

**Xerox 4050/4090/4450/4650  
Laser Printing Systems  
Print Description Language (PDL) Reference**

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

© 1991, 1992, 1993, 1994, 1995 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: 720P94090

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 the Xerox 6085 Professional Computer System using GlobalView software. The typeface is Optima.

---

---

# Table of contents

<b>Introduction</b>		xiii
	Document conventions	xiii
	Related publications	xiv
<b>1. Overview</b>		1-1
	Input datastreams	1-1
	Offline mode	1-1
	Host tape formats	1-2
	Data representation	1-2
	Packed data formats	1-2
	Record formats	1-3
	Record structure	1-3
	Multivolume processing	1-3
	Online mode	1-4
	4245 protocol emulation features	1-4
	Online-specific commands	1-4
	DJDE processing	1-5
	Copy-sensitive CMEs	1-5
	Report separation	1-5
	Online record length	1-5
	Online recovery	1-6
	Online dump	1-6
	Downloading files from the host to the LPS	1-9
	FCBs	1-10
	Vertical format control processing	1-10
	UCSBs for 3211	1-11
	PDL command and DJDE syntax	1-12
	Command length	1-13
	Identifiers	1-13
	Command keywords	1-14
	Parameter keywords	1-14
	Parameter options	1-14
	Value constants	1-15
	String constants	1-15
	JSL structure	1-18

	JDL name	1-18
	ID level	1-18
	System or JDL level	1-18
	Catalog level	1-19
	Job or JDE level	1-19
	Comments	1-19
	END command	1-20
	Hierarchy of replacement	1-20
	PDL library	1-24
JDL creation		1-25
	Required components of a JSL	1-25
	Steps in creating a JDL	1-25
JDL coding		1-26
Data definition commands		1-27
Print format commands		1-27
Print control (DJDE) commands		1-28
	DJDE orientation	1-29
	DJDE record specification	1-32
	Application of DJDEs	1-33
	DJDE operator information pages	1-35
	Job parameter modification restrictions	1-35
Logical processing commands		1-37
	Logical processing command format	1-38
	Logical processing commands with TEST parameters	1-39
	TEST expression definition	1-39
	String comparison concepts	1-39
	Character types	1-40
	Masked comparisons using default type assignments	1-40
JSL compilation		1-41
	Printing a job	1-41
<b>2. Data definition commands</b>		<b>2-1</b>
Input processing commands		2-1
	BLOCK	2-1
	ADJUST	2-1
	CONSTANT	2-1
	FORMAT	2-2
	LENGTH	2-2
	LMULT	2-2
	LTHFLD	2-3
	OFFSET	2-3

POSTAMBLE	2-3
PREAMBLE	2-4
ZERO	2-4
BLOCK example	2-4
<i>ac:CODE</i>	2-5
ASSIGN	2-5
DEFAULT	2-5
IDEN	2-6
OFFSET	2-6
OPRINFO	2-7
PREFIX	2-7
SKIP	2-8
<i>ac:PCC</i>	2-8
ADVTAPE	2-8
ASSIGN	2-9
DEFAULT	2-10
INITIAL	2-11
MASK	2-11
RECORD	2-12
ADJUST	2-12
CONSTANT	2-12
FORMAT	2-12
LENGTH	2-13
LMULT	2-13
LTHFLD	2-14
OFFSET	2-14
POSTAMBLE	2-14
PREAMBLE	2-14
STRUCTURE	2-15
RECORD example	2-15
<i>ac:TCODE</i>	2-17
DEFAULT	2-18
TASSIGN	2-18
TRESET	2-19
VOLUME	2-20
BMULT	2-20
CODE	2-20
EOV	2-21
HOST	2-22
INTERPRESS	2-23

LABEL	2-24
LCODE	2-24
LPACK	2-25
MAXLAB	2-25
MINLAB	2-25
OPTIMIZE	2-26
OSCHN	2-26
OSHDP	2-27
PLABEL	2-27
RMULT	2-27
RSAT	2-27
TCODE	2-28
UNPACK	2-28
VCODE (XDDI only)	2-29
VOLUME example	2-30
<b>3. Print format commands</b>	<b>3-1</b>
Print format command functions	3-1
ABNORMAL	3-1
ERROR	3-1
OTEXT	3-2
SECURITY	3-3
ACCT	3-4
DEPT	3-4
USER	3-5
ACCT example	3-6
<i>ac:CME</i>	3-6
CONSTANT	3-6
FONT	3-7
LINE	3-8
POSITION	3-9
EXPORT (XDDI only)	3-9
SEPARATORS	3-10
SNUMBER	3-11
SPLIT	3-11
SRECOVER	3-12
STIMING	3-12
LINE	3-13
DATA	3-13
FCB	3-14
FONTINDEX	3-14

---

MARGIN	3-16
OVERPRINT	3-16
PCC	3-17
PCCTYPE	3-17
UCSB	3-18
VFU	3-18
LINE example	3-19
MESSAGE	3-20
ITEXT	3-20
OTEXT	3-20
OUTPUT	3-21
BFORM	3-23
COLLATE	3-24
COPIES	3-24
COVER	3-24
CYCLEFORMS	3-25
DENSITY	3-26
DESTINATION	3-26
DUPLEX	3-26
FACEUP	3-27
FEED	3-27
FORMAT	3-28
FORMS	3-30
GRAPHICS	3-30
IMAGE	3-31
MODIFY	3-32
NTO1	3-32
NUMBER	3-33
OFFSET	3-34
PAPERSIZE	3-34
PURGE	3-35
RESOLUTION	3-35
SHIFT	3-36
SIZING	3-36
STAPLE	3-37
STOCKS	3-37
UNITS	3-38
OUTPUT examples	3-38
ac:PDE	3-38
BEGIN	3-40





TEST	4-13
RDELETE example	4-13
RFEED	4-14
TEST	4-14
ROFFSET	4-15
PASSES	4-15
TEST	4-15
ROFFSET example	4-16
RPAGE	4-16
SIDE	4-16
TEST	4-19
WHEN	4-19
RRESUME	4-21
BEGIN	4-21
TEST	4-21
RRESUME example	4-22
RSELECT	4-23
TEST	4-23
RSTACK	4-23
ACCTINFO	4-24
DELIMITER	4-24
HRPTNA	4-25
PRINT	4-25
TEST	4-26
RSTACK example	4-26
RSUSPEND	4-27
BEGIN	4-27
TEST	4-28
<i>ac:TABLE</i>	4-28
CONSTANT	4-29
MASK	4-29
<i>ac:TABLE</i> example	4-30

<b>5. Print control (DJDE) commands</b>	5-1
ALTER	5-1
ASSIGN	5-2
BATCH	5-2
BEGIN	5-3
BFORM	5-3
BOF	5-4
<i>C text</i>	5-4

CANCEL	5-4
COLLATE	5-5
COPIES	5-5
DATA	5-6
DEPT	5-6
DESTINATION	5-6
DUPLEX	5-7
END	5-7
FEED	5-7
FILE	5-8
FONTINDEX	5-11
FONTS	5-11
FORMAT	5-12
FORMS	5-12
GRAPHICS	5-13
IMAGE	5-14
ITEXT	5-15
JDE	5-15
JDL	5-16
MARGIN	5-16
MODIFY	5-16
NUMBER	5-17
OTEXT	5-18
OVERPRINT	5-18
PMODE	5-19
RFORM	5-19
RTEXT	5-20
SAVE	5-21
SEPARATORS	5-21
SHIFT	5-22
SIDE	5-23
SNUMBER	5-24
SPLIT	5-25
SRECOVER	5-26
STOCKS	5-27
TOF	5-27

**6. Graphics**

	6-1
Processing modes	6-1
Batch mode	6-1
Block mode	6-1

	Move mode	6-2
	Random mode	6-2
Tape formats		6-2
	Noninterleaved	6-2
	Document-interleaved	6-2
	Page-interleaved	6-3
	Batch mode	6-5
Online formats		6-5
	Noninterleaved	6-5
	Document- and page-interleaved	6-6
	Batch mode	6-6
	Document-interleaved graphic file transfers	6-7
	Management of image files	6-7
PDL command options		6-7
Performance considerations		6-8
Graphic features restrictions		6-8
<b>7. Paper stocks and clusters</b>		7-1
	Programmer and operator tasks	7-1
	The CLUSTER.LIB file	7-1
	Paper stock commands and DJDEs	7-2
	Cluster names and references	7-3
	Defining stocksets	7-3
	RAUX options	7-3
	Creating clusters	7-4
	Using clusters in print jobs	7-6
	Simple and OTEXT applications	7-6
	STOCKSET applications	7-6
	Mixing applications	7-8
<b>Appendices</b>		
	A. PDL command summary	A-1
	B. Character code assignments	B-1
	C. Offline specifications	C-1
	D. Online specifications	D-1
	E. 4850 Highlight Color and 4135 LPS compatibility	E-1
<b>Glossary</b>		GLOSSARY-1
<b>Index</b>		INDEX-1



The *Xerox 4050/4090/4450/4650 Laser Printing Systems Print Description Language Reference* explains the key concepts involved with the Xerox Print Description Language (PDL) and describes the commands and parameters you use to create and control print jobs.

This reference is intended for both novice and experienced PDL programmers. It assumes some prior knowledge or experience with computer programming.

---

## Document conventions

---

This manual uses the following conventions:

<b>UPPERCASE BOLD BLUE</b>	Uppercase bold blue text indicates required characters or command keywords.
<i>UPPERCASE BLUE ITALICS</i>	Uppercase blue italics indicate optional parameter keywords, characters, or values.
<i>Lowercase black italics</i>	Lowercase black italics indicate variable parameter options, (word, character, phrase, or value).
...	Ellipses indicate that you can repeat a parameter option, or list a series of parameter options.
<>	Angle brackets indicate keys on the system controller keyboard.
	The carat character represents a required space.
TERMINAL FONT	Terminal or monospace fonts are used to represent LPS screen responses.
UPPERCASE	Uppercase letters indicate command names and parameter keywords.
<b>CAUTION:</b>	Cautions appear immediately before any action or omission that may result in damage to your equipment, software, or data.
<b>WARNING:</b>	Warnings are associated with the safety of people.

---

## Related publications

---

You can find additional information related to the Xerox 4050/4090/4450/4650 LPS in the following publications.

<b>Publication</b>	<b>Number</b>
<i>Xerox 4050/4090/4450/4650 LPS Master Index</i>	720P94030
<i>Xerox 4050/4090/4450/4650 LPS Product Reference</i>	720P93010
<i>Xerox 4050/4090/4450/4650 LPS Bypass Transport Instructions, V3.5/3.8</i>	720P22320
<i>Xerox 4050/4090/4450/4650 LPS Bypass Transport Operator Training Guide Flipcharts Supplement</i>	720P22340
<i>Xerox 4050/4090/4450/4650 LPS Bypass Transport Operator Training Guide Supplement</i>	720P22330
<i>Xerox 4050/4090/4450/4650 LPS Command Reference</i>	720P94020
<i>Xerox 4050/4090/4450/4650 LPS Installation Planning</i>	720P92990
<i>Xerox 4050/4090/4450/4650 LPS Operator Command Summary</i>	720P93050
<i>Xerox 4050/4090/4450/4650 LPS Operator Guide</i>	720P94000
<i>Xerox 4050/4090/4450/4650 LPS Operator Training Guide</i>	720P22070
<i>Xerox 4050/4090/4450/4650 LPS Operator Training Guide Flipcharts</i>	720P22080
<i>Xerox 4050/4090/4450/4650 LPS System Administration Guide</i>	720P94010
<i>Xerox 4050/4090/4450/4650 LPS System Administration Quick Reference Card</i>	720P93090
<i>Xerox 4050/4090/4450/4650 LPS Forms Creation Guide</i>	720P93990
<i>Xerox 4050/4090/4450/4650 LPS Forms Creation Quick Reference Card</i>	720P93100
<i>Xerox 4050/4090/4450/4650 LPS Message Guide</i>	720P93980
<i>Xerox 4050/4090/4450/4650 LPS Print Description Language (PDL) Quick Reference Card</i>	720P93090
<i>Xerox Dynamic Document Interface Command Summary</i>	720P13680
<i>Xerox Dynamic Document Interface Operator Guide</i>	720P13670

This chapter contains basic information on using the Xerox Print Description Language (PDL) to create and control print jobs.

