	—
	CONSTANT	—	All	Y	N	—
	FORMAT	BIN	All	Y	N	—
	LENGTH	1330	All	Y	N	—
	LMULT	1	All	Y	N	—
	LTHFLD	0	All	Y	N	—
	OFFSET	0	All	Y	N	—
	POSTAMBLE	0	All	Y	N	—
	PREAMBLE	0	All	Y	N	—
	ZERO	NO	All	Y	N	—
BSELECT	TEST	—	All	Y	N	—
<i>ac</i> :CATALOG	—	—	All	Y	Y	—
<i>ac</i> :CME	CONSTANT	—	All	Y	Y	—
	FONTS	—	All	Y	Y	—
	INK	—	4850, 4890	Y	Y	—
	LINE	—	All	Y	Y	—
	POSITION	—	All	Y	Y	—
<i>ac</i> :CODE	ASSIGN	—	All	Y	Y	—
	DEFAULT	EBCDIC	All	Y	Y	—
<i>ac</i> :CRITERIA	CHANGE	—	All	Y	Y	—
	CONSTANT	—	All	Y	Y	—
	LINENUM	ALL LINES	All	Y	Y	—
END	—	—	All			—
EXPORT	SEPARATORS	NONE	4850, 4890, 4135, 4635	Y	Y	—
	SNUMBER	—	4850, 4890, 4135, 4635	Y	Y	Page
	SPLIT	—	4850, 4890, 4135, 4635	Y	Y	Page
	SRECOVER	PAGE	4850, 4890, 4135, 4635	Y	Y	Page
	STIMING	—	4850, 4890, 4135, 4635	Y	Y	Page

Table A-2. PDL commands and DJDEs (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
FILE	—	—	All	N	Y	Record
IDEN	OFFSET OPRINFO PREFIX SKIP	0 NO — 1	All All All All	Y Y Y Y	Y Y Y Y	— — — —
<i>ac</i> :IDR	ICATALOG ILIST PALETTE	DFAULT — DFAULT	4850, 4890 4850, 4890 4850, 4890	Y Y Y	Y Y Y	Page Page Page
JDE	INCLUDE	—	All	Y	Y	Page
JDL		—	All	Y	Y	Page
<i>ac</i> :JOB	INCLUDE	—	All	Y	Y	—
LINE	DATA FCB FONTINDEX INKINDEX MARGIN OVERPRINT PCC PCCTYPE UCSB VFU	offline=1,132 online=0,150 PROCESS NONE NONE 1,POS PRINT,NODISP 0,NOTRAN offline=ANSI online=IBM3211 PROCESS NONE	All All All All 4850, 4890 All All All All All	Y Y N Y Y Y Y Y N Y	Y Y Y Y Y Y N Y Y Y	Record Record — Page Page Page Record — — — —
MESSAGE	ITEXT OTEXT	NONE NONE	All All	Y Y	Y Y	Page Page

Table A-2. PDL commands and DJDEs (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
OUTPUT	BFORM	NONE <i>init</i> defaults to first or next copy	All	Y	Y	Page
	COLLATE	YES	All	Y	Y	Page
	COPIES	1	All	Y	Y	Page
	COVER	NONE	All	Y	Y	—
	CYCLEFORMS	NONE	All	Y	Y	—
	DENSITY	DEFAULT	All	Y	Y	—
	DESTINATION	NO	All	Y	Y	—
	DUPLEX	NO	All	Y	Y	Page
	FACEUP	NO	All	Y	Y	—
	FEED	OPR	All	Y	Y	Page
	FORMAT	FMT1	All	Y	Y	Page
	FORMS	NONE <i>init</i> defaults to first or next copy; FOR defaults to FORMAT if FORMS not fully keyed	All	Y	Y	Page
	GRAPHICS	NO	All	Y	Y	Record
	IDFAULT	first ink in ILIST parameter	4850, 4890	Y	Y	Page
	IDR	DFIDR.IDR	4850, 4890	Y	Y	Page
	IMAGE	inches, top left corner scaled at 1/1	All	Y	Y	—
	INVERT	NONE	4635	Y	Y	Page
	IRESULT	sysgened default	4850, 4890	Y	Y	Page
	MODIFY	NONE	All	Y	Y	Page
	NT01	NO	All	Y	Y	—
	NUMBER	NO	All	Y	Y	Page
	OFFSET	ALL	All	Y	Y	—
	OSTK	RES MOD and POS default=1; SIZE=.5 inch sysgened default	4135, 4635	Y	Y	—
	PAPERSIZE	YES	All	Y	Y	—
	PURGE	sysgened default	All	Y	Y	—
	RESOLUTION	NO	All	Y	Y	—
	SF1FUNCTION	NO	4850, 4890, 4135, 4635	Y	Y	Page
	SF2FUNCTION	NO	4850, 4890, 4135, 4635	Y	Y	Page
	SHIFT	—	All	Y	Y	Page
	SIZING	NO	All	Y	Y	—
	STAPLE	—	All	Y	Y	—
	STOCKS	—	All	Y	Y	Record
	SYSPPR	depends on papersize;	4135, 4635	Y	Y	—
	TMODE	default value is inches	4135, 4635	Y	Y	Page
	TRANS	—	4135, 4635	Y	Y	Page
	UNITS	—	All	Y	Y	—
XMP	DEFAULT	4850, 4890	Y	Y	Page	
XSHIFT	NONE	4635	Y	Y	Page	

Table A-2. PDL commands and DJDEs (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
ac:PCC	ADVTAPE	YES	All	Y	N	—
	ASSIGN	—	All	Y	N	—
	DEFAULT	DEFAULT	All	Y	N	—
	INITIAL	TOF	All	Y	N	—
	MASK	X'FF'	All	Y	N	—
ac:PDE	BEGIN	.18 IN, .66 IN	All	Y	Y	Page
	FONTS	L0112B	All	Y	Y	Page
	PMODE	LANDSCAPE	All	Y	Y	Page
RAUX	TEST	—	All	Y	Y	—
RDELETE	TEST	—	All	Y	Y	—
RECORD	ADJUST	0	All	Y	N	—
	CONSTANT	—	All	Y	N	—
	FORMAT	BIN	All	Y	N	—
	LENGTH	offline=133, online=150	All	Y	Y	—
	LMULT	1	All	Y	N	—
	LTHFLD	0	All	Y	N	—
	OFFSET	0	All	Y	N	—
	POSTAMBLE	0	All	Y	N	—
PREAMBLE	0	All	Y	N	—	
STRUCTURE	FB	All	All	Y	N	—
RFEED	TEST	no default for <i>clu-def</i>	4050, 4090, 4650 (V3.8), 4850 (V5.0), 4890, 4135, 4635	Y	Y	—
ROFFSET	PASSES	ALL	All	Y	Y	—
	TEST	—	All	Y	Y	—
ac:ROUTE	RFORM	NONE	All	Y	Y	Page
	RTEXT	NONE	All	Y	Y	Page
RPAGE	SIDE	NUFRONT, NOFFSET	All	Y	Y	Page
	TEST	—	All	Y	Y	—
	WHEN	TOP	All	Y	Y	—
RRESUME	BEGIN	NEXT	All	Y	Y	—
	TEST	—	All	Y	Y	—
RSELECT	TEST	—	All	Y	Y	—
RSTACK	ACCTINFO	—	All	Y	Y	—
	DELIMITER	NO	All	Y	Y	—
	HRPTNA	NONE	All	Y	Y	—
	PRINT	NONE	All	Y	Y	—
	TEST	—	All	Y	Y	—
RSUSPEND	BEGIN	NEXT	All	Y	Y	—
	TEST	—	All	Y	Y	—
ac:SEFFNT	SEFMAP	—	4635	Y	Y	Page
SEFFNT	MAP	—	4135, 4635	Y	Y	Page
	SEFMAP	—	4135, 4635	Y	Y	Page

Table A-2. PDL commands and DJDEs (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
<i>ac</i> :STOCKSET	ASSIGN	—	All	Y	Y	—
	INIFEED	<i>first stock-name</i>	All	Y	Y	—
	SYSPAGE	—	All	Y	Y	—
<i>ac</i> :SYSTEM		—	All	Y	Y	—
<i>ac</i> :TABLE	CONSTANT	—	All	Y	Y	—
	MASK	—	All	Y	Y	—
<i>ac</i> :TCODE	DEFAULT	—	All	Y	Y	—
	TASSIGN	—	All	Y	Y	—
	TRESET	—	All	Y	Y	—
<i>ac</i> :VFU	ASSIGN	—	All	Y	Y	Record
	BOF	66	All	Y	Y	Record
	TOF	1	All	Y	Y	Record
VOLUME	BMULT	1	All	Y	N	—
	CODE	EBCDIC	All	Y	Y	—
	EOV	NOPAUSE,NOEOF	All	Y	N	—
	HOST	IBMOS	All	Y	Y	—
	INTERPRESS	YES, NOCHECK, NOBREAKPAGE, SIMPLE, INTEGRAL, PERFORMANCE, RIP	All	Y	N	—
	LABEL	STANDARD	All	Y	N	—
	LCODE	EBCDIC	All	Y	N	—
	LPACK	NO	All	Y	N	—
	MAXLAB	81	All	Y	N	—
	MINLAB	80	All	Y	N	—
	OPTIMIZE	NONE	All	N	Y	—
	OSCHN	9	All	Y	N	—
	OSHDP	0	All	Y	N	—
	OSTLP	0	All	Y	N	—
	PLABEL	NO	All	Y	N	—
	RMULT	1	All	Y	N	—
	RSAT	REMOUNT	All	Y	N	—
	TCODE	EBCDIC	All	Y	Y	—
	UNPACK	NONE	All	Y	N	—

B. PDL command quick reference

Conventions

Symbols used in the command syntax are listed in table A-1.

Table B-1. **Symbols used in command syntax**

Symbol	Meaning
<i>ac</i>	Command identifier. Consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be alpha.
<i>dd</i>	Command identifier. Same as <i>ac</i> except restriction of having one alpha character is removed. This identifier applies only to SYSTEM (or JDL) and JOB (or JDE) command sets.
<i>id</i>	Reference to a command identified by an <i>ac</i> command identifier.
<i>sc</i>	String constant. A hexadecimal, octal, ASCII, EBCDIC, or character constant.
<i>value</i>	A decimal constant.
<i>italics</i>	Indicate variable names or numbers.
underlining	Indicates the default parameter or parameter option.
braces	{ } Indicate required options.
brackets	[] Indicate optional choices.
parenthesis	() Used when more than one parameter option is listed after a parameter.
. . .	(Ellipsis) Indicates repetition of a parameter option or a series of parameter options.