For a job to be printed on an LPS, you first create a file of PDL commands to define the source and format of the input media, processing requirements, and the print format, such as forms, fonts, accounting options, operator messages, and finishing. Each PDL command has a set of parameters that define these characteristics.

The source or uncompiled file of PDL commands is called a Job Source Library (JSL). 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 called a Job Descriptor Library (JDL).

The primary element of a JSL is a Job Descriptor Entry (JDE). The words "JDE" and "job" are used interchangeably. A JDE or job is a group of single data sets, called reports, that are delimited by the RSTACK command (discussed in the "Logical processing commands" chapter) or as separate files. Each JDE has a name that is specified when running the print job. If you do not override the system default settings with other particular print job characteristics in your JDL, the system default settings take effect. Default settings are set up during system generation.

Dynamic Job Descriptor Entries (DJDEs) can be inserted into the input data stream to enable you to modify print job characteristics dynamically on a page-by-page or record-by-record basis. A record is a line of data as defined in the RECORD command (discussed in the "Data definition commands" chapter).

---

## Input data streams

---

Before anything else, you must know the source of the input data and code the input data characteristics into your JSL. Data processed by the printing system may originate from several sources, including magnetic tape (offline processing), a host computer (online processing in 3211 or 4245 modes), through remote communications, or over an Ethernet connection.

---

## Offline mode

---

For offline operations, magnetic tape may be recorded in one of a variety of standard formats. As the programmer, you define the tape blocking and record format parameters. These parameters reduce physical tape blocks first to logical records, then to print lines. You can also code special processing commands that enable logical report processing.

Before selecting the PDL commands that describe a specific job tape, you must understand the concepts of host formats, block and record structure, tape translation code, and packed data formats.

## Host tape formats

---

The laser printing system (LPS) processes data tapes in standard host formats. Refer to the "Offline specifications" appendix for information on these host formats. You must specify the appropriate generated tape format in the HOST parameter of the VOLUME command. The VOLUME command is discussed in the "Data definition commands" chapter. The format of each tape is described in the *Xerox LPS Tape Formats Manual*.

## Data representation

---

The LPS recognizes EBCDIC, ASCII, several versions of BCD tape codes, and the Xerox LPS native format. It also accepts the VOLUME command CODE=NONE parameter which instructs the system not to translate the input data. If the codes are not sufficient for a particular tape, you may create a new code translation table or modify an existing one. Tables showing the correlation between standard recording codes and printed characters are in the "Character code assignments" appendix. 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 an open-reel 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. There are two methods of packing these bits together: using the Honeywell 6000 (T4X3), or the Honeywell 2000 (T4X3H2).

Whenever an unpacking method is included in the JDL, the system unpacks the characters before processing the data. 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 you can specify one of the three standard BCD CODE parameter options (H2BCD, H6BCD, and IBMBCD). The "Character code assignments" 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 LPS are either blocked or unblocked with a fixed length, a variable length, or an undefined format. The BLOCK and RECORD commands 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 the "Data definition commands" 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 portion of the record contains information provided by the application or user program running on the host system. The boundary between the two portions of the record is traditionally between the record length and the Printer Carriage Control (PCC) field. If there is no record length field, there is no operating system portion of the record. The input record characteristics that define the components of a record are described in the "Data definition commands" chapter.

---

## Multivolume processing

---

All multivolume reports that force input to make multiple passes over the data are handled in the following way. 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 Copy Modification Entries (CMEs); and, optionally, for multicopy reports that exceed the size of the print file. CMEs are discussed in the "Print format commands" chapter. The choices available for handling print file saturation are discussed under the VOLUME command RSAT parameter in the "Data definition commands" chapter.

If a multivolume report requires multiple passes, messages appear on the system controller display with instructions on which actions to take.

If you have multiple volume jobs and multiple laser printing systems, you may wish to print each volume on a separate system. This requires that each volume be processed independently. Refer to the VOLUME command EOVS parameter section in the "Data definition commands" chapter for more information about this processing.

If a system rollover occurs while processing a multivolume report, the system displays the following message:

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

before performing a forward tape space to the recovery point. 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.

## Online mode

---

A channel-attached LPS can receive input from the host computer in online 3211 or 4245 protocol emulation mode. In online mode, the data stream emulates IBM 3211 or 4245 line printer format, with or without embedded DJDEs. All commands coded for the 3211 are recognized by 4245 protocol emulation.

## 4245 protocol emulation features

---

The 4245 protocol emulation allows your LPS to correctly execute the printing instructions coded within the job stream for the IBM 4245 printer.

The 4245 is identical to the 3211 in that all 3211 commands are recognized by 4245 protocol emulation.

## Online-specific commands

---

You create an online Job Descriptor Library (JDL) and its Job Descriptor Entries (JDEs) 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 the following 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.

When `VOLUME HOST=IBMONL` is specified, the following defaults are automatically selected:

```
RECORD LENGTH=150;  
LINE    PCCTYPE=IBM3211,  
        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 channel command before sending it to the printer. Any commands other than those listed are rejected by the LPS online interface (and by the 3211 or 4245). Therefore, the only appropriate `PCCTYPE` for an online JSL file is `IBM3211`, which is the online default.

Because the carriage control command is sent prior to the data record (rather than as part of it), offset to fields (such as `DJDE` prefix and font index) are typically one byte less for online than offline mode, where the carriage control byte is part of the data record. For the same reason, the online `DATA` parameter of the `LINE` command typically has a default offset of zero rather than one.

The `OPTIMIZE` parameter of the `VOLUME` command allows you to select options that enhance online throughput. Refer to the "Data definition commands" chapter for additional information.

---

## DJDE processing

---

Printing is controlled through parameters from a user-defined JDL file which may be dynamically overridden by Dynamic Job Descriptor Entries (DJDEs). The DJDEs that you specify are processed by the LPS as part of the input print data from the host system. The "Print control (DJDE) commands" chapter defines their syntax and options. The IDEN command enables the system to identify DJDEs as part of the input data. Refer to the "Data definition commands" chapter for information on the IDEN command.

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 such as font files can be downloaded from the host using the FILE DJDE. This DJDE is described in the "Print control (DJDE) commands" chapter.

---

## Copy-sensitive CMEs

---

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, you must transmit data the same number of times that the data needs to be modified. For example, if you want six copies of a report and one CME applies to copies 1 through 4, and another CME applies to copies 5 and 6, you must transmit the report 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 "Print format commands" chapter for additional information.

---

## 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 trays. The two logical processing commands which can control online report separation are BANNER and RSTACK. Refer to the "Logical processing commands" chapter for additional information.

---

## Online record length

---

The default record length supported by Xerox laser printing systems is 150 bytes, the same default as the 3211 printer. However, you may choose to use larger record lengths with the following restrictions:

- For print record lengths from 151 bytes to 214 bytes, do not use the print position indexing feature that is normally

available on the 3211 or 4245. This feature is invoked only with a Forms Control Buffer (FCB). If FCB=IGNORE is invoked, this restriction is not significant.

- For print record lengths from 215 bytes to 2140 bytes, do not use the OPTIMIZE parameter of the VOLUME command, which provides online buffering for improved performance.
- If the data record transmitted from the host exceeds the RECORD command LENGTH parameter values specified, the record is truncated to the specified record length and no warning is provided. This action is consistent with the 3211 or 4245 printers.
- 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.
- You cannot change the LENGTH parameter of the RECORD command with 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.
- You can change the OPTIMIZE parameter of the VOLUME command with the DJDE JDE subject to the restrictions listed in the OPTIMIZE parameter section. Refer to the "VOLUME command" section of the "Data definition commands" chapter for additional information.

To select a record length longer than 150 bytes, you 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.

Figure 1-5 shows a complete online Job Source Library (JSL).

## Online recovery

---

After a system failure, data that has been acquired from the host and stored on the system disk can be imaged upon system restart, if you have requested job recovery. Only data that was being received at the time of the power failure is lost and must be retransmitted.

For power failures, the amount of data that requires retransmission can be up to 8 Kbytes. In all other rollover conditions, the data is normally recovered.

## Online dump

---

The online dump feature allows you to generate a hardcopy of the command and data transmission taking place between the host computer and the LPS.

### Starting and ending dump sessions

---

You may begin an online dump session with a START command when the LPS is online and in the idle state. You must use the DFLT JDE within the OLDUMP JDL to specify VOLUME HOST=OLDUMP and CODE=EBCDIC.

The dump session is terminated by the operator ENDJOB command or the offline operator command. Other online operator commands during a dump session perform the standard functions. These are described in your *Xerox 4050/4090/4450/4650 LPS Operator Guide*. Figure 1-1 is a sample printed online dump.

---

### Dump format

---

Each host command directs the printer to perform an operation or provides control information used during report printing. The operations performed are either print with carriage control, carriage control only, or printer control, such as load FCB. All print commands and some control commands transmit data to and from the LPS. The dump format is designed so that there is a HEADER and DATA segment for each of these commands. The HEADER segment consists of the following fields:

#### HOST COMMAND

The English translation of the host command in hexadecimal.

#### END STATUS

One byte of status in hexadecimal transmitted to the host.

#### SEQ#

An incremental count, in decimal, of the host commands received during a dump session. The maximum number generated is 999,999.

#### LENGTH

The length, in decimal, of the transmitted data.

#### DATE and TIME

The 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, as well as an 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.

Online dumps have the following restrictions:

- DJDEs, BANNER, and other logical processing functions are not processed when the dump job is running; you must enter an operator ENDJOB command to end the dump session.
- The NO-OP command from the host is not dumped since it is intercepted and processed by the OLI hardware and is transparent to the LPS software.
- Hardware errors are reported in the system error log, but are not dumped due to host, OLI, 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 command LENGTH parameter value are truncated without warning. If records longer than 150 bytes are to be sent, the RECORD command LENGTH parameter in the OLDUMP JDL should be modified to the longer length.

- The operator Host Interface Processor “HIP START ...” command is accepted, but HIP is unable to submit a print job while the system is online.

Figure 1-1. **Sample printed online dump**

---

## Downloading files from the host to the LPS

---

Use one of the following methods for downloading files:

- HOSTCOPY utility
- FILE Dynamic Job Descriptor Entry
- LPS and host file transfer.

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.

---

### HOSTCOPY

---

HOSTCOPY provides a means for accepting files transmitted from the host computer and storing those files on the LPS system disk. Files in the following formats may be transmitted:

- User-created card-image files
- LPS-labeled files (such as font, logo, and patch tapes). Refer to the *Xerox 4050/4090/4450/4650 LPS System Administration Guide* for information on the HOSTCOPY command.

The 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.

---

### FILE Dynamic Job Descriptor Entry

---

The FILE Dynamic Job Descriptor Entry (DJDE) 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. FILE provides a capability of loading card-image or LPS-labeled files to the LPS disk while a printing job is in progress.

FILE 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 "Print control (DJDE) commands" chapter for FILE syntax and to the *Xerox 4050/4090/4450/4650 LPS System Administration Guide* for information on card-image file processing.

---

### LPS and host file transfer

---

A third method of file transfer uses the LPS and host file transfer interface. This interface transfers user files and allows you 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.

## FCBs

---

The 3211 and 4245 Forms Control Buffer (FCB) defines channel positions and forms length. You may restrict the Forms Control Buffer information by accepting or suppressing the normal processing of host-transmitted FCB input. You can accomplish this by specifying the FCB=IGNORE parameter of the LINE command.

## Vertical format control processing

---

The interaction of the vertical format controls (VFU) in the JSL and the FCB sent from the host is as follows:

- When you enter a START command, the VFU table and its associated bottom-of-form (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 top-of-form (TOF) is not affected by the FCB.
- If you specify 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 is reactivated.
- When system generation 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 you specify 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.



---

## UCSBs for 3211

---

The 3211 Universal Character Set Buffer (UCSB) feature is supported by the LPS. LPS software uses UCSBs as a basis for generating 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. 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 or JDL is processed
- A new UCSB is transmitted from the host
- A new FOLD or UNFOLD command is received.

The selection of a folded or unfolded translate table is based on the FOLD and UNFOLD channel 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.

You may restrict UCSB information by accepting or suppressing the normal processing of host-transmitted UCSB data using 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.

You may change the UCSB parameter 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, you may do one of the following:

- Switch to UCSB=PROCESS in a selected JDE
- Load a UCSB with lowercase characters defined as not printable and with FOLD specified.

At the start of the next report, UCSB=IGNORE is reactivated and the translate table is rebuilt based on the CODE command in the JDL file.

---

### UCSB processing for 3211

---

When a UCSB is transmitted from the host, it is saved on an LPS disk. When sysgen is executed, the UCSB saved on the disk is reinitialized to define all characters as printable.

If you specify UCSB=PROCESS and the UCSB suppresses special characters used in DJDE syntax, for example, ; or ( ), DJDE syntax errors result.

Use UCSB=IGNORE if a code other than EBCDIC is specified.

If you specify UCSB=PROCESS, a UCSB load causes the CODE default to revert to EBCDIC until the end of the report, or until a selected JDE or JDL is processed.

UCSBs should be ignored in the 4245 mode by coding UCSB=IGNORE. The UCSB host commands (LOAD UCSB, FOLD, and UNFOLD) are processed if UCSB=PROCESS is specified in the Job Source Library (JSL); however, the host does not normally issue such commands in 4245 mode.

---

## PDL command and DJDE syntax

---

You must code each PDL command with the words placed in a specific order and using certain punctuation marks in specific locations. This ordering of words and punctuation is called the syntax of the PDL language.

A PDL command consists of as many as nine elements:

- Identifier, which is used for some commands so that they may be referenced by other commands
- Colon (:), which always follows an identifier
- Command keyword (required)
- Parameter keyword (left part)
- Equal sign (=), which joins a parameter keyword with its parameter options
- Parameter option (right part)
- Parentheses, which groups multiple parameter options together
- A comma or blank space, which separates one parameter from another within a command
- A semicolon (;) at the end of a command statement (required).

A command keyword and a semicolon are the two required elements.

Figure 1-2, a diagram of a VFU command statement, illustrates the command syntax rules. Refer to the "Introduction" for the conventions used in this reference to illustrate the syntax statement for each PDL command.

Figure 1-2. **Command statement components**

The PDL command in figure 1-2 has an identifier (VFU0005:), a command keyword (VFU), and three parameters with options: ASSIGN=(1,5), TOF=5, BOF=66.

## Command length

You code a JSL using PDL commands through the system editor task. The editor accepts up to 80 characters per line. A record is the complete set of PDL syntax elements for a command and frequently uses several lines. Refer to your *Xerox 4050/4090/4450/4650 LPS System Administration Guide* for information on using the editor.

## Identifiers

The purpose of an identifier is to provide a label for a command so that it can be referenced by other commands. If the identifier is coded with the command, it is called an ID command. Not all commands have identifiers. The following commands require identifiers:

- *ac*:CATALOG
- *ac*:CME
- *ac*:CODE
- *ac*:CRITERIA
- *ac*:IDR
- *dd*:JDE
- *dd*:JDL
- *dd*:JOB
- *ac*:PCC
- *ac*:PDE
- *ac*:ROUTE
- *ac*:STOCKSET
- *dd*:SYSTEM
- *ac*:TABLE
- *ac*:TCODE
- *ac*:VFU.

Identifiers have these requirements:

- 1 to 6 characters in length. The convention *ac* indicates that at least one of the characters must be an alpha (letter). *dd* indicates that the alpha restriction does not apply. These unrestricted identifiers apply only to the SYSTEM (or JDL) and JOB (or JDE) commands.
- No blanks within the identifier are allowed, for example:
  - Invalid: VFU 1: VFU;
  - Valid: VFU1: VFU;
- Must be separated from the command with a colon, for example: VFU1:VFU.
- Avoid these categories of restricted names to prevent possible errors:
  - Command keywords such as OUTPUT, PDE, and RSTACK
  - Parameter keywords such as LCODE, UCSB, and BFORM
  - Abbreviated operator command keywords CAR (CARTRIDGE) and TAP (TAPE)
  - DJDE keywords such as PMODE, RTEXT, and BATCH

- Resident task files such as .PDL, .FDL, .FNT, .LGO, .CME, .LIB, and .FRM
- System utilities such as INPUT, EDIT, DSR, IPD, and OCS.

## Command keywords

---

Command keywords have the following requirements:

- Every command must have a keyword.
- They must be coded with at least the first three characters of the command. For example, RECORD can be coded as REC, RECO, RECOR, or RECORD. The exception is FORMS. Do not abbreviate FORMS to FOR or FORM because the system interprets it as FORMAT.

## Parameter keywords

---

Some commands have multiple parameters available. For example, the OUTPUT command has the following parameters:

```
OUTPUT FORMS=form-id,  
        DUPLEX=YES or NO,  
        COPIES=value;
```

FORMS, DUPLEX, and COPIES are parameter keywords. Parameter keywords have the following requirements:

- Must be coded with at least the first three characters of the parameter keyword. For example, DUPLEX can be coded as DUP, DUPL, DUPL, or the entire keyword DUPLEX.
- Must be followed by an equal sign (=) and at least one parameter option.

## Parameter options

---

Most parameter keywords have multiple parameter options available. Parameter options must be:

- Placed on the right side of the equal sign (=)
- Enclosed in parentheses if there are multiple options for the same parameter
- Either keywords or variable names and values, for example:

```
OUTPUT FORMS=form-id,  
        DUPLEX=YES or NO,  
        COPIES=value;
```

YES and NO are parameter option keywords; *form-id* and *value* are variables.

Parameter options are also called right part constants and may take one of two forms: value constants and string constants.

---

## Value constants

---

Value constants have arithmetic values. You should express them as decimal numbers. They may also be expressed as hexadecimal, octal, or character values, but these expressions are not recommended. Decimal constants may be signed (positive or negative) and in some cases may have fractional digits, for example:

```
PDE      BEGIN=(1,1,.37);
BLOCK    LENGTH=1320;
RECORD   LENGTH=132;
OUTPUT   IMAGE=(1.30CM,0.85IN);
```

---

## String constants

---

String constants are normally used to specify strings of characters or to reference identifier parameters. String constants may be expressed as keywords, variable names, hexadecimal, character, ASCII, EBCDIC, octal, or H2 and H6 values, but not as decimal numbers.

---

### Keywords

---

Keywords are terms that direct the system to perform specific predetermined activities. Keywords always consist of the same characters, for example:

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

YES, CONTINUE, WAIT, and BOTH are keywords.

---

### Variable names

---

You may use string constants to name such objects as forms, files, fonts, departments, and so on. Each name you assign identifies the unique object you wish to specify for your print jobs, for example:

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

SMPLE, SMPBK, BLUCVR, and CME12 are variable names.

## Hexadecimal

---

Hexadecimal constants are 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 be immediately preceded by the character sequence X apostrophe (X') to indicate that the following expression is in hexadecimal and ended with an apostrophe, for example:

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

## Characters

---

Character constants are 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 be immediately preceded and 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 fashion, such as the hexadecimal constant X'7D'. Character constants are inherently defined as EBCDIC and take their actual values from the standard EBCDIC table definition. Refer to the "Character code assignments" appendix for EBCDIC table definitions.

## ASCII

---

ASCII constants are used as string constants. Each character results in one byte. The constants must be preceded by the character sequence A apostrophe (A'), and followed by an apostrophe character. For example:

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

The ASCII string type allows you to embed hexadecimal representations of characters in a string. This is done by preceding the hexadecimal representation of the character with an exclamation point (!) 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 character sequence 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 exclamation (!) 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 in hexadecimal 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 character sequence 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 the 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 the "Character codes assignment" appendix), no specification greater than X'3F' generates a legal character. If anything from X'40' to X'FF' is coded, the system 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';
```

The T3 TABLE identifier in figure 1-5 has a repeat count.

## JSL structure

---

PDL commands are organized within a JSL in four groupings called command levels:

- ID level
- System (or JDL) level
- Catalog level
- Job (or JDE) level.

You may code commands within these levels following the JDL coding, which identifies the Job Descriptor Library. You do not need to use all command levels in a JSL. Besides the JDL coding, only the job command level is required. However, JSLs typically include several command levels. Some JSLs define only one print job, but it is much more common to find JSLs structured to include multiple jobs, as shown in figures 1-4 and 1-5.

There are many steps in creating a JSL and many ways to specify your print job requirements. The first element of a JSL is the JDL coding which names the JDL file. Any command may be coded at any of the four command levels but the following descriptions present the recommended and most common usage.

### JDL name

---

Begin your JSL by deciding on a name for the compiled JDL. Select a name that is related to the function of the JDL and identifies the type of jobs you are creating. For example, if your site has several input sources, you might want to name the JDL for an online job JDLONL: JDL;

Typically, you use the name of the command (JDL) as the first three characters. The format for naming a JDL is:

JDL-*id*:JDL;

For example, you might name a JDL with highlight color jobs JDLHLC:JDL;

HLC reminds you that the JDL contains highlight color jobs.

If DFAULT is coded as the JDL name, you do not need to specify the name when entering the START command. The START command is discussed in the "JSL compilation" section of this chapter and in your *Xerox 4050/4090/4450/4650 LPS Operator Guide*.

### ID level

---

Commands requiring identifiers are typically coded at this level. They must be coded in the library before they can be referenced by commands in other levels within the library. Therefore, ID level commands are the first to appear in a JSL after the JDL coding itself. The VFU command is usually coded at this level, as shown with VFU001:VFU in figure 1-4.

### System or JDL level

---

PDL commands coded at the system level establish default print job characteristics. Commands that are common to the JSL print jobs are listed at this level so that they do not need to be coded



in every job in the JSL. System level commands, however, may be overridden by commands at the job command level. In figure 1-4, the system level VOLUME command is overridden by the VOLUME parameter in 2:JDE at the job level.

## Catalog level

---

Commands coded at the catalog level are a subset of jobs within the JSL. They may be specified in one or more jobs within the JDL. If, for example, you want to print three jobs, two of which use a different VFU and form than the other job, you could code the different LINE and OUTPUT commands at this level and then include the catalog in the appropriate jobs at the JOB level. Use the following command to create a catalog name:

*catalog-name*: CATALOG;

The *catalog-name* may consist of 1 to 6 alphanumeric characters, for example, in figure 1-4:

CATPOW: CATALOG

The values of the commands coded at the catalog level override those coded at the ID and system levels when a catalog is referenced in a job. You reference catalogs at the job level with the JOB or JDE INCLUDE command. Refer to jobs 2:JDE and 3:JDE in figure 1-4 for an example of this setup, where the job level INCLUDE command references catalogs CATPOW and CATGRP, respectively. CATPOW and CATGRP override the VOLUME BLOCK and RECORD commands that are specified at the system level.

## Job or JDE level

---

Commands unique to a single job are coded at this level under a specific job name. Use the following command to create a job name:

*jde-name*:JOB INCLUDE=*catalog-name*;

The *jde-name*, like any identifier, may consist of 1 to 6 alphanumeric characters. The *catalog-name* option of the INCLUDE parameter is the identifier for the catalog you wish to specify for your print job, for example, in figure 1-4:

2:JOB; INCLUDE=CATPOW

If DFAULT is coded for a *jde-name*, you do not need to specify it when entering the START command. The START command is discussed in the "JSL compilation" section of this chapter.

## Comments

---

Comments make the JSL easier to understand and maintain. You can use comments to clarify where command levels begin and provide information to other programmers about particular aspects of the JSL or jobs.

Comments may appear anywhere within the JSL and must be preceded by the character sequence slash asterisk (/\*) and terminated with the character sequence asterisk slash (\*). There are numerous comments at the job level in figure 1-4.

Nested comments may be set within other comments. There is no practical limit to the level of nesting possible as long as each nested comment is preceded by the `/*` and terminated with the `*/`. The following is an acceptable nested comment format:

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

## END command

---

Every JSL must conclude with an END command so that the LPS knows where the PDL instructions end. The command is coded in the first column of the JSL as shown in figures 1-4 and 1-5.

The format is always:

```
END;
```

Use two consecutive END commands to indicate the end of a series of JDLS:

```
END;  
END;
```

## Hierarchy of replacement

---

The system default values shown in the "PDL command summary" appendix and in the command chapters 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 1-3.

Figure 1-3. Hierarchy of command replacement

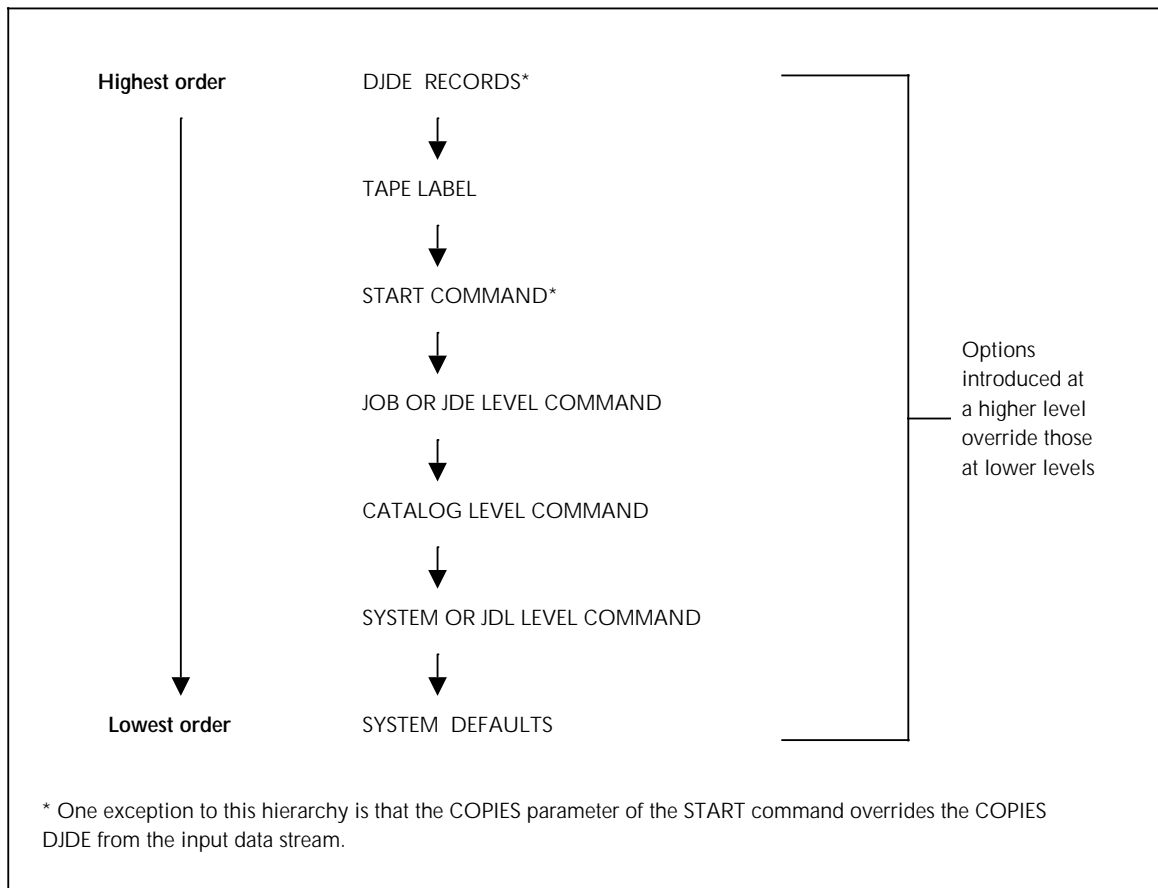


Figure 1-4 shows a coded JDL that contains four jobs. A command to specify the recording code (CODE parameter of the VOLUME command) of the input data appears in the following three places:

- At the system (or JDL) level, the default recording code of the input data is ASCII (VOLUME CODE=ASCII).
- At the catalog command level, the recording code of the input data is EBCDIC (VOLUME CODE=EBCDIC).
- At the job or JDE command level for job two, the recording code of the input data is Printable EBCDIC (PEBCDIC). The PDL command VOLUME CODE=PEBCDIC at the job level overrides both catalog and system level CODE parameters.

For JOB3, the recording code of the input data is EBCDIC, as specified at the catalog level. For JOB1, the recording code of the input data is ASCII, since neither the catalog nor job level overrides are coded and the system level command controls input.

Figure 1-4. Sample offline JSL

```

IBMPDL:  SYSTEM; /* JDL CODING /**COMMENTS ARE BOLDED**/* /
          /*****ID LEVEL COMMANDS*****/
VFU001: VFU          ASSIGN=(1,5), ASSIGN=(2,10), ASSIGN=(3,15),
                   TOF=5, BOF=66;
PDE1:    PDE          BEGIN = (175 DOTS, 300 XDOTS),
                   FONTS = (UN110E, PR110E, PR124B);
PDE2:    PDE          BEGIN = (0.725 IN, 1.000 IN),
                   FONTS = (L0512C, L05SCA, L05ITA),
                   PMODE = LANDSCAPE;
TAB1:    TABLE      CONSTANT = (A 'REPORT');
CRI1:    CRI          CONSTANT = (2,6, EQ, TAB1);
          /***** SYSTEM LEVEL COMMANDS*****/
          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;
          IDEN         PREFIX = A'DJDE', SKIP = 6, OFFSET = 2,
                   OPRINFO = YES;
          RSTACK       TEST = (CRI1), DELIMITER = YES;
          ACCT         USER=(BIN,TRAY);
          /***** CATALOG LEVEL COMMANDS *****/
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;
CATGRP:CATALOG;
          VOLUME      HOST=GRASP, CODE=EBCDIC;
          BLOCK        LENGTH=4096, PREAMBLE=0, ZERO=YES;
          RECORD       LENGTH=135, STRUCTURE=VB, PREAMBLE=1,
                   LTHFLD=1, FORMAT=BIN, OFFSET=0, ADJUST=2;
          /***** JOB OR JDE LEVEL COMMANDS *****/
          1:JDE INCLUDE=CATPOW;
                   VOLUME      HOST=POWERVS; CODE = PEBCDIC;
                   ABNORMAL ERROR = CONTINUE, IMISMATCH = CONTINUE;
          2:JDE INCLUDE=CATGRP;
          DFLT:JDE;
END;

```

Figure 1-5. Sample online JSL

```

JDLONL:  JDL;      /** JDL CODING /**COMMENTS ARE BOLDDED IN THIS
                   SAMPLE***/**/
                   /****** ID LEVEL COMMANDS *****/
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;
P1:      PDE       FONTS=(P0612B, 3), BEGIN=(2,1,5), PMODE=PORTRAIT;
P2:      PDE       FONTS=L0112A, BEGIN=(.55, .55);
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=(10)**';
C3:      CRITERIA CONSTANT=(0,10,EQ,T3),
                   LINENUM=(50,10);
                   /****** SYSTEM LEVEL COMMANDS *****/
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;
RSTACK    TEST=C1, DEL=YES;
          /****** JOB LEVEL COMMANDS *****/
DFLT: JOB; /* JOB WITH NO BANNER PAGES */
          IDEN PREFIX='$DJDE$', SKIP=8, OFFSET=1, OPRINFO=NO;
          OUTPUT FORMAT=FMT3, DUPLEX=YES, SHIFT=YES;
HDRP:JOB; /*JOB WITH HEADER PAGES ONLY */
          BANNER TEST=C1, HCOUNT=2 ,TCOUNT=0;
TRLP: JOB; /*JOBS WITH TRAILER PAGES ONLY*/
          BANNER TEST=C2, HCOUNT=0, TCOUNT=3;
BOTH: JOB; /* JOBS WITH BOTH HEADER AND TRAILER PAGES */
          BANNER TEST=(C1 OR C2), HCOUNT=2, TCOUNT=3;
          LINE  DATA=(0, 75), PCCTYPE=NONE, FONTINDEX=1;
END;

```

## PDL library

---

If you have multiple PDL commands of the same type, such as CMEs and PDEs, you may want to create separate files for them. These separate files allow you to group like specifications together and make your JSLs shorter, more efficient, and easier to modify and maintain.

When the JSLs are compiled, these library files create separate object files for your reports. PDL-related library files are .STK for paper stock specifications, .PDE for format, and .CME for Copy Modification Entry specifications. You can only call out these specifications as separate object files via DJDEs.

Avoid these categories of restricted file names to prevent possible errors:

- Command keywords such as OUTPUT, PDE, and RSTACK
- Parameter keywords such as LCODE, UCSB, and BFORM
- Abbreviated operator command keywords such as CAR (CARTRIDGE) and TAP (TAPE)
- DJDE keywords such as PMODE, RTEXT, and BATCH
- Resident task files such as .PDL, .FDL, .FNT, .LGO, .CME, .LIB, and .FRM
- System utilities such as INPUT, EDIT, DSR, IPD, and OCS.

---

## JDL creation

---

JSLs are the source files you create and then compile to create a JDL. There are many methods and elements you can use in creating a JDL. The following sections provide basic information and a typical sequence you can use in your JDL development.

---

### Required components of a JSL

---

A JSL may consist of numerous commands and command levels, but every set of PDL commands requires the following three elements in order to be compiled on the LPS:

- JDL coding—The first line of code in your JSL is a command to name the JDL. Each JDL must have a unique identifier. This JDL identifier compiles to become the external name of the JDL file which you then specify in order to run jobs contained within the JDL.
- JOB command—Within each JSL, there must be one JOB command (also referred to as a JDE or Job Descriptor Entry). The JOB command allows you to override the system level PDL commands on a job-by-job basis. This enables you to customize jobs independently and store them in a single PDL library.
- END command—Every JSL must conclude with an END command.

The syntax of these commands is provided in the “JSL structure” section earlier in this chapter. The system defaults apply to any print job characteristics that are not specified in the JSL.

---

### Steps in creating a JDL

---

If you want to create a JSL using specifications other than those defined in the defaults, you have many PDL commands available for your JSL and many ways of organizing them. The following procedure is one typical method of creating a JDL:

1. Identify the source media and data format.
2. Enter the editor task.

Refer to your *Xerox 4050/4090/4450/4650 LPS System Administration Guide* for instructions if you are not familiar with using the editor.

3. Code the JDL name.
4. Decide on the command levels you will use.
5. Code the data definition commands.
6. Design the layout of your print job.
7. Code the print format commands.
8. Build forms for the job, if necessary.

Refer to your *Xerox 4050/4090/4450/4650 LPS Forms Creation Guide* for information on creating forms.

9. Identify dynamic print requirements.

10. Code the IDEN command if you want to use Dynamic Job Descriptor Entries.
11. Identify logical processing requirements.
12. Code the logical processing commands.
13. Key in the END command to complete the JSL.
14. Compile the JSL to create a JDL file.

Keep the following in mind when coding your JSL:

- If you are not sure what specifications to select, try running the print job using the system defaults and then modify the JSL to meet your requirements. This is also helpful when you are modifying an existing JSL.
- If you do not know the source of data on a tape, run a data dump and match it to samples in the *Xerox Tape Formats Manual* to determine its origin.
- You may want to run the job after step 4 to see the general structure of the print job first and then make necessary adjustments. You may have to run the job several times during the development process to check your JSL specifications.
- You may want to design your form around the data placements on the page.
- It is sometimes helpful to sketch the page layout, including page orientation, logos, fonts, and the point of origin for printing on the page.
- Use tab spacing to create columns for the command identifiers, commands, and parameters in your JSLs. Tabs are not required but they make identifying these elements easier, thus reducing the chance of error. The JSLs in figures 1-4 and 1-5 are set up in column format with tabs. Generally, only identifiers and the END command begin in the first column of the JSL. Other commands are indented.

---

## JDL coding

---

After you determine your input source, code the JDL name, design the page layout, and decide on the command levels you are using, you are ready to begin coding the appropriate commands. This section explains the following three types of commands:

- Data definition
- Print format
- Logical processing.

The syntax, parameters, and options for each command are in the chapter corresponding to the command type.

This section also describes the use of DJDEs. DJDE syntax and parameters are in the "Print control (DJDE) commands" chapter.

For more information on graphics, and paper stocks and clusters, refer to the chapters of the same name.



---

## Data definition commands

---

As described in the “Input data streams” section in this chapter, there are a number of sources of input data streams. Every JSL must specify the data source and the data format. This is accomplished with the data definition commands.

To specify the following information, code the appropriate command, as shown in table 1-1.

Table 1-1. **Data definition command usage**

To specify	Use this command
Input data block characteristics	BLOCK
Input code translation table	CODE
Printer Carriage Control code table	PCC
Input data record characteristics	RECORD
Marked comparison type assignments	TCODE
Input medium characteristics	VOLUME

The following are the essential data definition commands:

- VOLUME command HOST parameter—Specifies the source of the input data stream.
- RECORD command—Specifies the format in which the data was recorded as well as the record length.
- BLOCK command—For offline data streams, specifies the maximum number of characters in a block.