Table B-2. PDL command quick reference

Command	Syntax	Default
ABNORMAL	ABNORMAL ERROR = {CONTINUE ABORT STOP} ABNORMAL IMISMATCH = {CONTINUE ABORT STOP} ABNORMAL ISUBSTITUTE = {ANY NONE} ABNORMAL OTEXT = {WAIT NOWAIT} ABNORMAL SECURITY = {YES NO}	STOP STOP ANY NOWAIT NO
ACCT	ACCT DEPT = <i>sc</i> ACCT USER = {BIN TRAY BOTH NONE}	<i>jdl-name</i> BIN
BANNER	BANNER HCOUNT = <i>value</i> BANNER HJOBNO = {(<i>offset</i> , <i>length</i>) NONE} BANNER HRPTNA = {(<i>offset</i> , <i>length</i>) NONE} BANNER TCOUNT = <i>value</i> BANNER TEST = <i>test-exp</i> BANNER TYPE = {DATE BANNER}	<i>test-exp</i> NONE NONE 0 — BANNER
BDELETE	BDELETE TEST = <i>test-exp</i>	—
BLOCK	BLOCK ADJUST = <i>value</i> BLOCK CONSTANT = <i>sc</i> BLOCK FORMAT = <i>type</i> BLOCK LENGTH = <i>value</i> BLOCK LMULT = <i>value</i> BLOCK LTHFLD = <i>value</i> BLOCK OFFSET = <i>value</i> BLOCK POSTAMBLE = <i>value</i> BLOCK PREAMBLE = <i>value</i> BLOCK ZERO = {YES NO}	0 — BIN 1330 1 0 0 0 0 0 —
BSELECT	BSELECT TEST = <i>test-exp</i>	—
<i>ac</i> :CATALOG	<i>catalog-name</i> : CATALOG	—
<i>ac</i> :CME <i>4850 command only</i>	<i>ac</i> :CME CONSTANT = <i>sc</i> CME FONTS = <i>value</i> CME LINE = { <i>n</i> (<i>n</i> , <i>m</i>) (<i>n</i> , -1)} CME POSITION = <i>n</i> CME INK = { <i>inkindex</i> 0}	— — — — —
<i>ac</i> :CODE	<i>ac</i> :CODE ASSIGN = {(<i>input</i> , <i>output</i>) (<i>input</i> , (<i>output</i> , ..., <i>output</i>))} <i>ac</i> :CODE DEFAULT = <i>code-type</i>	— EBCDIC

Table B-2. PDL command quick reference (continued)

Command	Syntax	Default
ac:CRITERIA	ac:CRITERIA CHANGE = (<i>offset</i> , <i>length</i> , NE, LAST) CONSTANT = (<i>offset</i> , <i>length</i> , {EQ NE}, <i>tab-id</i>) ac:CRITERIA LINENUM = (<i>int</i> , <i>count</i>)	— — ALL LINES
END	END;	—
EXPORT	SEPARATORS = {FIRST LAST [,D]} (BOTH[,D]) NONE} SNUMBER = (<i>Inum</i> , <i>cnum</i> , <i>findex</i>) SPLIT = {NOW (<i>min</i> , <i>max</i>) OFF} SRECOVER = {PAGE SEGMENT ASK DEVICE} STIMING = {(INTERVAL,sec) (DELAY,sec) (INTERVAL,sec ,DELAY,sec)}	NONE — — PAGE —
IDEN	OFFSET = <i>value</i> OPRINFO = {YES NO} PREFIX = <i>sc</i> SKIP = <i>value</i>	0 NO — 1
ac:IDR (4850 LPS only)	IDR ICATALOG = <i>ink-catalog-name</i> IDR ILIST = {' <i>ink-name</i> ' (' <i>ink-name</i> ',[, ' <i>ink-name</i> '],[,...])} IDR PALETTE = ' <i>palette-name</i> '	DFAULT — DFAULT
JDE	[INCLUDE = (<i>catalog-id</i> ₁ [, <i>catalog-id</i> ₂] [,...])]	—
JDL	JDL = <i>jdl-id</i>	—
ac:JOB	[INCLUDE = (<i>catalog-id</i> ₁ [, <i>catalog-id</i> ₂] [,...])]	—
LINE	LINE DATA = (<i>pdo</i> , <i>length</i>) LINE FCB = {IGNORE PROCESS} LINE FONTINDEX = { <i>offset</i> <i>offset</i> ([, <i>init-val</i>] [, <i>bit-opt</i>]) NONE} LINE INKINDEX = { <i>offset</i> <i>offset</i> ([, <i>init-val</i>] [, <i>bit-opt</i>]) NONE} LINE MARGIN = { <i>value</i> (<i>value</i> , <i>value-type</i>)} LINE OVERPRINT = (<i>over-opt</i> , <i>disp</i>) LINE PCC = (<i>offset</i> , <i>trans-type</i>) LINE PCCTYPE = <i>pcc-name</i> LINE UCSB = {IGNORE PROCESS} LINE VFU = { <i>vfu-id</i> NONE}	offline=1,134; online=0,150 PROCESS NONE NONE 1,POS PRINT,NODISP 0,NOTRAN offline=ANSI online=IBM3211 PROCESS NONE
MESSAGE	MESSAGE ITEXT = { <i>sc</i> (<i>sc</i> , <i>passnum</i>) NONE} MESSAGE OTEXT = { <i>sc</i> (<i>sc</i> , <i>passnum</i>)[, WAIT] <i>sc</i> [, END][, WAIT] NONE}	NONE NONE

Table B-2. PDL command quick reference (continued)

Command	Syntax	Default
OUTPUT	OUTPUT BFORM = { <i>form-id</i> (<i>form-id</i> [, <i>ini</i> <i>f</i> [, <i>copies</i>]] [, INK[S], <i>inkref</i> [, <i>inkref</i>] [, ...]])}	NONE
	OUTPUT COLLATE = {YES NO}	YES
	OUTPUT COPIES = <i>number</i>	1
	OUTPUT COVER = <i>cover-opt</i>	NONE
	OUTPUT CYCLEFORMS = { <i>form-id</i> [, <i>form-id</i>] [, ...] (<i>form-id</i> [, (INK[S], <i>inkref</i> [, <i>inkref</i>] [, ...])]) ([, <i>form-id</i> [INK[S], <i>inkref</i> [, <i>inkref</i>] [, ...]])] NONE}	NONE
	OUTPUT DENSITY = {FIX NOFIX DEFAULT}	DEFAULT
	OUTPUT DESTINATION = {BIN TRAY EXPORT}	—
	OUTPUT DUPLEX = {YES NO}	NO
	OUTPUT FACEUP = {YES NO}	NO
	OUTPUT FEED = { <i>stock-reference</i> <i>stock-name</i> MAIN AUX OPR}	OPR
	OUTPUT FORMAT = <i>pde-id</i>	FMT1
	OUTPUT FORMS = { <i>form-id</i> (<i>form-id</i> [, <i>init</i> [, <i>copies</i>]] [, (INK[S], <i>inkref</i> [, <i>inkref</i>] [, ...])]) NONE}	NONE <i>init</i> defaults to first or next copy FOR defaults to FORMAT if FORM[S] is not fully keyed in.
	OUTPUT GRAPHICS = {NO YES MOVE BATCH (YES, NOSUB) (MOVE, NOSUB)}	NO
	OUTPUT IDFAULT = <i>inkref</i>	First ink in ILIST parameter
	OUTPUT IDR = <i>idr-name</i>	DFIDR.IDR
	OUTPUT IMAGE = (<i>vpos</i> {un cm in dots xdots} <i>hpos</i> {, {un cm in dots xdots} [, <i>n</i> [, <i>d</i>]]} [, INKS, <i>inkref</i> [, <i>inkref</i>] [, ...])	Inches, top left corner scaled at 1/1
	OUTPUT INVERT = {FRONT BACK BOTH NONE}	
	OUTPUT IRESULT = {BLACK COLOR}	Sysgened default
	OUTPUT MODIFY = { <i>cme-id</i> (<i>cme-id</i> [, <i>init</i> [, <i>copies</i>]) NONE}	NONE

Table B-2. PDL command quick reference (continued)

Command	Syntax	Default
OUTPUT	OUTPUT NTO1 = {YES NO <i>n</i> } OUTPUT NUMBER = {(<i>pnum</i> , <i>lnum</i> , <i>cnum</i> [, [<i>findex</i>][, <i>inkref</i>]) NO} OUTPUT OFFSET = {ALL FIRST NONE} OUTPUT OSTK = (<i>clu-def</i> , [<i>mod</i>], [<i>pos</i>], [RES NORES], [TABS] , [<i>size</i> DOTS IN CM][, L P]) OUTPUT PAPERSIZE = {A3 A4 B4 USLEGAL USLETTER (<i>x</i> , <i>y</i>)} OUTPUT PURGE = {YES NO} OUTPUT RESOLUTION = {300 600} OUTPUT SF1FUNCTION = {YES NO} OUTPUT SF2FUNCTION = {YES NO} OUTPUT SHIFT = {(<i>v</i> ₁ , <i>v</i> ₂) YES NO} OUTPUT SIZING = {SEMIAUTO BEST EXACT} OUTPUT STOCKS = <i>stockset-name</i> OUTPUT SYSPPR = {USLEGAL A3 A4 B4 USLETTER (<i>x</i> , <i>y</i>)} OUTPUT TMODE = { <i>width</i> (<i>width</i> [, <i>width unit</i>])} OUTPUT TRANS = {YES NO} OUTPUT UNITS = { <i>size in dots</i> } OUTPUT XMP = {DEFAULT REPORT} OUTPUT XSHIFT = {NO <i>value</i> }	NO NO ALL Sysgened default YES Sysgened default NO NO NO SEMIAUTO — Paper size of the last sheet that is before the system page. NO — DEFAULT NO
<i>ac</i> :PCC	<i>ac</i> :PCC ADVTAPE = {YES NO} <i>ac</i> :PCC ASSIGN = {(<i>byte</i> , <i>ccln</i>) (<i>byte</i> , <i>ccln</i> [, <i>ccln</i>][, ...])} <i>ac</i> :PCC DEFAULT = { <i>ccln</i> <i>pcctype</i> } <i>ac</i> :PCC INITIAL = {BOF TOF} <i>ac</i> :PCC MASK = { <i>value</i> }	YES — DEFAULT TOF X'FF'
<i>ac</i> :PDE	<i>ac</i> :PDE BEGIN = (<i>vpos</i> , <i>hpos</i>) <i>ac</i> :PDE FONTS = {(<i>f</i> ₁ [, <i>f</i> ₂][, ...]) ((<i>f</i> ₁ , <i>s</i> ₁)[, <i>f</i> ₂ , <i>s</i> ₂][, ...]) ((<i>f</i> ₁ , <i>s</i> ₁ [pi <i>xdots</i> dots])[, <i>f</i> ₂ , <i>s</i> ₂ [pi <i>xdots</i> dots]][, ...])} <i>ac</i> :PDE PMODE = {LANDSCAPE PORTRAIT}	.18 IN, .66 IN L0112B LANDSCAPE
RAUX	RAUX TEST = <i>test-exp</i>	—
RDELETE	RDELETE TEST = <i>test-exp</i>	—

Table B-2. PDL command quick reference (continued)

Command	Syntax	Default
RECORD	RECORD ADJUST = <i>value</i> RECORD CONSTANT = <i>sc</i> RECORD FORMAT = <i>type</i> RECORD LENGTH = <i>value</i> RECORD LMULT = <i>value</i> RECORD LTHFLD = <i>size</i> RECORD OFFSET = <i>value</i> RECORD POSTAMBLE = <i>length</i> RECORD PREAMBLE = <i>length</i> RECORD STRUCTURE = <i>structure-type</i>	0 — BIN offline=133, online=150 1 0 0 0 0 0 FB
RFEED	RFEED TEST = (<i>test-exp</i> , <i>clu-def</i>)	no default for <i>clu-def</i>
ROFFSET	ROFFSET PASSES = {FIRST ALL} ROFFSET TEST = <i>test-exp</i>	ALL —
<i>ac</i> :ROUTE	<i>ac</i> :ROUTE RFORM = { <i>form-id</i> NONE} <i>ac</i> :ROUTE RTEXT = { <i>sc</i> (<i>sc</i> , [<i>passnum</i> ALL][<i>,line</i> , <i>col</i> [, <i>fontindex</i>]])] <i>rtext-id</i> NONE	NONE NONE
RPAGE	RPAGE SIDE = { <i>side-opt</i> <i>offset-opt</i> } RPAGE TEST = <i>test-exp</i> RPAGE WHEN = {BOTTOM NOW TOP}	NUFRONT, NOFFSET — TOP
RRESUME	RRESUME BEGIN = {CURRENT NEXT} RRESUME TEST = <i>test-exp</i>	NEXT —
RSELECT	RSELECT TEST = <i>test-exp</i>	—
RSTACK	RSTACK ACCTINFO = (<i>offset</i> , <i>length</i>) RSTACK DELIMITER = {YES NO} RSTACK HRPTNA = {(<i>offset</i> , <i>length</i>) NONE} RSTACK PRINT = {BIN BOTH TRAY NONE} RSTACK TEST = <i>test-exp</i>	— NO NONE NONE —
RSUSPEND	RSUSPEND BEGIN = {CURRENT NEXT} RSUSPEND TEST = <i>test-exp</i>	NEXT —
<i>ac</i> :SEFFNT	SEFFMAP = ((<i>font</i> ₁ , <i>font</i> ₂), [(<i>font</i> _{<i>m</i>} , <i>font</i> _{<i>n</i>})])	—
SEFFNT	SEFFMAP = (((<i>font</i> ₁ , <i>font</i> ₂), [(<i>font</i> _{<i>m</i>} , <i>font</i> _{<i>n</i>})] NONE} MAP = <i>file-name</i>	— —

Table B-2. PDL command quick reference (continued)

Command	Syntax	Default
<i>ac</i> :STOCKSET	ASSIGN = { <i>stock-descriptor</i> (<i>stock-descriptor</i> ₁ , <i>stock-descriptor</i> ₂)[,...]} <i>ac</i> :STOCKSET IN FEED = { <i>stock-name</i> <i>stock-reference</i> } <i>ac</i> :STOCKSET SYSPAGE = { <i>stock-name</i> <i>stock-reference</i> MAIN AUX}	— first <i>stock-name</i> —
<i>ac</i> :SYSTEM		—
<i>ac</i> :TABLE	<i>ac</i> :TABLE CONSTANT = (<i>sc</i> ₁ , <i>sc</i> ₂)[,...]) MASK = (<i>ignore-char</i> ([, <i>charspeci</i> ₁ , <i>charspeci</i> ₂)[,...]))	— —
<i>ac</i> :TCODE	DEFAULT = <i>tcode-type</i> TASSIGN = (<i>typespec</i> , {' <i>inputspec</i> ' ₁ (' <i>inputspec</i> ' ₂)[,...])} TRESET = (<i>typespec</i> , {' <i>inputspec</i> ' (' <i>inputspec</i> ' ₁)[,...]) ALL}	— — —
<i>ac</i> :VFU	VFU ASSIGN = {(<i>channo</i> , <i>lineno</i>) <i>channo</i> , <i>lineno</i> [, <i>lineno</i>][,...]} VFU BOF = <i>value</i> VFU TOF = <i>value</i>	— 66 1
VOLUME	VOLUME BMULT = <i>value</i> VOLUME CODE = { <i>keyword</i> <i>id</i> NONE} VOLUME EOVS = {(PAUSE,EOF) (NOPAUSE,NOEOF)} VOLUME HOST = <i>host-type</i> VOLUME INTERPRESS = {([YES] [CHECKSUM NOCHECKSUM] [,BREAKPAGE NOBREAKPAGE] [,ROBUST SIMPLE] [,INTEGRAL] [,PERFORMANCE])}] VOLUME LABEL = <i>label-type</i> VOLUME LCODE = { <i>keyword</i> <i>id</i> } VOLUME LPACK = {YES NO} VOLUME MAXLAB = <i>value</i> VOLUME MINLAB = 80 VOLUME OPTIMIZE = { <i>keyword</i> (<i>keyword</i> [, <i>keyword</i> [, <i>keyword</i>] (,...]) NONE} VOLUME OSCHN = <i>value</i> VOLUME OSHDP = <i>value</i> VOLUME OSTLP = <i>value</i> VOLUME PLABEL = {YES NO} VOLUME RMULT = <i>value</i> VOLUME RSAT = {SPLIT REMOUNT} VOLUME TCODE = { <i>tcode-type</i> <i>tcode-id</i> } VOLUME UNPACK = {T4X3 T4X3H2 UNIVAC NONE}	1 EBCDIC NOPAUSE,NOEOF IBMOS YES, NOCHECKSUM, NOBREAKPAGE, SIMPLE, INTEGRAL, PERFORMANCE, STANDARD EBCDIC NO 81 80 NONE 9 0 0 NO 1 REMOUNT EBCDIC NONE

C. Character code assignment

This appendix illustrates the following character code assignments:

- IBM BCD code set
- Honeywell 200/2000 BCD code set
- Honeywell 6000 BCD code set
- Fieldata translation
- UNIVAC ASCII character set
- Standard ASCII character set
- Standard EBCDIC character set
- Xerox EBCDIC to extended ASCII hexadecimal translation values

IBM BCD code set

Figure C-1 presents the IBM BCD code set.

Figure C-1. **IBM BCD code set**

		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.
^tCorresponds to BCD code set used by IBM users and defined by PDL command CODE = IBMBCD.

Honeywell 200/2000 BCD code set

Figure C-2 presents the Honeywell 200/2000 BCD code set.

Figure C-2. Honeywell 200/2000 BCD code set

		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	+	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.
[†]Corresponds to BCD code set used by Honeywell 200/2000 users and defined by PDL command CODE = H2BCD.

Honeywell 6000 BCD code set

Figure C-3 presents the Honeywell 6000 BCD code set.

Figure C-3. Honeywell 6000 BCD code set

		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	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

^tCorresponds to BCD code set used by Honeywell 600/6000 series SSF tapes and defined by PDL command CODE = BCD or CODE = H6BCD.

Fielddata translation

Figure C-4 presents the Fielddata translation.

Figure C-4. Honeywell 6000 BCD code set

		Most Significant Bits								
		0	1	2	3	4	5	6	7	
Octal	(columns)	000	001	010	011	100	101	110	111	
(rows)	Binary									
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 C-5 presents the UNIVAC ASCII character set.

Figure C-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 C-6 presents the standard ASCII character set.

Figure C-6. **Standard ASCII character set**

		MOST SIGNIFICANT BITS															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Least Significant Bits	Hexadecimal	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	0000			0	@	P	c	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	l	k								
	C	1100		}	,	<	L	\	l								
	D	1101		~	-	=	M]	m								
	E	1110		¼	.	>	N	^	n								
	F	1111		½	/	?	O	_	o								

Standard EBCDIC character set

Figure C-7 presents the standard EBCDIC character set.

Figure C-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								c	i	r	z		l	R	Z	9
	A	1010					c	!	^	:								
	B	1011					.	\$,	#	{	}						
	C	1100					<	*	%	@								
	D	1101					()	_	'			[]				
	E	1110					+	;	>	=								
	F	1111					!	~	>	"								

Xerox EBCDIC to extended ASCII hexadecimal translation values

Figure C-8 presents the Xerox EBCDIC to extended ASCII hexadecimal translation values.

Figure C-8. Xerox EBCDIC to extended ASCII hexadecimal translation values

XEROX EBCDIC TO EXTENDED ASCII HEXADECIMAL TRANSLATION VALUES

VALUES ON PERIMETER ARE EBCDIC. VALUES WITHIN BLOCKED RECTANGLE ARE ASCII. EXAMPLE: EBCDIC VALUE OF AB IS EQUIVALENT TO AN ASCII VALUE OF 7D.