Refer to the “Data definition commands” chapter for a complete description of all of these commands and their parameters.

---

## Print format commands

---

Print format commands specify the physical characteristics of a print job, such as collating, two-sided printing, fonts, forms, and accounting sheets. They also define the placement of data on the page and system responses to error conditions, among other functions.

To specify the following print format information, use the corresponding command, as shown in table 1-2.

Table 1-2. Print format command usage

To specify	Use this command
Error condition activities, which operator activities are permitted, and ink discrepancy controls	ABNORMAL
User and system accounting tasks	ACCT
Replacement of certain parts of report output on copies with predefined static data or font changes within variable data	CME
Margin control, overprinting, or carriage control	LINE
Messages sent to operators during job processing	MESSAGE
Print mode, forms, paper stock, offsetting, stapling, two-sided (duplex) or one-sided (simplex) printing, among other output characteristics	OUTPUT
Page format, orientation, and fonts	PDE
Messages and form for routing pages	ROUTE
Sets of stocks for a report	STOCKSET
Vertical format control	VFU

The first command coded with a JSL is typically the VFU command (with an identifier), which defines vertical tabs and is used in conjunction with the PCC command. Refer to figures 1-4 and 1-5 for examples. The LINE command references VFU specifications for use in the print jobs and tells the system what part of the data in each record is printed. For this reason, JSLs typically also include a LINE command.

Every JDL requires an OUTPUT command statement. Refer to the "Print format commands" chapter for a complete description of these commands and their parameters.

---

## Print control (DJDE) commands

---

Dynamic Job Descriptor Entries (DJDEs) are commands sent through the input data stream that override the printing characteristics specified in a Job Descriptor Entry (JDE). The IDEN command coded in a Job Descriptor Entry (JDE) notifies the system that DJDE records are included in the input data stream. The IDEN command is described in the "Print format commands" chapter. DJDE syntax and options are described in the "Print control (DJDE) command" chapter.

---

Some of the benefits of changing the job parameters with DJDEs are as follows:

- The printing system does not stop between reports, or 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 via DJDEs instead of being stored in the printing system via JDLs.
- Varying numbers of copies can be generated automatically, with routing or distribution notification sent to the operator.
- Unusual processing requirements may be satisfied through the use of DJDEs.

## DJDE orientation

---

There are two types of DJDEs: page-oriented and record-oriented. The orientation of the DJDEs is listed in the "Print control (DJDE) commands" chapter.

### Page-oriented DJDEs

---

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 on a copy-by-copy basis.

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 data on a page takes effect on the current page. A page-oriented DJDE positioned after the first print data on a page takes effect on the next page, unless end-of-report is encountered.

Use the page-oriented DJDEs described in table 1-3.

Table 1-3. Page-oriented DJDE usage

To specify	Use this command
Location of the starting print line of a logical page	BEGIN
The form printed on the back side of a printed page	BFORM
Placing pages of a job into sets	COLLATE
Number of copies or sets of pages to produce	COPIES
Accumulation of accounting statistics for reports on a department name basis	DEPT
Printing on both sides of a piece of paper	DUPLEX
Stock (type of paper) on which the page is printed	FEED
Location in the input record where an index to the specified font is stored	FONTINDEX
Fonts used in input data or variable (CME) data	FONTS
A new Page Descriptor Entry (PDE) to control formatting	FORMAT
The form merged on the printed page	FORMS
Text message displayed to operators during input processing	ITEXT
The JDE used within the selected JDL at the next page boundary	JDE
Name of the JDL invoked at the next page boundary	JDL
Left printing margin within each logical page	MARGIN
CME used on the next page	MODIFY
Page numbering control	NUMBER
Text message displayed to operators during job printing	OTEXT
Printing orientation for each page	PMODE
If a form is printed on all RTEXT pages	RFORM
Text printed on separate pages preceding a report	RTEXT
Image shift on the page for binding purposes	SHIFT
Repositioning of a new logical page to the first logical page of a specified side of a sheet of paper	SIDE
Redefinition of STOCKSET command parameters for the page	STOCKS

## Record-oriented DJDEs

Record-oriented DJDEs take effect immediately at the first record following the DJDE packet, that is, after an END command. These DJDEs also may appear at report boundaries to change all pages in a report on a copy-by-copy basis. Table 1-4 lists these DJDEs and their uses.

Table 1-4. **Record-oriented DJDE usage**

To specify	Use this command
Assign VFU channel to a page line number or set of line numbers	ASSIGN
Bottom-of-form line number	BOF
Comment text in the DJDE record	<i>C text</i>
Location and length of printable data within an input record	DATA
Enable files to load to system disk while print job in progress	FILE
Instruct the system when overprint lines occur	OVERPRINT
Top-of-form number	TOF

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.

Record-oriented DJDEs related to graphics are listed in table 1-5.

Table 1-5. **Graphics-related page-oriented DJDEs**

To specify	Use this command
New imaging parameters for graphics	ALTER
Normal processing on online banner pages for batch mode jobs	BATCH
Automatic reimaging (hold) of LOGO, IMAGE, or GRAPHICS DJDEs	CANCEL
DJDE is a graphic sentinel	GRAPHICS
In batch mode, define new imaging parameters for the graphic	IMAGE
.IMG files updating current report are not purged when report output processing is completed	SAVE

## DJDE record specification

---

The system looks for DJDEs in the input data stream only if a DJDE prefix has been specified by an IDEN command within the startup (START command) JDL used to process the job. DJDE information is contained in one or more data records, each of which may be up to the maximum record length specified in the startup JDL (or JDE). 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. Examples of DJDE records are illustrated in figures 1-6 and 1-7.

Consider the following when preparing DJDE records as part of the input data stream:

- 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.
- A DJDE record parameter that is split and continued on the next DJDE record should be terminated with a comma, semicolon sequence (,;).
- The prefix in the DJDE record may appear after DJDE parameters 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.
- For delimited logical 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 changed-based logical 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 DJDE packet 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.
- 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 specified channel, 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 specific parameters included in a DJDE packet are the only ones modified when the DJDE is applied. The only exceptions are the DJDEs JDE and JDL, 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, one of the following occurs:

- If in multireport mode or operating online, the next report begins with all of the parameters specified in the startup JDE or JDL (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 or JDL invoked by the new START command.

Using the DJDE JDE 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 JDE, 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 recognizes that 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 ASSIGN= to the DJDE packet to change the applicable channel, such as channel 1, to the new line number. Since the DJDE ASSIGN is line-oriented, it is implemented on the next line and will affect 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 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.

The following example of an IDEN command shows the multiple-record DJDEs specified in figure 1-6:

```
IDEN    PREFIX='MTEST', SKIP=7, OFFSET=1,
        OPRINFO=YES;
```

Figure 1-6. **Multiple record DJDE**

```
*MTEST C MULTI RECORD DJDE EXAMPLE:
*MTEST FORMS = (XEROX 1, 1, 3), FORMAT = XPDE 12, 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;
* Printer control byte
```

The following example of an IDEN command shows the single-record DJDE specified in figure 1-7:

```
IDEN    PREFIX='RTEST', SKIP=9, OFFSET=3,
        OPRINFO=YES;
```

Figure 1-7. **Single record DJDE**



---

## DJDE operator information pages

---

The changes to the JDE that are specified in the DJDE are incorporated when the system encounters the END parameter. 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 delayed.

The following message:

```
***MISSING END COMMAND OR MISSING PAGE BOUNDARY
```

is automatically printed on an OPRINFO page at the end of a report if the following are true:

- 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 definition of the input source cannot be changed with a DJDE. The system does not allow these basic parameters to be changed using the DJDEs JDE or JDL. The JDE specified by the DJDE JDE is referred to as a selected JDE.

Parameters which cannot be changed with a selected JDE or JDL are shown in table 1-6.

Table 1-6. **Parameters that cannot be changed via selected JDEs or JDLs**

Command	Parameter
BANNER	All
BLOCK	All
IDEN	All
OUTPUT	OFFSET
RECORD	All
VOLUME	HOST, LABEL, UNPACK

Do not attempt to change the online BANNER command offset criteria or the VOLUME command OPTIMIZE parameters in a selected JDE because unpredictable results may occur.

Selected JDEs or JDLs should specify all JDE parameters that stay the same as the original JDE, as well as the changes. JDE parameters that can be changed are shown in table 1-7.

Table 1-7. **Parameters that can be changed via selected JDEs or JDLs**

Command	Parameter
ABNORMAL	SECURITY, ERROR
ACCT	All
LINE	DATA, FONTINDEX, INKINDEX, MARGIN, OVERPRINT, PCCTYPE, VFU
MESSAGE	All
OUTPUT	All except OFFSET
PDE	All
RAUX	All
ROFFSET	All
ROUTE	All
RPAGE	All
RSTACK	TEST (online only)
VOLUME	TCODE, CODE

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 that 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.

---

## Logical processing commands

---

Logical processing commands enable you to specify special functions to be performed on a record, a set of records, or a block. The system determines whether or not to perform the function based on the tests and criteria you set up. The CRITERIA and TABLE identifier commands contain the parameters upon which the system makes the determination. CRITERIA and TABLE are usually ID level commands within the JSL.

After you specify the CRITERIA and TABLE parameters, you can reference these command identifiers in the TEST parameters of the logical processing commands. For example, in figure 1-4, CRI1 is referenced in the TEST parameter of the RSTACK command at the system level. In figure 1-4, criteria C1 is also referenced by the RSTACK command at the system level and then again in jobs HDRP and BOTH at the job level.

To specify the following logical processing characteristics, code the appropriate command, as shown in table 1-8.

Table 1-8. Logical processing command usage

To specify	Use this command
Banner page detection for the online mode only	BANNER
Block deletion for the offline mode only	BDELETE
Block selection for the offline mode only	BSELECT
Logical processing functions	CRITERIA
Select page from the auxiliary tray for offline and online modes	RAUX
Record deletion for the offline mode only	RDELETE
The clusters to be used when the given criteria are met	RFEED
Page offset in the stacker trays for the offline and online modes	ROFFSET
Logical page repositioning for the offline and online modes	RPAGE
Resumption of printing for the offline mode only	RRESUME
Record selection for the online and offline modes	RSELECT
End of report for the offline and online modes	RSTACK
Suspension of printing for the offline mode only	RSUSPEND
One or more constants for logical processing	TABLE

## Logical processing command format

To fully define a logical processing command, 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 *test-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 "ac:CRITERIA" command section of "Logical processing commands" chapter for more information.

---

## Logical processing commands with TEST parameters

---

Each logical processing command (except CRITERIA and RFEED) has a TEST parameter that 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.

---

### TEST expression definition

---

The syntax of the *test-expression* for a logical processing command can be one of the following:

TEST=(*cri-id-1*);

TEST=(*cri-id-1* AND, *cri-id-2*);

TEST=(*cri-id-1* OR, *cri-id-2*);

*cri-id-1* and *cri-id-2* are identifiers for either the change mode or the constant mode CRITERIA commands. If only *cri-id-1* is coded, the test is satisfied if the criteria in *cri-id-1* is satisfied. If *cri-id-1* and *cri-id-2* are both coded and the keyword AND is coded, the test is true only if the 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* is true. If the test is satisfied, the logical processing function is performed.

The RSTACK and BANNER test expression commands used in the startup JDE or JDL remain in effect throughout the entire job. They are not overridden by different specifications used in a selected JDL or JDE.

---

### String comparison concepts

---

String comparisons for logical processing are defined with the CRITERIA and TABLE commands. The TABLE command specifies one or more string constants. The CRITERIA command describes a field in a record and compares it to the values in the TABLE command (constant mode) or to the contents of the field in the previous record (change mode). Figure 1-8 illustrates string comparison parameters. These comparisons test for one of the following conditions:

- Absolute equality—Tested either by a change mode CRITERIA command or by a constant mode CRITERIA command, which references a TABLE command that does not have a MASK command coded.
- Equality under a mask—Tested when the referenced TABLE command has a MASK parameter coded.

Comparing strings for equality under a mask means that you specify that one of the following tests should be performed for each character position of the input data string:

- Compare the character in that position for absolute equality
- Ignore the character in that position (consider the character to compare as equal)
- Compare the character in that position for type (for example, alphabetic, numeric, or a specially defined type).