Hexadecimal		EBCDIC MOST SIGNIFICANT DIGIT																
		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	
EBCDIC LEAST SIGNIFICANT DIGIT	0	0000	20	FA	EA	DA	20	26	2D	BO	1E	1F	8C	80	1A	1C	5C	30
	1	0001	20	F9	E9	D9	CA	C1	2F	AF	61	6A	7B	81	41	4A	9F	31
	2	0010	20	F8	E8	D8	C9	CO	B8	AE	62	6B	73	82	42	4B	53	32
	3	0011	20	F7	E7	D7	C8	BF	B7	AD	63	6C	74	83	43	4C	54	33
	4	0100	20	F6	E6	D6	C7	BE	B6	AC	64	6D	75	84	44	4D	55	34
	5	0101	20	F5	E5	D5	C6	BD	B5	AB	65	6E	76	85	45	4E	56	35
	6	0110	20	F4	E4	D4	C5	BC	B4	19	66	6F	77	86	46	4F	57	36
	7	0111	20	F3	E3	D3	C4	BB	B3	18	67	70	78	87	47	50	58	37
	8	1000	20	F2	E2	D2	C3	BA	B2	AA	68	71	79	88	48	51	59	38
	9	1001	20	F1	E1	D1	C2	B9	B1	60	69	72	7A	89	49	52	5A	39
	A	1010	20	FO	EO	DO	60	21	5E	3A	A9	90	91	A8	A7	A3	9E	1B
	B	1011	FF	EF	DF	CF	2E	24	2C	23	1A	1C	7D	17	A6	A2	9D	99
	C	1100	FE	EE	DE	CE	3C	2A	25	40	10	12	7E	7F	13	8F	15	98
	D	1101	FD	ED	DD	CD	28	29	5F	27	8A	8D	5B	5D	A5	8E	9C	97
	E	1110	FC	EC	DC	CC	2B	3B	3E	3D	BB	7C	11	95	14	A1	9B	96
	F	1111	FB	EB	DB	CB	1B	1D	3F	22	94	93	92	16	A4	A0	9A	20

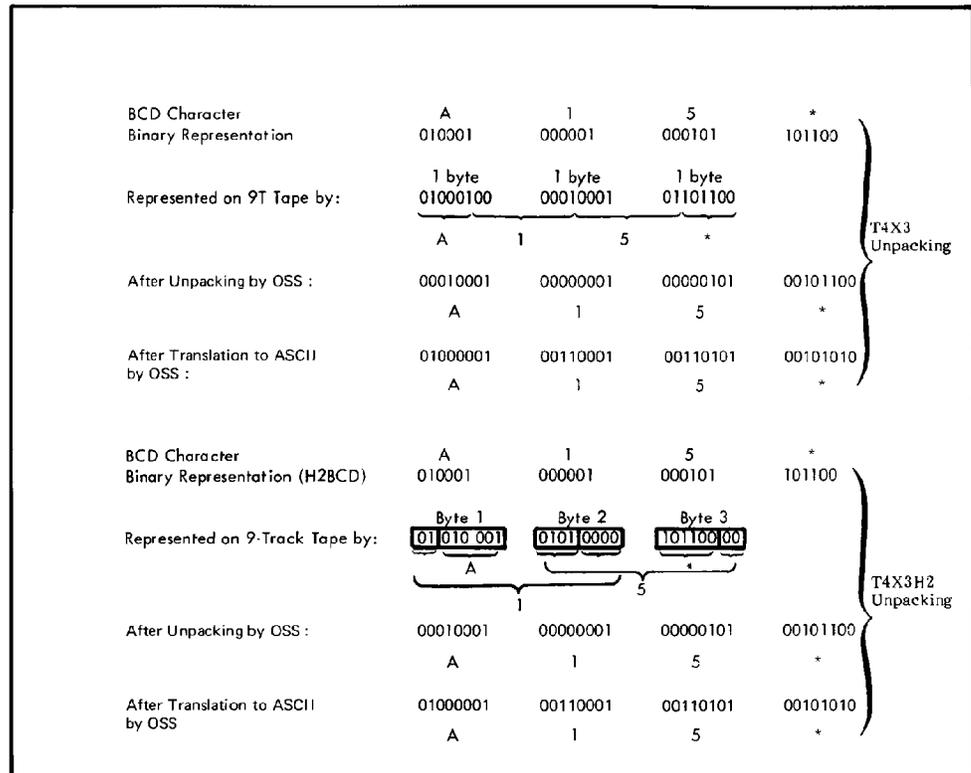
D. Offline specifications

This appendix provides offline specifications.

Input unpacking examples

Figure D-1 shows several input unpacking examples.

Figure D-1. Input unpacking examples



Valid host computer and label specifications

Figure D-2 shows the valid host computer and label specifications.

Figure D-2. Valid host computer and label specifications

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 B2700 medium B3500 systems	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

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 parameter for each type of host.

Host system JDLs on system software tape

Job description library (JDL) source files supplied on operating system software (OSS) tape are summarized in table D-1.

Table D-1. JDL source files on OSS tape

JDL	JDEs provided for																
BUR	Burroughs medium system (B2500, B2700, B3500, B3700, and B4700) Burroughs large systems-labeled printer backup and standard ANSI-labeled tapes (B6700 and B7700)																
DUMP or DUMPA4	Tape dumps of various tape formats where: <table border="1"> <thead> <tr> <th>JDE</th> <th>Characteristics</th> </tr> </thead> <tbody> <tr> <td>1-or-H6BCD</td> <td>H6000 BCD</td> </tr> <tr> <td>2-or-EBCDIC</td> <td>EBCDIC</td> </tr> <tr> <td>3-or-ASCII</td> <td>ASCII</td> </tr> <tr> <td>4-or-H2BCD</td> <td>H2000 BCD</td> </tr> <tr> <td>5-or-IBMBCD</td> <td>IBM BCD</td> </tr> <tr> <td>6-or-UNIVAC</td> <td>UNIVAC Fieldata</td> </tr> <tr> <td>7-or-ICL</td> <td>ICL 2900 VME/B</td> </tr> </tbody> </table>	JDE	Characteristics	1-or-H6BCD	H6000 BCD	2-or-EBCDIC	EBCDIC	3-or-ASCII	ASCII	4-or-H2BCD	H2000 BCD	5-or-IBMBCD	IBM BCD	6-or-UNIVAC	UNIVAC Fieldata	7-or-ICL	ICL 2900 VME/B
JDE	Characteristics																
1-or-H6BCD	H6000 BCD																
2-or-EBCDIC	EBCDIC																
3-or-ASCII	ASCII																
4-or-H2BCD	H2000 BCD																
5-or-IBMBCD	IBM BCD																
6-or-UNIVAC	UNIVAC Fieldata																
7-or-ICL	ICL 2900 VME/B																
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 or without banner or trailer pages																
POWGRP	IBM POWER VS, POWER 4.0 POWER 4.1 or 4.2; IBM DOS GRASP (including Interpress tapes)																
RSX	RSX-11 (FLX) ASCII tapes (including Interpress tapes)																
UNIVAC	Univac DF-formated tapes																
XEROX	ANSI-labeled tapes Unlabeled tapes Xerox ANSI-labeled tapes JDEs to print unknown tapes Printing tapes without attempting to deblock the tape properly																

After an LPS software system is created by the user, the source for these JDLs is resident in the JSL directory. Use the Editor PRINT parameter or the PDL compiler to obtain a printout of the JDL(s) desired.

Interpress data from magnetic tape

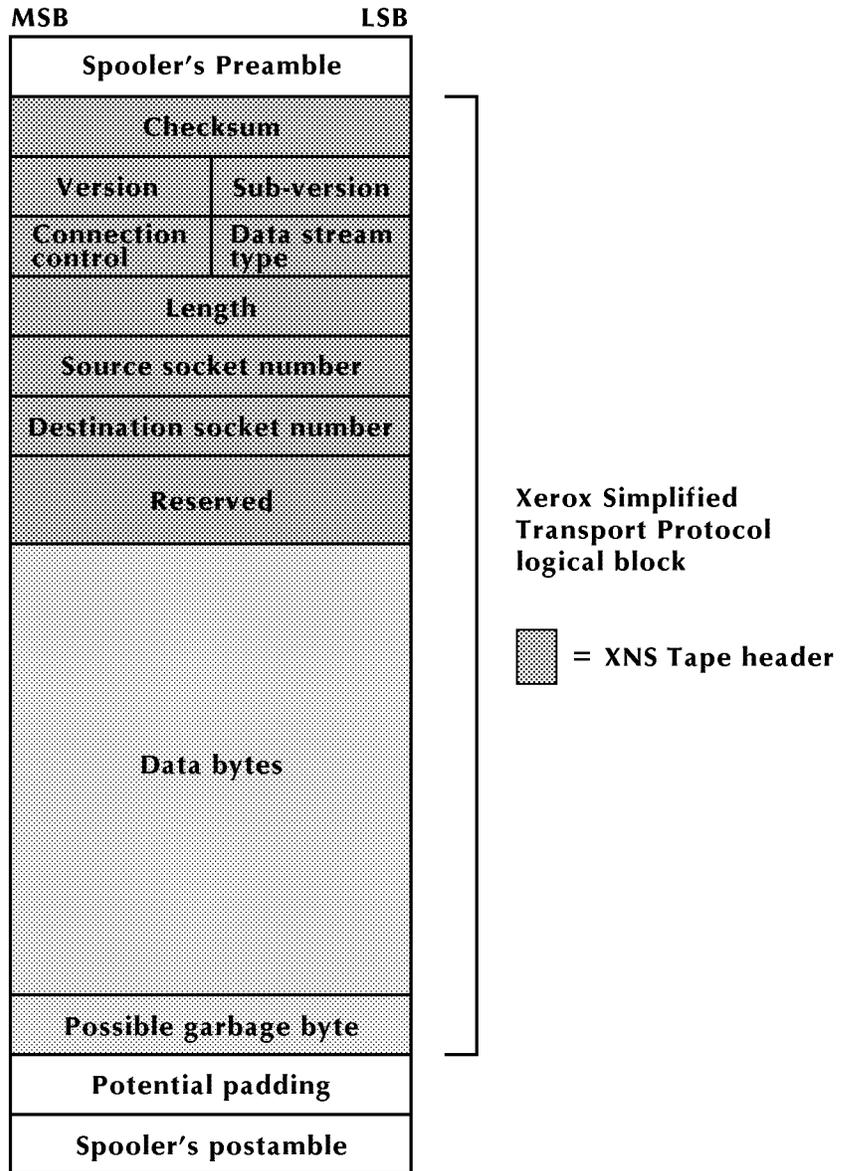
A limited set of Xerox Network Systems (XNS) transactions is supported for magnetic tape input to receive Interpress data from tape. Supported transactions are:

- Courier-encoded routing packets
- Courier-encoded procedure call message packets. (The procedure call supported is "Xerox NS version 3.0 Printing.Print.")
- Bulk data packets. The bulk data is Interpress Commercial set 3.0 encoded data. Interpress Publication set is supported. XNS protocols are described further in the *LPS Operations Reference*.

Tape format

Figure D-3 shows the tape format.

Figure D-3. **Tape format**



Interpress tape input must be formatted as follows:

Information is encapsulated on one tape in logical blocks which consist of a header (described in detail in the Xerox Simple Transport Protocol Standard) followed by data. A logical block is defined here as equivalent to one user record.

The header consists of:

- 2 bytes - Checksum (per the Internet Datagram Protocol)
- 2 bytes - Protocol version number (per the Sequenced Packet Protocol)

- 1 byte - Connection control (per the Sequenced Packet Protocol)
- 1 byte - Data stream type (per the Sequenced Packet Protocol)
- 2 bytes - Length (per the Internet Datagram Protocol)
- 2 bytes - Source socket number (per the Internet Datagram Protocol)
- 2 bytes - Destination socket number (per the Internet Datagram Protocol)
- 4 bytes - Reserve for future use

Points to note

Version 0.1, (version 0 in the high-order byte; subversion 1, in the low order byte) is the only version currently supported.

Currently, the end-of-message bit is defined for the connection control field. It is set in the header of any logical block containing the last byte of a data stream. Bit 3 (10 hexadecimal) is the assigned bit for the End-of-Message flag, as defined by the Sequenced Packet Protocol.

The four data stream type values assigned are as follows:

0 = Courier-encoded print message

1 = Bulk data, such as Interpress

2 = Courier-encoded routing data

254 = End

The data stream types and the connection control fields are planned to be compatible with the Sequenced Packet Protocol.

The logical block must be an integral number of 16-bit words, while the data contained in it is not so constrained, implying that there may be a garbage byte at the end of the logical block. This garbage byte is not included in the length field but is included in the checksum.

Instead of using addresses as defined in the Internet Datagram Protocol, routing information (destination and source) is represented by Courier-encoded clearinghouse three-part names, that is, Name:Domain:Organization. The Courier encoding of a clearinghouse name is a RECORD of three STRINGs. If the names are unknown/nonexistent, they are represented by strings of zero (0) length.

A file is defined as all the information between the start-of-file and end-of-file label groups. If the tape format supports it, a file may span volumes. A file fully contains one or more documents. (A document may not span file boundaries.) Information is stored on a file in the following order: 1 routing logical block 1 or more Courier-encoded print procedure message logical blocks 0 or more bulk logical blocks Optional repetitions of b- followed by c-type logical blocks An end logical block.

The recipient may choose not to verify the checksum as the default. However, some means must be provided for enabling checksum verification if the sender specifically requests the service. The presence of valid checksums (as opposed to the null value, all bits, that is, FFFF hexadecimal) does not guarantee that checksums are verified by the recipient.

Usage requirements

To input Interpress data through magnetic tape:

- The tape's format must comply with the Xerox Magnetic Tape Format Standard and the Xerox Simple Transport Protocol Standard.
- Interpress must be specified from magnetic tape in the job START parameter. The START command has the following fields:

```
START [jde],[jdl],[S or M],[copies],[REPORTS:r1,r2,...],INTERPRESS
```

(Note that the keyword INTERPRESS must be included; fields in brackets are optional.)
- Interpress must be specified in the job descriptor library (JDL) by using the Print Description Language (PDL) parameter command:

```
VOLUME INTERPRESS=YES.
```
- The parameter to enable error checker for Interpress tapes is provided, on a job basis, through the PDL parameter command:

```
VOLUME INTERPRESS=CHECKSUM (or NOCHECKSUM).
```

(Note that this software checking is in addition to any other error checking mechanism supplied by the tape driver's hardware.)
- An appropriate JDL must be set up so that the data extraction processor can skip over the host-specific printer carriage control bytes and other control bytes normally associated with the line printer, record-orientated data. (Refer to the *Xerox LPS Tape Formats Manual* for details.)

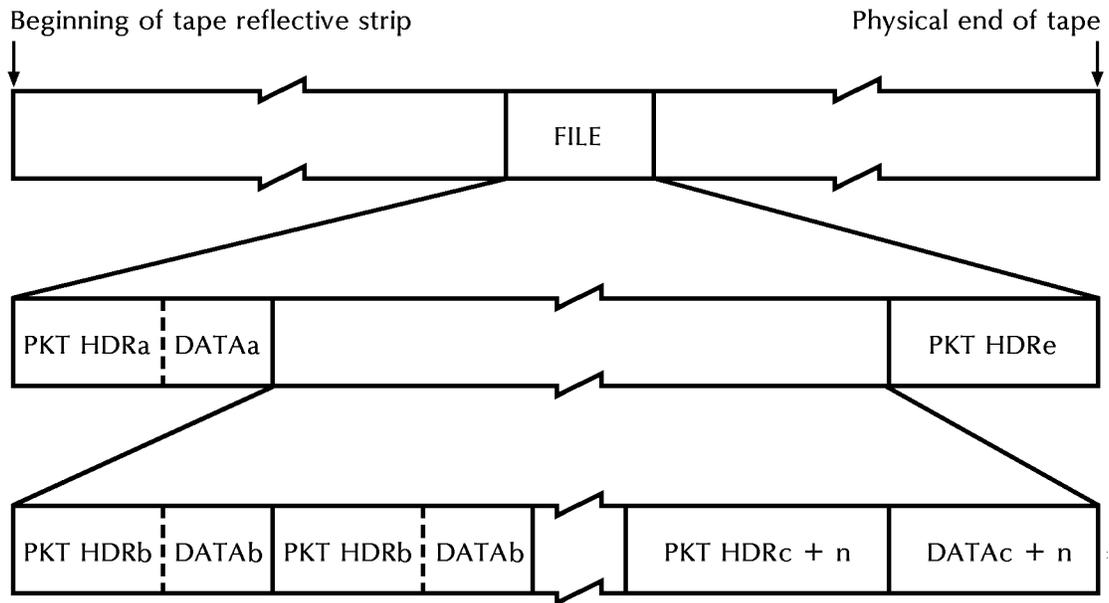
Notes and restrictions:

- A logical block may not span a physical block, and a physical block may not contain more than one logical block. A logical block is defined as equivalent to one user data record. (Refer to "Xerox Simplified Transport Protocol logical block" example earlier in this appendix.)
- The data portion of the logical block is restricted to a minimum length of 1024 bytes. The restrictions are placed to optimize performance. The restriction on the maximum logical block length is placed so the Interpress processing does not have a degrading impact on the available user space in input. It is best to create logical blocks of precisely 1040 bytes in length. This block length is the sum of 16 XNS control bytes and 1024 data bytes. When the length of the block is less than the minimum physical block length, padding bytes of unspecified value are appended to make up a physical block of the appropriate length. These padding bytes are discarded by the recipient of the system.
- All routing packet information is ignored by the LPS.
- The job is aborted when the checksum verification is enabled but fails.
- The PDL parameter command, VOLUME PLABEL=YES or NO, is not supported for Interpress tapes. This parameter specifies if the tape labels are to be printed with the job.

Sample layout of data (Interpress)

Figure D-5 represents the layout of data on tape for a simple print request of one copy of the accompanying Interpress master.

Figure D-5. **PKT HDR (Packet Header)**



The dashed delimiters in the diagram above represent the boundaries within the logical blocks. The solid delimiters represent any required physical block padding and the inter record gap.

Tables D-2 through D-12 explain the diagram further.

Table D-2. **PKT HDRa (Address logical block header**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
1002	Connection control (last packet of this data stream type = 10 hexadecimal, in high byte); data stream type in (logical address = 2, in low byte)
0014	Length of packet + data, in bytes
0000	Source socket (zero = unknown)
0005	Destination socket (socket # for Courier = 5)
0000 0000	Reserved for future use

Table D-3. **DATAa (Address logical block data, Courier encoded)**

Hexadecimal value	Meaning
0000	Source address (Courier-encoded string of zero length = no source address)
0000	Destination address (Courier-encoded string of zero length = no destination address)

Table D-4. **PKT HDRb (Service request logical block header)**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
1000	Connection control (last packet of this data stream type = 10 hexadecimal, in high byte); data stream type (Courier =0, in low byte)
0022	Length of packet + data, in bytes
0000	Source socket (zero = unknown)
0005	Destination socket (socket # for Courier = 5)
0000 0000	Reserved for future use

Table D-5. **DATAb (Service request logical block data, Courier encoded)**

Hexadecimal value	Meaning
0000 0000 0000 0004 0003 0000 0001 0000 0000	Courier encoding of a Printing.Print, version 3 procedure call

Table D-6. **PKT HDRc (Bulk data logical block header)**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
0001	Connection control (not the last packet of this data stream type = zero, in the high byte); data stream type (bulk data = 1, in low byte)
0410	Length of packet + data, in bytes; first block in Interpress master
0000	Source socket (zero = unknown)
0005	Destination socket (socket # for Courier = 5)
0000 0000	Reserved for future use

Table D-7. **PKT HDRc (Bulk data logical block data)**

Hexadecimal value	Meaning
496E 7465 7270 7265 7373 2F58 6572 6F78 2F33 2E30 20...	Interpress/Xerox/3.0 ... rest of data for first block of Interpress master

Table D-8. **PKT HDRc + 1 (Bulk data logical block header)**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
0001	Connection control (not the last packet of this data stream type = zero, in high byte); data stream type (bulk data = 1, in low byte)
0410	Length of packet + data, in bytes; next block of Interpress master
0000	Source socket (zero = unknown)
0005	Destination socket (socket # for Courier = 5)
0000 0000	Reserved for future use

Table D-9. **DATAc + 1 (Bulk data logical block data)**

Hexadecimal value	Meaning
...	>more of Interpress master

This logical block is followed by as many additional logical blocks as needed to contain the master. The last bulk data logical block has the following structure:

Table D-10. **PKT HDRc + n (Bulk data logical block header)**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
1001	Connection control (last packet of this data stream type = 10, in high byte); data stream type (bulk data = 1, in low byte)
nnnn	Length of packet + data, in bytes; last block of Interpress master, if length is odd, a garbage byte is appended to data
0000	Source socket (zero = unknown)
0005	Destination socket (socket # for Courier = 5)
0000 0000	Reserved for future use

Table D-11. **Data +n (Bulk data in logical block data)**

Hexadecimal value	Meaning
...	...end of Interpress master

Table D-12. **PKT HDRe (End of logical block header; there is no data associated with this logical block)**

Hexadecimal value	Meaning
FFFF	Checksum (Don't Check = FFFF hexadecimal)
0001	Protocol version # = 0.1 (zero in high byte; 1 in low byte)
10FE	Connection control (last packet of this data stream type = 10, in high byte); data stream type (End = 254, in low byte)
0010	Length of packet + data, in bytes
0000	Source socket (zero = unknown)
0005	Destination socket (socket for # for Courier = 5)
0000 0000	Reserved for future use

E. Editor quick reference

This appendix summarizes the basic Editor commands and steps for creating and modifying JSLs. For detailed information on using the Editor, refer to your *Xerox DocuPrint 96/DocuPrint 96MX LPS Operations Reference*.

Note that you may enter the Editor commands by keying in only the first three letters, for example, CLEAR can be entered as CLE, and INSERT can be entered as INS.

Creating a new file on system disk

When you have prepared your JSL and are ready to enter it onto system disk, you must use the Editor. To use the Editor for coding your JSL, follow these steps:

- Step 1.** Gain access to the file by entering the Editor. Enter:
LOG 3,password
- Step 2.** Begin the editing session by entering:
EDIT
- Step 3.** Clear the work file of all data, enter:
CLEAR
- Step 4.** Verify the new file name you have selected and protect existing files by entering:
GET file-id

The *file-id* consists of the file name and file type, represented as *file-name.file-type*, for example XAMPLE.JSL.