Figure 1-8. **String comparison parameters**

---

## Character types

---

An attribute called type can be associated with any character in a character set. The possible types are identified by the integers 1 to 7. It is possible for any character to be untyped, to have one type, or to have multiple types.

Defining of character types (the associating of type numbers with any group of characters in a character set) is described in the "ac:TCODE" command section in the "Data definition commands" chapter.

For every standard character set, there is a set of standard default type assignments which is sufficient for most applications without modification. These assignments define the number characters 0 to 9 to be type 1 characters and the lowercase and uppercase alphabetic characters (a through z and A through Z) to be type 2 characters. These default type assignments are 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 that the following commands be entered:

- TCODE parameter of the VOLUME command. This parameter allows you to select a set of standard default type assignments.

- MASK and CONSTANT parameters of the TABLE command. Together, these commands define exactly how the comparisons for a TABLE command are made using the set of type assignments specified by the TCODE parameter of the VOLUME command.

---

## JSL compilation

---

Now that you have all the necessary commands describing your input data and output data, and you have structured them in a complete JSL source file, you need to have the printing system compile your JSL to produce a JDL object file.

The command needed to compile your JSL is entered at the operator console on the Xerox LPS. The format of the PDL (compiler) command is as follows:

PDL *file-id*, TRAY

The *file-id* should be the same as the JDL-id within the file. The optional TRAY parameter is used to get the compiled listing of your code delivered to the sample tray on the printing system.

Refer to your *Xerox 4050/4090/4450/4650 LPS Command Reference* for a complete list of available PDL command options.

---

## Printing a job

---

Once your JSL compiles without error, you can tell the Xerox LPS to print your report by entering the START command at the operator console on your Xerox LPS.

The START command calls in your compiled Job Descriptor Library (JDL) and the Job Descriptor Entry (JOB) within the library in order to print a specific report. The format of the basic START command is as follows:

START *jde-id, jdl-id, [input-device]*

For example, to print the HDRP job in the JDLONL sample JSL in figure 1-5, you would compile the JSL and then enter the following START command:

START HDRP, JDLONL

Notice there are no semicolons after either the PDL or START command. These are operator commands and have different syntax than PDL commands.

Refer to your *Xerox 4050/4090/4450/4650 LPS Operator Guide* for a complete list of available START command options.





---

## 2. Data definition commands

This chapter describes the PDL commands that control and define input processing. The general functions of each command are described in this chapter.

---

### Input processing commands

---

The system input processor decodes and formats input data from an offline magnetic tape, a host-attached channel interface, a remote communication device, or an Ethernet interface. This section describes input processing commands.

---

#### BLOCK

---

Defines block length characteristics and recording modes for offline data sources.

---

##### ADJUST

---

Specifies a block length adjustment value that is added to or subtracted from the contents of the block length field to determine the true block length.

<b>Syntax</b>	<b>BLOCK ADJUST=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> Range for a value is -127 to +127. You must select a value less than the BLOCK LENGTH parameter. Use the plus (+) sign or minus (-) sign to specify a positive or negative adjustment.
<b>Default</b>	The default is 0.
<b>Example</b>	BLOCK ADJUST=-127
<b>Considerations</b>	Refer to the LENGTH parameter for more information.

---

##### CONSTANT

---

Specifies that the block delimiter string constant and all data following the constant are ignored until the end of the block.

<b>Syntax</b>	<b>BLOCK CONSTANT=</b> <i>sc</i>
<b>Parameter options</b>	<i>sc</i> A hexadecimal, octal, ASCII, EBCDIC, or character string constant. Specifies a length for the constant of one to four bytes.
<b>Default</b>	There is no default.
<b>Example</b>	BLOCK CONSTANT=-127

**Considerations** If a block delimiter constant is positioned and is part of a record or block, use caution. The system truncates the record or block. As a result, the data is not the format you specified.

---

**FORMAT**

---

Specifies the recording mode of the block length field.

**Syntax** **BLOCK FORMAT=mode**

**Parameter options**

*mode*

**BIN**

Refers to a binary recording mode.

**DEC**

Refers to a decimal recording mode.

**PACK**

Refers to a packed with no sign recording mode.

**PKSG**

Refers to a packed with sign recording mode.

**Default** The default is BIN.

**Considerations** You can override the values for FORMAT if RECORD STRUCTURE changes through ANSI, IBM/OS Standard, or Honeywell 2000 COBOL label processing.

---

**LENGTH**

---

Specifies the longest physical block processed.

**Syntax** **BLOCK LENGTH=value**

**Parameter options**

*value*

Specifies an integer from 12 to 24,576 bytes in length.

**Default** The default is 1,330 bytes.

**Example** **BLOCK LENGTH=15**

**Considerations** The maximum block size that the system can process is dependent on the available task memory and the invoked processing features. For offline processing, the tape label may override a coded LENGTH parameter, and is limited by the maximum block size (24,576 bytes). ANSI, IBM/OS Standard, or Honeywell 2000 COBOL labels that specify block length can override the LENGTH parameter. The length on a 4-by-3 packed format tape and Honeywell 600 is the number of 6-bit bytes or characters in the tape block.

---

**LMULT**

---

Specifies a multiplication factor applied to the contents of the block length field to determine the true block length.

**Syntax** **BLOCK LMULT=value**

**Parameter options**

*value*

Specifies an integer from 1 to 15 that is multiplied by the value in the length field you specify using the LENGTH parameter. This computes the number of bytes in a block.

**Default** The default is 1.

**Example** BLOCK LMULT=15  
**Considerations** Refer to the LENGTH parameter for more information.

---

### LTHFLD

Specifies the length of the field containing the block length.

**Syntax** **BLOCK LTHFLD=***value*  
**Parameter options** *value*  
 An integer from 0 to 5 that specifies in bytes the size of the length field.

**Default** The default is 0.

**Example** BLOCK LTHFLD=5  
**Considerations** If you set the size to zero, the block length field on the tape is the actual block length. The LTHFLD parameter may be overridden if RECORD STRUCTURE is changed through ANSI, IBM/OS Standard, or Honeywell 2000 COBOL label processing.

---

### OFFSET

Specifies the location of the block length offset.

**Syntax** **BLOCK OFFSET=***value*  
**Parameter options** *value*  
 Specifies an integer from 0 to LENGTH-LTHFLD-1. The offset is the number of bytes from the first byte of a block to the block length field.

**Default** The default is 0.

**Example** BLOCK OFFSET=2  
**Considerations** The values for OFFSET may be overridden if RECORD STRUCTURE is changed through ANSI, IBM/OS Standard, or Honeywell 2000 COBOL label processing.

---

### POSTAMBLE

Specifies the length in bytes of the data at the end of each tape block (byte offset from the end of a block to the end of the last logical record).

**Syntax** **BLOCK POSTAMBLE=***value*  
**Parameter options** *value*  
 Specifies an integer from 0 to the block length.

**Default** The default is 0.

**Example** BLOCK POSTAMBLE=351  
**Considerations** Do not code the length of the block delimiter constant as the BLOCK POSTAMBLE since both lengths are subtracted from the end of the block.  
 Refer to the OFFSET parameter for information on overriding this parameter.

**PREAMBLE**

---

Specifies the byte offset from the first byte of a tape block to the first byte of the first logical record.

**Syntax**

**BLOCK PREAMBLE=***value*

**Parameter options**

*value*

Specifies an integer from 0 to the block length.

**Default**

The default is 0.

**Example**

BLOCK PREAMBLE=8

**Considerations**

The PREAMBLE parameter may be overridden if RECORD STRUCTURE is changed through ANSI, IBM/OS Standard, or Honeywell 2000 COBOL label processing.

The search for the block delimiter constant starts after the BLOCK PREAMBLE and proceeds to the first appearance of the constant.

**ZERO**

---

Specifies the end of block indicator.

**Syntax**

**BLOCK ZERO=***value*

**Parameter options**

*value*

NO

Indicates that the value for the end of a tape block is not zero in the block length field.

YES

Indicates that the value for the end of a tape block is zero in the record length field before applying the record length adjustment. Data that follows the record is ignored through the end of the block.

**Default**

The default is NO.

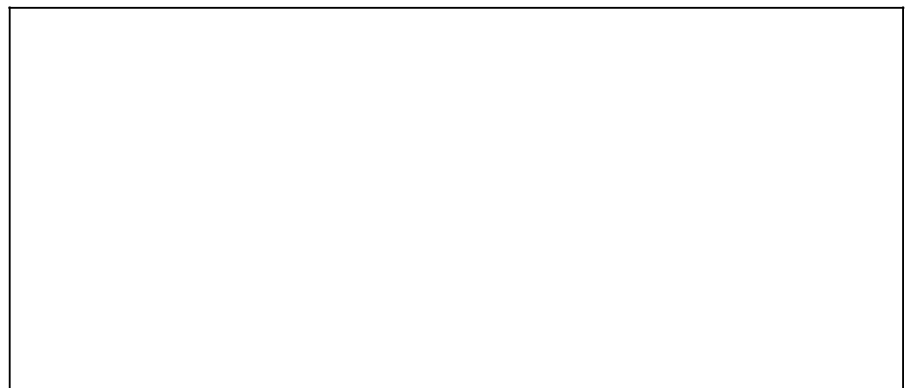
**BLOCK example**

---

Figure 2-1 shows an example of how you can code BLOCK.

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

Figure 2-1. **Sample BLOCK command diagram**



**ac:CODE**

Defines the input code of the translation table.

When you require a user-defined code translation table, you reference CODE by the CODE or LCODE parameters of VOLUME. The USER option of the CODE and LCODE parameters of VOLUME reference a user-defined code translation table in which no command identifier is coded.

A command identifier of the type *ac* is optional for the first CODE command within a JDE. Thereafter, each additional CODE command must include the *ac* identifier. The *ac* identifier consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). Make sure that one of the characters is a letter.

**ASSIGN**

Defines your code assignment exceptions or the entire user character translation table.

<b>Syntax</b>	<i>ac:CODE ASSIGN=(field1,(field2,...))</i>
<b>Parameter options</b>	<p><i>field1</i> Defines the input code.</p> <p><i>field2</i> Defines the output code, such as hexadecimal or decimal, that corresponds to the input code.</p>
<b>Default</b>	There is no default.
<b>Example</b>	<i>ac:CODE ASSIGN=(X'5B',(X'40',X'40',X'40'))</i>

**DEFAULT**

Specifies a base code from which you can make your code assignment exceptions. A code-type specifies the base code. Your exceptions are specified with the ASSIGN parameter.

<b>Syntax</b>	<i>ac:CODE DEFAULT=code-type</i> or <i>ac:CODE DEFAULT=value</i>
<b>Parameter options</b>	<p><i>code-type</i> ASCII BCD EBCDIC PEBCDIC H2BCD H6BCD IBMBCD</p> <p><i>value</i> A value is a one-byte hexadecimal, octal, or alpha constant. If you specify value, all inputs are coded to that specified value.</p>
<b>Default</b>	The default is EBCDIC.

<b>Example</b>	<p>ac:CODE DEFAULT=EBCDIC</p> <ul style="list-style-type: none"><li>Assume that your input tape is recorded in EBCDIC. On output, however, codes 5B, 5C, and 5D (characters \$ *) respectively, are assigned to the character blank (X'40'). The command to modify the EBCDIC base table is as follows: <pre>CODE1:CODE DEFAULT=EBCDIC, ASSIGN=(X'5B', X'40'), ASSIGN=(X'5C', X'40'),ASSIGN=(X'5D', X'40');</pre></li><li>In the following example, consecutive input codes such as X'5C' and X'5D' need not be specified to accomplish code modification. <pre>CODE2:CODE DEFAULT=EBCDIC, ASSIGN=(X'5B',X'40',X'40',X'40');</pre></li></ul>
<b>Considerations</b>	<p>You must code the DEFAULT parameter prior to any ASSIGN parameters for the assignment exceptions to take effect. A DEFAULT parameter following any defined corresponding ASSIGN parameter options causes this correspondence to be replaced by the DEFAULT parameter.</p>

---

## IDEN

---

Invokes DJDE processing. 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.

An IDEN command which is coded in a 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 that the system can identify and locate any DJDE records. The following is an example of a coded IDEN command:

```
IDEN PREFIX='RTEST',SKIP=9,OFFSET=3,OPRINFO=YES;
```

---

### OFFSET

---

Defines the starting position of the prefix string and DJDE parameters within the record.