A message appears on the system controller display if a file already exists with the *file-id* you have selected. To protect the original file, clear the work file, rename your file, and repeat the GET *file-id* command.

Step 5. Number the lines

Automatically by entering:

INSERT *n,y*

n refers to the starting line, also called record number. *y* refers to increments at which the line numbers should appear. For example, the command INSERT 10,10 means to insert lines starting with 10 and increase in increments of 10. The first record then would be 00010, with subsequent records numbered 00020, 00030, 00040, and so on.

Manually by entering the number you wish to assign the line you are specifying. This must be done for every line you enter, whether it is a full command line, a continuation of a previous command line, or a comment line.

For example, if you wish to add line 00011, following 00010, you would simply enter 11.

Step 6. Input the text, your JSL coding, by entering the text after the line number which appears on the system controller display.

For example, the display would show:

```
EDIT>INSERT 10,10  
00010.
```

You would then enter your text following 00010:

```
00010      XAMPLE:JDL;
```

The display would then provide the next record, 00020 for you to enter another line of text.

Step 7. Save the text file you have created on system disk by entering:**SAVE *file-id***

Remember that *file-id* consists of *file-name.file-type*.

Step 8. Terminate the editing session.

To end the editing session without printing the work file, simply enter

END

To end the editing session and print the work file, enter:

PCE

PCE represents PRINT, CLEAR, and END. This provides a print version of your source, clear the workfile, and end the editing session in one command.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

- Step 1.** Gain access to the file by entering:
LOGON 3, *password*
- Step 2.** Begin the editing session by entering:
EDIT
- Step 3.** Clear the work file of all data by entering:
CLEAR
- Step 4.** Retrieve the file you want by entering:
GET *file-name.file-type*

Displaying the text of a file

If you wish to see the lines of source code in the work file, you can use any of the following options:

- Display the entire work file by entering

DISPLAY

- Display a particular portion of the work file by entering:

DISPLAY *n-y*

n is the starting line number to be displayed.

y is the ending line number to be displayed.

- Display a single line, or record, of the work file by entering:

DISPLAY *n*

n is the desired line.

- Display a particular word or phrase within a work file wherever it exists by entering:

FIND/*x*

x is the word or phrase to be found and displayed.

- Display the starting and ending line numbers of the work file if you wish to know on which lines the work file is located by entering:

KEYS

The system provides these line numbers, as in this example:

```
EDIT KEYS
BEGINNING LINE NUMBER 00010
ENDING LINE NUMER 00190
EDIT.
```

- Print a copy of the work file by entering:

PCE

This command is an abbreviated form of the PRINT, CLEAR, and END commands.

Be sure to save the work file before using the PCE command by entering:

SAVE *file-id*

The *file-id* consists of the file name and *file* type, represented as *file-name.file-type*.

Modifying the text in a file

Use the following procedures to change lines of the source code in your program:

- Step 1.** Modify a portion of the work file line
or
Modify entire lines of the work file.
- Step 2.** Save your source code file.
- Step 3.** Terminate the editing session.

Modifying a portion of the work file line

To modify portions of a line, you must first enter:

MODIFY *n*

n is the number of the line to be modified, for example, MODIFY 60.

Then you can perform any of the following activities:

- Substituting one set of characters for another by entering:

S/x/y* or *C / x / y /

x is the text to be replaced.

y is the text to be inserted.

or

S [k]/x/y/* or *C [k]/x/y/

k is the occurrence of string *x* in each line specified in the MODIFY command that is affected by the S or C command. *k* defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string *x*, in the specified lines are affected.

Add a space between S or C and *k* in the command.

- Overlay one set of characters over another by entering:

O/x/y/

x is the text to be replaced.

y is the new text.

or

O [k]/x/y/

k is the occurrence of string *x* in each line specified in the MODIFY command that is affected by the OVERLAY command. *k* defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string *x*, in the specified lines are affected.

Add a space between O and *k* in the command.

When O is used to replace characters that are not at the end of the line, be sure that the characters and spaces inserted equal those being replaced.

- Insert a set of characters before other characters by entering:

P/x/y/

x is the word following the text to be added.

y is the text to be added.

or

P [k]/x/y/

k is the occurrence of string *x* in each line specified in the MODIFY command that will be affected by the P command. *k* defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string *x*, in the specified lines are affected.

Add a space between P and *k* in the command.

- Insert a set of characters following other characters by entering:

F/x/y/

x is the word preceding the text to be added.

y is the text to be added.

or

F [k]/x/y/

k is the occurrence of string *x* in each line specified in the MODIFY command that is affected by the F command. *k* defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string *x*, in the specified lines are affected.

Add a space between F and k in the command.

Modifying entire lines

To change entire lines, or records, or your JSL coding, you may perform any of the following activities:

- Move lines in a file by entering:

MOVE *n-m,p,i*

n is the starting line number.

m is the ending line number.

p is the new starting line number.

i is the increment of line numbering you want.

For example, MOVE 80-11-,51,1 results in the following:

Old	New
00080	00051
00090	00052
00100	00053
00110	00054

- Remove lines from a file by entering:

REMOVE *n*

or just enter the line number.

n is the line to be removed.

- Duplicate lines in a file by entering:

DUPLICATE *n-m,p,i*

n is the starting line number.

m is the ending line number.

p is the new starting line number.

i is the increment of line numbering you want.

- Replace lines with new lines by entering:

REPLACE *n-m,i*

n is the starting line number.

m is the ending line number.

i is the increment of line numbering you want.

This command deletes all lines from *n* to *m*. It initiates automatic line numbering at *n* for you to enter code and increments each line by your value for *i*.

- Renumber lines of text by entering:

RENUMBER *n,i*

n is the starting line number.

i is the increment of line numbering you want.

After modifying a file, you may find significant gaps in the sequence of line numbers. The RENUMBER command allows you to renumber the lines by establishing your starting line number *n* and incrementing value *i*.

- Merge one file with another by entering:

MERGE *file-id,n,i*

file-id consists of filename.file-type and specifies the file to be merged into the one you are currently working on.

n is the starting line where the merged file is to start.

i is the line number increment you wish to use.

Saving your source code file

After you have finished editing your file, save it by entering:

SAVE *file-id*

file-id consists of *file-name.file-type*.

Terminating the editing session

To end the editing session without printing the work file, enter:

END

To end the editing session and print out the work file, enter:

PCE

PCE represents the commands to PRINT the work file, CLEAR the work file, and END the editing session.

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.
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. 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.
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 DocuPrint 96/DocuPrint 96MX 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 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.
MICR	Magnetic Ink Character Recognition. MICR allows a computer to read characters printed in special metallic ink by scanning the shapes of their magnetic fields. MICR fields are often imprinted on bank checks before processing.
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.
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, 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.
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 4135 LPS operates in a multi-pitch mode, allowing the mode to be adjusted according to the width of the paper used.
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 4135 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).
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.
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")
MICR	Magnetic Ink Character Recognition
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
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
XNS	Xerox Network Systems
XPAF, XPF	Xerox Printer Access Facility
XPMF-VMS	Xerox Print Management Facility - VMS Version
XPPI	Xerox Pen Plotter Interface
XPS	Xerox Publishing System

Numerals

- 3211/4245 mode, 4-58
- 4850/4890 HighLight Color LPS
 - applications, 3-29 to 3-31
 - forms, 3-30 to 3-31
- 4050/4090/4650/9700F LPS,
 - creating applications, 3-31
 - running jobs, 3-31

A

- ABNORMAL command, 6-2 to 6-10
- ABORT parameters, 6-3
- ac:CME command, 6-18 to 6-24
- accounting
 - data, 6-11
 - installation reports, 6-13 to 6-15
 - usage, overall, 6-12
- ACCT command parameters
 - specifying
 - JSL, 3-12
 - print format, 6-11 to 6-17
- ACCTINFO parameter, 7-37
- ADJUST parameter, 4-2
- Advanced Image Subsystem (AIS), 1-3
- ALTER parameters, 8-20
- applications
 - 4850/4890, running, 3-29 to 3-30
 - 4050/4090/4650/9700F
 - creating, 3-31
 - running job, 3-31
 - mixing, 5-5
 - stockset, 5-5
 - types, 3-4
- ASCII constants, 2-7
- ASSIGN parameters
 - CODE, 4-9
 - DJDE, 8-21
 - STOCKSET, 6-100
 - VFU, 6-103

B

- banner page detection, 4-61
- BANNER command, 7-13 to 7-17
- BATCH parameter, 8-20
- BDELETE command, 7-18 to 7-19

- BEGIN parameter
 - DJDE, 8-22
 - PDE, 6-89
 - RRESUME, 7-42
 - RSUSPEND, 7-42
- BFORM parameter
 - DJDE, 8-23
 - OUTPUT, 6-50
- BLOCK command
 - examples, 4-8
 - parameters
 - ADJUST, 4-2
 - CONSTANT, 4-3
 - FORMAT, 4-3
 - LENGTH, 4-4
 - LMULT, 4-4
 - LTHFLD, 4-5
 - OFFSET, 4-5
 - POSTAMBLE, 4-6
 - PREAMBLE, 4-6
 - ZERO, 4-7
 - points to note, 4-7 to 4-8
- BOF parameter
 - DJDE command, 8-24 to 8-25
 - VFU command, 6-103
- Bottom-Of-Form, *see* BOF
- BSELECT command, 7-18 to 7-19
- buffer, page, 1-7

C

- C parameter, 8-25
- capabilities, 1-1
- card-image files, 8-66 to 8-67
- change mode criteria, 7-4
- channel-attached LPS, 4-58
- character
 - see also* fonts
 - constants, 2-7
 - dispatching, 1-3
 - spacing, 3-26
 - strings, 6-19
 - types, 7-10
- characteristics, defining, 6-32
- checkpoint data, 6-9
- clusters
 - checking, 5-5
 - creating, 5-8 to 5-9
 - defining, 5-6 to 5-10