<b>Syntax</b>	<b>IDEN</b> <i>OFFSET</i> = <i>value</i>
<b>Parameter options</b>	<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. You can use a negative number.
<b>Default</b>	The default is 0.
<b>Example</b>	IDEN OFFSET=12

**OPRINFO**

---

Specifies whether the DJDE record is printed and delivered to the tray.

<b>Syntax</b>	<b>IDEN</b> <i>OPRINFO=value</i>
<b>Parameter options</b>	<i>value</i> NO Specifies that the DJDE record does not print. YES Specifies that the DJDE record does print and is delivered to the tray.

**Default** The default is NO.

**Considerations** When you select OPRINFO=YES, the length of the DJDE record that prints on the sheet is the amount that fits on the width of the paper. Using OPRINFO=YES with portrait orientation may cause page setup errors if the DJDE record is too long.

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 xerographic mode for the previous page 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**

---

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, it examines the rest of the record for DJDE parameters.

<b>Syntax</b>	<b>IDEN</b> <i>PREFIX=sc</i>
<b>Parameter options</b>	<i>sc</i> Specifies the search criteria 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. Any records within the data stream that contain an identification field equal to the specified prefix string <i>sc</i> are recognized and processed as DJDE records.
<b>Default</b>	There is no default.
<b>Example</b>	IDEN PREFIX='MTEST'

**SKIP**

---

Defines the starting positions of the prefix string and DJDE parameters within the record.

<b>Syntax</b>	<b>IDEN SKIP=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies the offset to the starting column of the DJDE parameters. 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 DJDE parameters. You can use a negative number.
<b>Default</b>	The default is 1.
<b>Example</b>	IDEN SKIP=13

**ac:PCC**

---

Creates a table, or modifies an existing standard table of one-byte Printer Carriage Control codes and defines their action. Line spacing, skip-to-channel, and printing actions are all defined by this command.

A command identifier of the type *ac* is required for this command. The *ac* identifier consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). Make sure that one of the characters is alpha.

You can specify multiple user-defined PCC tables, but only one may be without a command identifier. The corresponding PCCTYPE parameter on the LINE command references each table with a command identifier. You can use the parameter keyword USER to reference any user-defined PCC table for which no command identifier is coded.

You can end a PCC command with a semicolon and start another PCC command to continue specification of the carriage control codes. You can use multiple PCC commands within a single PCC table definition as long as there are no intervening non-PCC commands.

**ADVTAPE**

---

Specifies whether the carriage control will advance to a new page when two successive channel-skip parameters are issued with no intervening print. For example, on most printers, the actions PSK1 (print-to-channel-1) followed by SK1N (skip-to-channel-1, do not print) would print a blank page. On a 1403 printer, these actions would not cause a blank page to print.

<b>Syntax</b>	<b>ac:PCC ADVTAPE=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> YES Specifies that multiple skips are allowed.  NO Specifies that multiple skips result in only one skip action by the carriage control.
<b>Default</b>	The default is YES.



**Considerations** SK1P (skip-to-channel-1) followed by a second SK1P results in a page transition since printing occurred on the first page, even if only blanks printed.

---

## ASSIGN

---

Specifies Printer Carriage Control exceptions to a default table.

**Syntax** *ac:PCC ASSIGN=(byte,action)*  
 or  
*ac:PCC ASSIGN=(byte,(action,space-print<sub>1</sub>,space-print<sub>2</sub>,spaceprint<sub>3</sub>))*

**Parameter options**

*byte*

Defines a Printer Carriage Control byte from 0 to 255 (X'00' to X'FF').

*action*

Specifies the action you need to take when the Printer Carriage Control *byte* causes the bottom-of-form (BOF) to occur:

TOF

Instructs the LPS to go to the top-of-form (TOF) on the next page and to stop spacing.

IGN

Instructs the LPS to ignore the BOF and continue spacing through the end of the physical page. The page then transitions to top-of-form and spacing continues.

OVR

Instructs the LPS to go to the top-of-form (TOF) on the next page and to continue spacing.

*space-print<sub>1</sub>*

Specifies the carriage action to be taken before printing. For example, ANSI carriage control.

SP*m*

Space *m* lines before printing.

SK*n*

Skip-to-channel-*n* before printing.

*space-print<sub>2</sub>*

Specifies whether printing occurs for this record.

P

Print the output data at the line number computed after *field 1* is processed.

N

No printing occurs for this record (default).

*space-print<sub>3</sub>*

Specifies the carriage action to be taken after printing. For example, IBM1403 carriage control.

SP*m*

Space *m* lines after printing.

SK*n*

Skip-to-channel-*n* after printing.

The range for the variables *m* and *n* is 0 to 15.

	Each <i>space-print</i> field is optional; however, you must specify at least one <i>space-print</i> parameter.
<b>Default</b>	The default is OVR.
<b>Example</b>	ac:PCC ASSIGN=(255,TOF)
<b>Considerations</b>	<p>You do not have to specify consecutive byte values. For example, you can code the following command:</p> <pre>ASSIGN=(X'60',SP1),ASSIGN=(X'61',SP2),ASSIGN=(X'62',SP3);</pre> <p>as a single command:</p> <pre>ASSIGN=(X'60',SP1,SP2,SP3);</pre> <p>If the LINE command specifies byte translation, 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'.</p>

---

**DEFAULT**

---

	Allows you to select a set of Printer Carriage Control codes. You may select a table and then modify specific control codes with the ASSIGN parameters.
<b>Syntax</b>	<i>ac:PCC DEFAULT=action</i> or <i>ac:PCC DEFAULT=pcctype</i>
<b>Parameter options</b>	<i>action</i> Specifies the action performed when a code has not been specifically assigned.  TOF Instructs the LPS to go to the top-of-form (TOF) on the next page and to stop spacing.  IGN Instructs the LPS to ignore bottom-of-form (BOF) and continue spacing through the end of the physical page. The page then transitions to TOF and spacing continues.  OVR Instructs the LPS to go to the TOF on the next page and to continue spacing.  <i>pcctype</i> ANSI B2500 B2700 B3500 B3700 B4700 B6700 H2000 H6000 IBM1401 IBM1403 US70 XEROX NONE

<b>Default</b>	The system default PCC tables are defined in the LPS-standard EBCDIC. The default is a table of PSP1 codes (print-and-space-1).
<b>Example</b>	ac:PCC DEFAULT=B4700
<b>Considerations</b>	The DEFAULT parameter must precede any ASSIGN parameter. Any preceding ASSIGN parameter is not incorporated into the PCC table.

---

## INITIAL

---

	Specifies the initial reference point (TOF or BOF) from which a report performs its first carriage control function.
<b>Syntax</b>	ac:PCC INITIAL= <i>reference point</i>
<b>Parameter options</b>	<p><i>reference point</i></p> <p>TOF Specifies the control program to perform the first spacing, skipping, or printing action from the top of the form.</p> <p>BOF Specifies the control program to perform the first spacing, skipping, or printing action from the bottom of the form.</p>
<b>Default</b>	The default is TOF.

---

## MASK

---

	Renders inaccessible unnecessary bits from the carriage control byte.
<b>Syntax</b>	ac:PCC MASK= <i>value</i>
<b>Parameter options</b>	<p><i>value</i></p> <p>Specifies an 8-bit value that is included with the Printer Carriage Control byte being processed through the AND option. AND occurs after translation. The result of this process is to mask off bits from the carriage control byte code that are not relevant to the specific operation.</p>
<b>Default</b>	The default is X'FF'.
<b>Example</b>	<p>In this example, X'40' (space-1-line-and-print), X'F1' (skip-to-channel-1-and-print), and X'F8 (print-and-skip-to-channel-8) are added to the standard IBM1403 carriage control table:</p> <pre>PCC1:PCC DEFAULT=IBM1403,INITIAL=TOF,ADVTAPE=NO, ASSIGN=(X'40',SP1P),ASSIGN=(X'F1',SK1P),ASSIGN=(X'F8', PSK8);</pre>

---

**RECORD**

---

Defines the characteristics of the offline input data record.

---

**ADJUST**

---

Specifies an adjustment value that you add or subtract from the contents of the record length field to determine the true record length.

<b>Syntax</b>	<b>RECORD ADJUST=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies the record adjustment length. The value is an integer from -127 to +127 that you add or subtract from the value in the length field of every record. Make sure the value is not greater than the record length field. You can make the first character plus (+) or minus (-).
<b>Default</b>	The default is 0.
<b>Example</b>	RECORD ADJUST=-127

---

**CONSTANT**

---

Specifies a constant string you use to signal the end of a record. The record delimiter constant string is not included in the print line.

<b>Syntax</b>	<b>RECORD CONSTANT=</b> <i>sc</i>
<b>Parameter options</b>	<i>sc</i> Specifies a hexadecimal, octal, or alpha constant. You can choose 1 to 4 bytes for the length of the constant.
<b>Default</b>	There is no default.
<b>Example</b>	RECORD CONSTANT=2
<b>Considerations</b>	If the STRUCTURE parameter changes to an undefined length (U) in label processing, the CONSTANT parameter may be enabled. Nevertheless, no definition is assumed for the constant string. The default must be 0 or it must be defined in the Job Descriptor Entry.

---

**FORMAT**

---

Specifies the format of the record length field.

<b>Syntax</b>	<b>RECORD FORMAT=</b> <i>format</i>
<b>Parameter options</b>	<i>format</i> BIN Refers to a binary recording mode. DEC Refers to a decimal recording mode. PACK Refers to a packed with no sign recording mode. PKSG Refers to a packed with sign recording mode.
<b>Default</b>	The default is BIN.

<b>Example</b>	RECORD FORMAT=PACK
	<b>LENGTH</b>
	Specifies a length of the longest logical record.
<b>Syntax</b>	<b>RECORD</b> <i>LENGTH</i> = <i>value</i>
<b>Parameter options</b>	<i>value</i> Online systems: the maximum value without print position indexing is 214 (optimize mode) or 2140 (non-optimize mode). Offline systems: the value is an integer from 1 to 310.
<b>Default</b>	The offline default is 133. The online default is 150.
<b>Example</b>	RECORD LENGTH=214
<b>Considerations</b>	The tape label contents may override a record length value. If you define a record length larger than the default block length (1330), you must specify a block length large enough to hold the record.  You can override the LENGTH parameter with ANSI, IBM OS/Standard, or Honeywell 2000 COBOL labels which specify record length.  The length on a 4-by-3 packed format tape is the number of 6-bit bytes or characters in the record.  When you change the DATA parameter of the LINE command to a value greater than the default value, you must also change the LENGTH parameter of the RECORD command accordingly.  You cannot use print position indexing for online processing if you specify a LENGTH parameter that is greater than 150.
	<b>LMULT</b>
	Specifies a multiplication factor applied to the contents of the record length field to determine the true record length.
<b>Syntax</b>	<b>RECORD</b> <i>LMULT</i> = <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies a multiplication factor that you multiply by the value in the length field to compute the number of bytes in the record. The value ranges from 1 to 15.
<b>Default</b>	The default is 1.
<b>Example</b>	RECORD LMULT=14

**LTHFLD**

---

Specifies the length of the field containing the record length.

<b>Syntax</b>	<b>RECORD</b> <i>LTHFLD=value</i>
<b>Parameter options</b>	<i>value</i> Specifies, in bytes, the record length field length. The value ranges from 0 to 5. If the size is set to 0, record lengths are not contained in the records, and the record length is the maximum LENGTH for each record.
<b>Default</b>	The default is 0.
<b>Example</b>	RECORD LTHFLD=4

**OFFSET**

---

Specifies location of the record length field.

<b>Syntax</b>	<b>RECORD</b> <i>OFFSET=value</i>
<b>Parameter options</b>	<i>value</i> Specifies the location of the record length field offset. This offset is a byte offset from the first byte of the record to the record length field. The value ranges from 0 to LENGTH-LTHFLD-1.
<b>Default</b>	The default is 0.
<b>Example</b>	RECORD OFFSET=2

**POSTAMBLE**

---

Specifies the length of any extraneous data at the end of the record.