- features, 5-1
- processing overview, 5-1
- specifying, 5-4 to 5-5
- storing
 - MAIN, 5-3
 - AUX, 5-3
 - AUTO, 5-3
- usage, 5-4
- CMEs
 - cataloged, 6-19
 - specific commands, online, 4-59
 - specifying, 6-18
 - using, 3-14
- CODE command
 - ASSIGN parameter, 4-9
 - DEFAULT parameter, 4-10
- coded instructions, see programs
- coding
 - features, 2-16
 - process, 2-16
- COLLATE parameter
 - DJDE, 8-26
 - OUTPUT, 6-48
- columns, creating JSL, 3-9
- combining modes, 7-8
- commands
 - END, 2-13
 - IDEN parameter, 1-6
 - JOB, 2-13
 - keyword, 2-3
 - levels
 - catalog, 2-12 to 2-13
 - JDE, 2-12
 - JDL, 2-12
 - JSL, 2-12 to 2-13
 - PDL, 3-6
- communication operator subsystem, 1-5
- comparing strings, 7-10
- compiling
 - files, 3-2
 - JSL, 3-18 to 3-19
 - PDL, 1-6
- components
 - command
 - comments, 2-2 to 2-4
 - identifier, 2-3
 - keyword, 2-3
 - parameter, 2-4
 - hardware, AIS, 1-3
 - LPS hardware, 1-3
 - PDL
 - identifier, 3-6
 - keyword, 3-7
 - parameters, 3-7
 - physical, 1-2
 - software, 1-4

- CONSTANT
 - mode criteria, 7-7
 - parameters
 - ac:CME, 6-19
 - BLOCK, 4-3
 - TABLE, 7-45
 - constants
 - right-part, 2-6
 - string
 - ASCII, 2-7
 - character, 2-7
 - EBCDIC, 2-8
 - H2 and H6, 2-8
 - hexadecimal, 2-7
 - keyword, 2-6
 - octal, 2-8
 - variable name, 2-6
- CONTINUE parameter, 6-3
- COPIES parameter
 - DJDE definitions, 8-26
 - OUTPUT command, 6-51
- COPIES=processing, 8-17
- Copy Modification Entries, see CMEs
- COVER parameter options, 6-52
- creating
 - application types, 3-4
 - files, 2-16
 - job source library
 - columns, 3-9
 - editor, 3-11
 - identifier, naming, 3-11
 - requirements, 3-9
 - specifying, 3-10
 - VFU, 3-11
- CRITERIA
 - commands, 7-3 to 7-9
 - identifiers, 7-3
- CYCLEFORMS parameter, 6-53

D

- data
 - form, 1-11
 - print line, 6-33
 - processing
 - CME, 1-12
 - DJDE, 1-12
 - JDE, 1-12
 - raw, 1-11
 - unformatted, 1-11
 - variable, 1-11
- DATA parameters
 - DJDE command, 8-28
 - LINE command, 6-33
- DEFAULT parameter, 4-10

defining
 characteristics, 6-32
 clusters, 5-6 to 5-10
 paper requirements, 3-14
 system responses, 6-3

deleting specialized records, 7-24

delimiter
 accounting page, 7-36
 display, 7-36
 modes, 7-35
 parameters, 7-38

DENSITY parameter, 6-54

DEPT parameter, 6-16

DESTINATION parameter
 DJDE command, 8-27
 JSL command, 3-14
 OUTPUT command, 6-54

disk
 access, separate, 9-9
 creating system, E-1 to E-9
 stored files, E-3

display, status, 7-36

DJDEs
 application, 8-12 to 8-13
 file
 effectiveness, 8-65
 parameter, 8-64 to 8-97
 replacement, 8-65
 graphics, options, 9-8
 online restrictions, 8-19
 operator information, 8-14
 page-oriented, 8-3 to 8-5
 parameter definitions, 8-20 to 8-63
 processing
 optimization, 8-19
 specific commands online, 4-59
 record-oriented, 8-6
 record specification, 8-10 to 8-11
 using, 8-2

document and page interleaves, 9-6

document interleaved file transfers, 9-7

downloading 4850/4890 LPS applications, 3-30

dump, online, 4-66

DUPLEX parameter
 DJDE command, 8-29
 OUTPUT command, 6-55

duplex printing
 DJDE, 8-17
 RAUX command, 7-20

Dynamic Job Descriptor task, *see* DJD

Dynamic Job Descriptor Entries, *see* DJDEs

E

EBCDIC constants, 2-8
 editing sessions, terminating, E-5

editor task, 1-6

End-of-report, 7-13

END command
 DJDE parameter, 8-29
 JSL, creating, 3-15

entry lines, modifying, E-5 to E-6

ERROR parameter, 6-3

EXPORT command, 6-25 to 6-31

F

FACEUP parameter, 6-55

FCB parameter
 LINE command, 6-33
 online printing system, 4-58

FDL
 compiler, 1-7
 processing, 1-7

features and functions, PDL, 1-1

FEED parameter
 DJDE command, 8-30
 EXPORT command, 6-56

fieldata translation, C-4

file control program, *see* FCP

files
 compiling, 3-2
 creating, 2-16
 document interleaved, 9-9
 downloading, 4-68
 parameters, 8-31

FONTINDEX parameter
 DJDE, 8-32
 LINE command, 6-34

fonts
 memory, 3-27
 parameter
 ac:CME command, 6-18
 DJDE parameter, 8-33
 PDE command, 6-89
 spacing, 3-26 to 3-27

FORMAT parameters
 BLOCK command, 4-3
 DJDE parameter, 8-34
 OUTPUT command, 6-57

formatted data, 1-12

Forms Control Buffer, *see* FCB

FORMS
 definition command, 1-11
 parameter
 DJDE, 8-35
 OUTPUT command, 6-58

Forms Description Language, *see* FDL

Form Source Language, *see* FSL

G

graphics

- considerations, 9-1
- DJDE options, 9-8
- feature restrictions, 9-10
- handling modes
 - batch, 9-3
 - block, 9-3
 - move, 9-2
 - random, 9-2
- inputting, 9-2
- memory, 1-3
- online, 9-9
- parameters
 - DJDE, 8-37 to 8-38
 - OUTPUT command, 6-59
- PDL command options, 9-8
- performance, 9-9
- position, 6-72
- random mode, 9-9
- restrictions, 9-10

grouping PDL commands, 2-16

H

H2 and H6 constants, 2-8

hardware

- components, 1-2
- firmware, 1-2

HCOUNT parameter, 7-14

hexadecimal constants, 2-7

hierarchy of replacement, 2-16 to 2-21

HJOBNO parameter, 7-14

Honeywell

- 200/2000 BCD code set, C-2
- 6000 BCD code set, C-3

HOSTCOPY, 9-6

HRPTNA parameter

- BANNER command, 7-15
- RSTACK, 7-38

I

IBM BCD code set, C-1

ID level

- catalog level, 2-12 to 2-13
- commands
 - JDE, 2-13
 - job, 2-13
- END command, 2-13 to 2-15
- JDL, 2-12
- system, 2-12

IDEN command parameter, 8-7

identifier commands, 2-3

IMAGE

see also system page

data, 6-70

parameters

- DJDE command, 8-38 to 8-39
- OUTPUT command, 6-60

shift, 6-67

IMISMATCH parameter, 6-3

information, tracking, 1-6

INIFEED parameter, 6-101

input data

characteristics, 3-12

parameters, setting up, 3-12

processing commands

- BLOCK, 4-2 to 4-8
- CODE, 4-9 to 4-11
- PCC, 4-12 to 4-18
- RECORD, 4-19 to 4-26
- SEFFNT, 4-27 to 4-31
- TCODE, 4-32 to 4-37
- VOLUME, 4-38 to 4-57

processing tasks, 1-5

records, 6-33

sources, 4-58

streams, 1-7

types

- formatted, 1-11 to 1-12
- unformatted, 1-11 to 1-12

input unpacking examples, D-1

INVERT parameter

DJDE, 8-39

OUTPUT commands, 6-61

ISUBSTITUTE parameter, 6-6

ITEXT parameter, 8-40

J

JDE

- creating, 4-59
- parameters, 8-40

JDL

- creating specific commands online, 4-59 to 4-60
- parameters, 8-41
- system software tapes, D-4

JOB command, 2-13

job control block, 1-6

Job Descriptor Entry, *see* JDE

Job Descriptor Library, *see* JDL

job flow process, 1-9 to 1-10

job parameter restrictions, 8-15 to 8-16

Job Queue Entry, *see* JQE

Job Source Library, *see* JSL

JQE parameter, 1-9

JSL

- columns, 3-9
- creating, 3-11 to 3-17
- editor, 3-11
- identifier, naming, 3-11
- requirements, 3-9
- specifying, 3-10
- structure, 2-10 to 2-15
- VFU, 3-11

K

keyword constant, 2-6

L

Laser Printing System, *see* LPS

LENGTH parameters, 4-4

levels

- Command, 2-10 to 2-11
- ID
 - catalog, 2-12
 - END, 2-13 to 2-15
 - JDE, 2-13
 - JDL, 2-12
 - JOB, 2-13
 - system, 2-12

LINE command

- ac:CME, 6-21
- parameters, 6-32 to 6-43
- references, 3-12

LINENUM parameters, 7-5

list parameter options, 2-5

LMULT parameters, 4-4

log, page, 1-7

logical

- pages, multiple, 6-91
- processing
 - commands, 7-1 to 7-2
 - formatting, 7-2
- TEST parameters, 7-2

LPS components

- channel attached, 4-58
- hardware, 1-2
- operating system software (OSS), 1-4 to 1-5
- processing, 3-2
- software, 1-2, 1-4 to 1-5
- types, 1-2 to 1-4

LTHFLD parameter, 4-5

M

magnetic tapes

- formatting, D-6 to D-7
- Interpress data, D-5 to D-14
- usage requirements, D-8 to D-9

MAP parameter, 8-41

MARGIN parameter

- DJDE command, 8-42
- LINE command, 6-35

marker pages, 6-9 to 6-10

masked comparisons using

- default type, 7-11
- non-default type, 7-12

memory

- font, 3-27
- graphic, 3-27

MESSAGE command, 6-44 to 6-46

mixing applications, 5-5

modes

- combining, 7-8
- delimiter, 7-35
- non-delimiter, 7-35
- non-stop, 6-3

MODIFY

- parameter
 - DJDE command, 8-43
 - OUTPUT command, 6-62
- text files, E-5

multiple logical pages

- DJDE command, 8-18
- PDE command, 6-90

multivolume processing, 4-72

N

NTO 1 parameter, 6-63

NUMBER parameter

- DJDE command, 8-44
- OUTPUT command, 6-64

O

OCS

- processing, 1-9
- task, 1-6

octal constants, 2-8

OFFSET parameters

- BLOCK command, 4-5
- OUTPUT command, 6-65
- DJDEs, 8-7

Online Interface unit (OLI), 4-58

online

- 3211/4245 mode, 4-59
- dump, 4-64 to 4-65
- graphic formats
 - batch mode, 9-7
 - document and page interleaves, 9-6
 - document interleaved file transfers, 9-7
 - management, 9-7
 - noninterleaved, 9-6
- printing systems, 4-58 to 4-69
- record length, 4-64

- recovery, 4-65
- report separation, 4-61
- restrictions, DJDE, 8-19
- specific commands
 - CMEs, 4-61
 - DJDE processing, 4-61
 - JDE, creating, 4-59
 - JDL, creating, 4-59
 - optimization, 4-61
- Operator Communication Subsystem, *see* OCS
- operator
 - functions, restricting, 6-2
 - information, DJDE, 8-14
- operating system
 - executive, 1-6
 - software, 1-5 to 1-6
- OPRINFO parameter, 8-8
- options, cover parameter, 6-52
- ordered stocks, 5-10
- OSTK parameter, 6-66
- OTEXT parameter
 - ABNORMAL command, 6-7
 - DJDE command, 8-45
 - MESSAGE command, 6-46
- OUTPUT command, 6-47 to 6-89
- output processing task, 1-7
- OVERPRINT parameters
 - DJDE command, 8-46
 - LINE command, 6-36

P

- packed data formats, 4-70
- page
 - buffer, 1-7
 - log, 1-7
 - marker, 6-7
 - numbering, 6-64
 - orientation
 - landscape, 3-21 to 3-22
 - portrait, 3-21 to 3-23
 - sizes
 - edgemarking, 3-20
 - physical, 3-20
 - system, 3-20
- Page Descriptor Entry, *see* PDE command
- PAGES parameter, 6-7
- paper requirements, defining, 3-14
- PAPERSIZE parameter, 6-67
- parameter
 - commands, 2-4
 - ITEXT, 8-40
 - options, 2-5
 - rules, 8-23
- PASSES parameter, 7-26

- PCC command
 - parameters
 - ADVTAPE, 4-12
 - ASSIGN, 4-13 to 4-15
 - DEFAULT, 4-16
 - INITIAL, 4-17
 - MASK, 4-17
- PCCTYPE parameter, 6-38
- PDE command, 6-89
- PDL
 - capabilities, 1-1
 - command structure
 - components, 2-2 to 2-3
 - line length, 2-2
 - commands, grouping, 2-16
 - compiler, 1-7
 - features and functions, 1-1
 - filename
 - definitions, 3-18
 - options, 3-18
 - parameter options, 3-19
 - instructions, 1-7
 - processor, 1-6
 - related programs and tasks, 1-6 to 1-7
 - syntax, 3-6 to 3-8
- PMODE parameter
 - DJDE command, 8-47
 - PDE command, 6-93 to 6-94
- POSITION parameter, 6-22
- POSTAMBLE, 4-6
- PREAMBLE, 4-6
- PREFIX parameter, 8-8
- preparing print jobs, 1-8
- Print Description Language, *see* PDL command
- Printer Carriage Control, *see* PCC command
- printing
 - duplex, 7-20
 - format commands, 6-1
 - simplex, 7-20
- programs
 - coded instructions, 1-2
 - file control, 1-6
- programmer, 1-8
- PURGE parameter, 6-68

R

- random mode, graphics, 9-9
- RAUX
 - command, 7-20 to 7-21
 - duplex printing, 7-20
 - simplex printing, 7-20
 - test, 7-20
- raw data, *see* unformatted data
- RDELETE command, 7-24

RECORD

- command parameters
 - ADJUST, 4-19
 - CONSTANT, 4-20
 - FORMAT, 4-20
 - LENGTH, 4-21
 - LMULT, 4-21
 - LTHFLD, 4-22
 - OFFSET, 4-22
 - POSTAMBLE, 4-23
 - PREAMBLE, 4-23
 - STRUCTURE, 4-24
- formats, 4-71
- structure, 4-71
- recovery, automatic, 6-9
- registration shift and skew
 - landscape, 3-24
 - portrait, 3-25
 - margins, maintaining, 3-25
- REP parameter, 6-8
- replacement, hierarchy, 2-16 to 2-21
- reports
 - batch mode, 9-7
 - processing, 9-7
 - online, 9-7
 - separation, 7-13
- Report task (RPT), 1-7
- repositioning logical page, 7-29
- RESOLUTIONS
 - parameters, 6-68
 - statements, 6-68
- restricting operator functions, 6-2
- restrictions, graphics, 9-10
- RFEED command, 7-22
- RFORM parameters
 - DJDE command, 8-47
 - ROUTE command, 6-97
- Right-part constants, 2-6
- ROFFSET command, 7-26
- ROUTE command, 6-96 to 6-99
- RPAGE command, 7-29
- RRESUME commands, 7-41 to 7-44
- RSELECT command, 7-24
- RSTACK
 - commands, 7-35 to 7-40
 - delimiter modes, 7-35
 - online usage, 7-36 to 7-37
- RSUSPEND commands, 7-41 to 7-44
- RTEXT command
 - files, cataloged, 6-96
 - parameters
 - DJDE command, 8-48
 - ROUTE command, 6-98 to 6-99

S

- SAVE parameter, 8-49
- SECURITY parameter, 6-8
- SEFFNT parameters
 - MAP, 4-29
 - SEFMAP, 4-29
- SEFMAP parameter, 8-50
- segment management, 6-25
- SEPARATORS
 - parameter
 - DJDE command, 8-51
 - EXPORT command, 6-26
- separate disk access, 9-9
- SF1FUNCTION parameter
 - DJDE command, 8-52
 - OUTPUT command, 6-69
- SF2FUNCTION parameter
 - DJDE command, 8-52
 - OUTPUT command, 6-69
- shift and skew, registration
 - landscape, 3-24
 - portrait, 3-25
 - margins, maintaining, 3-25
- SHIFT parameter
 - DJDE command, 8-53
 - OUTPUT command, 6-70
- side parameter
 - DJDE command, 8-54
 - RPAGE command, 7-29
- simple and OTEXT applications, 5-4
- simplex printing, RAUX, 7-20
- SIZING parameter, 6-71
- SKIP parameter, 8-9
- SNUMBER parameter
 - DJDE command, 8-55
 - EXPORT command, 6-27
- software operating system, 1-4 to 1-5
- source code files, saving, E-9
- spacing
 - characters, 3-26
 - fixed, 3-26
 - proportional, 3-26
- special conditions, 6-44
- special processing, *see* logical processing
- specifications, IDEN command parameter, 1-7
- specifying
 - ACCT command parameters, 3-12
 - CRITERIA commands, 7-6 to 7-7
 - DJDE, usage, 3-13
 - formats, 3-13
 - input parameters, 4-1
 - LINE command parameters, 3-12
 - logical
 - functions, 7-1
 - processing, 3-13

- output requirements, 3-15
 - VFUs, 3-11
- SPLIT parameter
 - DJDE command, 8-56
 - EXPORT command, 6-28
- SRECOVER parameter
 - DJDE command, 8-57
 - EXPORT command, 6-29
- standard character set
 - ASCII, C-6
 - EBCDIC, C-7
- Start command, 1-9
- status display, 7-36
- STIMING parameter
 - DJDE command, 8-58
 - EXPORT command, 6-29
- STOCKS parameter
 - DJDE command, 8-58
 - OUTPUT command, 6-72
- STOCKSET
 - applications, 5-5
 - changes, 5-10
 - command, 6-100 to 6-102
- string
 - comparisons, 7-10
 - constants
 - ASCII, 2-7
 - character, 2-7
 - EBCDIC, 2-8
 - H2 and H6, 2-8
 - hexadecimal, 2-7
 - keyword, 2-6
 - octal, 2-8
 - variable name, 2-6
- subfield, 7-14
- subsystem, operator communication, 1-6
- suppressing, print function, 7-42
- symbols, command syntax
 - DJDE, A-1 to A-6
 - PDL quick reference, B-1 to B-7
- syntax, PDL, 3-9
- SYSPAGE parameter, 6-101
- SYSPPR parameter, 6-73
- system
 - administrator, 1-8
 - default values, 2-16
 - disk
 - creating, E-1 to E-7
 - stored files, E-3
 - executive task, operating, 1-6
 - pages, 6-97
 - software tapes, JDL, D-4

T

- TABLE commands, 7-45 to 7-48
- tape formats
 - batch mode, 9-5
 - computer, host, 4-70
 - document interleaved, 9-4
 - noninterleaved, 9-3
 - page interleaved, 9-4
- tasks
 - editor, 1-6
 - operating system executive, 1-6
- TCODE command parameters
 - DEFAULT, 4-33
 - TASSIGN, 4-34
 - TRESET, 4-35
- TCOUNT parameter, 7-15
- TEST parameter
 - BANNER, 7-16
 - CRITERIA, 7-6
 - RAUX, 7-20
 - RFEED, 7-22
- third-party devices, 6-69
- TMODE parameter
 - DJDE, 8-59
 - OUTPUT command, 6-74
- TOF parameters
 - DJDE command, 8-59
 - VFU command, 6-103
- top-of-form (TOF) line, 6-103
- tracking information, 1-7
- trailer banner pages, 7-15
- TRANS parameter
 - DJDE command, 8-60
 - OUTPUT command, 6-75
- transparencies, 6-75
- two-hole paper, 6-76
- TYPE parameter, 7-16

U

- UCSB
 - parameter
 - ONLINE printing system, 4-62 to 4-63
 - LINE command, 6-39
 - processing, 4-62
- unformatted data, 1-11 to 1-12
- UNITS parameter, 6-75
- UNIVAC ASCII character set, C-5
- Universal Character Set Buffers (UCSBs), 4-62 to 4-63
- user-controlled offsets, 7-26
- USER parameter, 6-16

V

valid host specifications
 computer, D-2 to D-3
 label, D-2 to D-3
value constants, 2-6
variable
 data, 1-7
 name constants, 2-6
VFU parameter 6-103 to 6-106
VOLUME command parameters
 BMULT, 4-39
 CODE, 4-40 to 4-41
 EOV, 4-41
 HOST, 4-42 to 4-44
 INTERPRESS, 4-45
 LABEL, 4-46
 LCODE, 4-47
 LPACK, 4-48
 MAXLAB, 4-48
 MINLAB, 4-49
 OPTIMIZE, 4-50
 OSCHN, 4-51
 OSHDP, 4-51
 OSTLP, 4-52

PLABEL, 4-52
RMULT, 4-53
RSAT, 4-53
TCODE, 4-54
UNPACK, 4-55

W

WAIT messages, 6-7
work file, modifying, E-5 to E-7

X

Xerographic Mode Persistence (XMP), 6-77
Xerographic Mode Switching (XMS), 6-77
XMP parameter, 6-77
XMS parameter, 6-77
XSHIFT parameter
 DJDE command, 8-60
 OUTPUT command, 6-76

Z

ZERO parameter, 4-7