<b>Syntax</b>	<b>RECORD</b> <i>POSTAMBLE=value</i>
<b>Parameter options</b>	<i>value</i> Specifies the length in bytes. The length is an integer from 0 to record length.
<b>Default</b>	The default is 0.
<b>Example</b>	RECORD POSTAMBLE=4

**PREAMBLE**

---

Specifies the offset to the user portion of the record. This is the record preamble length.

<b>Syntax</b>	<b>RECORD</b> <i>PREAMBLE=value</i>
<b>Parameter options</b>	<i>value</i> Specifies the offset from the first byte of the record to the first byte of the user's portion of the record. The length is an integer from 0 to record length.
<b>Default</b>	The default is 0.
<b>Example</b>	RECORD PREAMBLE=4

**STRUCTURE**

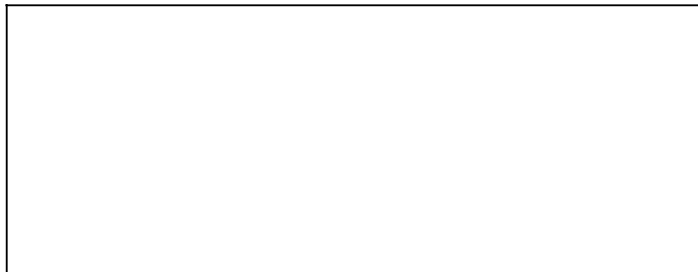
	Specifies the general structure of the input data.
<b>Syntax</b>	<b>RECORD STRUCTURE</b> = <i>structure-type</i>
<b>Parameter options</b>	<i>structure-type</i> FB (fixed length blocked) F (fixed length) V (variable length) VB (variable length blocked) U (undefined length) UB (undefined length blocked).
<b>Default</b>	The default is FB.
<b>Example</b>	RECORD STRUCTURE=FB
<b>Considerations</b>	For offline processing, the tape label contents may override this parameter.

**RECORD example**

A sample input record is shown in figure 2-3. A RECORD command for the sample input record is coded as follows and illustrated in figure 2-2:

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

Figure 2-2. **Record format types**



**Considerations** The system may override the values of LTHFLD, OFFSET, FORMAT, and PREAMBLE if ANSI, IBM OS/Standard, or Honeywell 2000 COBOL label processing has changed the STRUCTURE parameter.

Figure 2-3. **Sample input record**



---

**ac:TCODE**

---

Defines a set of input-character-to-type assignments for the masked string comparisons in a report. You must use TCODE when it is not possible to use one of the standard sets of default type assignments without modification. Code the TCODE command as follows:

- Use a command identifier of the type *ac* for the TCODE command within a JDE. The *ac* identifier consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter.
- 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).
- Use a hyphen ( - ) to specify a range of contiguous characters where the contiguity is based on actual codes.
- For input codes associated with ASCII characters, code the letter A hexadecimally as 41, the letter B as 42, and so forth up to the letter Z as 5A.
- For input codes associated with EBCDIC characters, be aware that the contiguity of the input codes is interrupted between the letters I and J, and between the letters R and S. This break requires you to specify three separate ranges.
- Use the TASSIGN parameters to define additional characters sets.

The LPS accepts lowercase characters expressed literally or symbolically. Determine whether the operator's console can process and display lowercase characters literally rather than symbolically.

Table 2-1 shows how you can code the TCODE command to specify particular character sets.

Table 2-1. Coding examples for the TCODE command

Type	Interpretation
1	Defines numeric characters 0 through 9 by default.
2	Defines alphabetic characters A through Z and a through z by default.
3	Defines uppercase alphabetic characters A through Z by TCODE command.
4	Defines lowercase alphabetic characters a through z by TCODE command.
None	Defines characters which are neither numeric or alphabetic.

**DEFAULT**

Specifies an initial set of character type assignments with masked comparisons. You can assign defaults for any standard character set by coding the appropriate code-type.

**Syntax** *ac:TCODE DEFAULT=code-type*  
or  
*ac:TCODE DEFAULT=value*

**Parameter options** *code-type*  
ASCII  
BCD  
EBCDIC  
PEBCDIC  
H2BCD  
H6BCD  
IBMBCD

*value*  
A value is an integer from 0 through 7. If you specify 0, all the characters are untyped. If you specify 1, all the characters are type 1. If you specify 7, all the characters are type 7, and so forth.

**Default** There is no default.

**Example** *ac:TCODE DEFAULT=BCD*

**Considerations** You can modify the initial set of type assignments by specifying the TASSIGN and TRESET parameters.

**TASSIGN**

Associates one or more specified characters with one or more specified code-types. The parameter also disassociates one or more characters from all code-type assignments. The TASSIGN parameter may be specified more than once in a TCODE command.

**Syntax** *ac:TCODE TASSIGN=(typespec, 'inputspec')*  
or  
*ac:TCODE TASSIGN=(typespec, ('inputspec<sub>1</sub>', 'inputspec<sub>n</sub>'))*

<b>Parameter options</b>	<p><i>typespec</i> Specifies a single integer or a series of integers. When coded as a single integer, the value range is 0 through 7. When coded as as a series of integers, the value range is 1 through 7.</p> <p>When a <i>typespec</i> is coded as a single integer, all specified characters are associated with the code-type.</p> <p>When a <i>typespec</i> is coded as a series of integers, the specified characters are each associated with all of the specified types. A <i>typespec</i> may also be specified with the following values:</p> <ul style="list-style-type: none"> <li>• NUMERIC instead of 1</li> <li>• ALPHA instead of 2.</li> </ul> <p>When a <i>typespec</i> is coded as 0, all characters specified by an <i>inputspec</i> are disassociated from all type code assignments. Thus, the TASSIGN parameter may untype a specified set of characters.</p> <p><i>'inputspec'</i> Specifies a single character, a series of characters, or a range of characters. The syntax indicates the set of input codes corresponding to characters bounded by and including the specified characters.</p>
<b>Default</b>	There is no default.
<b>Example</b>	ac:TCODE TASSIGN=(3,('A',-, 'J', 'S', -, 'Z'));

---

**TRESET**

	Disassociates one or more characters in a set from one or more types.
<b>Syntax</b>	<p>ac:TCODE TRESET=(<i>typespec</i>, '<i>inputspec</i>') or ac:TCODE TRESET=(<i>typespec</i>, ('<i>inputspec</i><sub>1</sub>', '<i>inputspec</i><sub>n</sub>')) or ac:TCODE TRESET=(<i>typespec</i>, ALL)</p>
<b>Parameter options</b>	<p><i>typespec</i> Specifies a single integer or series of integers from 1 through 7.</p> <p>When a <i>typespec</i> is coded as a single integer, all specified characters are associated with the type code.</p> <p>When a <i>typespec</i> is coded as a series of integers, the specified characters are each associated with all of the specified types. A <i>typespec</i> may also be specified with the following values:</p> <ul style="list-style-type: none"> <li>• NUMERIC instead of 1</li> <li>• ALPHA instead of 2.</li> </ul> <p><i>'inputspec'</i> Specifies a single character, a series of characters, or a range of characters. The syntax indicates the set of input codes corresponding to characters bounded by and including the specified characters.</p> <p>ALL Specifies that all characters in the code table are disassociated from the specified types.</p>

<b>Default</b>	There is no default.
<b>Considerations</b>	You can specify the TRESET parameter more than once in the TCODE command.

## VOLUME

---

Specifies the characteristics of the input medium.

A JDL is supplied with the LPS software to dump the contents of a tape. The following standard tape formats may be dumped with this JDL:

- ASCII
- EBCDIC
- H2000 BCD
- H6000 BCD
- IBM BCD
- UNIVAC FIELDATA.

Check the listing of the DUMP.JSL file to determine the appropriate JDE for the tape you want to dump. You can dump other tape formats by specifying the HOST type (DUMP or OCTDUMP) and the appropriate parameters, such as CODE and UNPACK, in a user-coded JDL.

### BMULT

---

Specifies a multiplication factor which you use to determine true block length.

**Syntax**  
**Parameter options**

**VOLUME** *BMULT=value*  
*value*

Specifies a multiplication factor which 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 to 15.

**Default**

The default is 1.

**Example**

VOLUME BMULT=13

### CODE

---

Specifies which code translation table the system uses to interpret the input data.

**Syntax**  
**Parameter options**

**VOLUME** *CODE=code-type*  
*code-type*

Specifies a specific, standard, system-defined code translation table. The options are:

- ASCII
- BCD
- EBCDIC
- H2BCD
- H6BCD
- IBMBCD
- PEBCDIC

**NONE**

Specifies that data is not translated by input processing.

**USER**

Refers to the single unlabeled CODE command defined within the JDE or JDL.

*id*

Specifies a variable identifier you use in the CODE command to refer to a specific user-defined code translation table. The identifier *id* is required when multiple CODE commands are contained within the JDL.

**Default**

The default is EBCDIC.

**Example**

VOLUME CODE=BCD

**Considerations**

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 *id* to correspond to the identifier label assigned to the defining CODE command.

If the system reports missing fonts when running XPPI jobs with CODE=NONE specified, the operator can elect to continue by pressing <CON>. 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 inches.

Input data is translated according to the CODE parameter 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.

**EOV**

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

**Syntax**

**VOLUME** *EOV*=(*action*<sub>1</sub>,*action*<sub>2</sub>)

**Parameter options***action*<sub>1</sub>**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.

*action*<sub>2</sub>**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 multi-reel tapes is required to avoid these page format problems.

## NOEOF

Specifies normal end-of-volume processing.

**Default** The default is NOPAUSE,NOEOF.

**Example** VOLUME EOV=(PAUSE,EOF)

---

**HOST**

---

Specifies the computer or host operating system that generated the input data file.

**Syntax** VOLUME *HOST*=*host-type*

**Parameter options**

*host-type*  
ANSI  
B2500  
B2700  
B3500  
B3700  
B4700  
B6700  
DEC  
PDP11  
DUMP  
GRASP  
H2000  
H6000  
IBMOS  
IBMDOS  
IBMONL  
ICL2900  
NCR  
OCTDUMP  
OLDUMP  
OSWTR  
POWER  
POWERVS  
RSX11  
UNDEF  
UNIVAC 1100  
US70  
XEROX  
ENET

**Default** The default is IBMOS.

**Example** VOLUME HOST=ANSI

**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 has changed the system default values, the job may print incorrectly. If this happens, separate and run online and offline jobs independently from each other.

## INTERPRESS

Provides additional control over the Interpress environment.

### Syntax

**VOLUME** *INTERPRESS= YES, error checking, pagebreak, thrupt, scale, RES, image*

### Parameter options

#### YES

Specifies that the data stream is Interpress. NO is not an option.

#### *error checking*

##### 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.

#### *pagebreak*

##### BREAKPAGE

Requests a trailing break page if printing instructions are present in the job.

##### NOBREAKPAGE

Specifies a break page only if the system detects errors.

#### *thruput*

##### ROBUST

Invokes a process which supports more Interpress structures if the Interpress data stream is complex. Most Interpress masters are in final form, making ROBUST unnecessary.

##### SIMPLE

Provides increased performance if the Interpress master is not complex.

#### *scale*

##### INTEGRAL

Requests integral scale factors from 1 to 8 for IMG, IM6, and most RES (Xerox Compressed) images.

##### FRACTIONAL

Specifies fractional scaling of images with the Raster Image Processor (RIP) option when selected through JSL.

#### RES

##### PERFORMANCE

Specifies, with the Raster Imager Processor (RIP) that Raster Encoding Standard (RES) files that contain sampled color are imaged by the graphics hardware whenever possible. The RIP is used only when graphics hardware capabilities are exceeded.

##### COMPLIANCE

Specifies, with the Raster Imager Processor (RIP) that all RES files that contain sampled color and *priorityImportant* bitmap graphics are imaged on the RIP.

*image*

## RIP

Instructs the system, with the Raster Image Processor (RIP) option, to use RIP to image the job.

*NORIP*

Instructs the system not to use the Raster Image Processor (RIP) even if it is present.

**Default** The defaults are: YES, NOCHECKSUM, NOBREAKPAGE, SIMPLE, INTEGRAL, PERFORMANCE, and RIP.

**LABEL**

---

Specifies the type of label on the input tape.

**Syntax** **VOLUME LABEL=***label-type*

**Parameter options***label-type*

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).

**Default** The default is STANDARD.

**Example** VOLUME LABEL=ANSI

**LCODE**

---

Specifies the code translation table to use in interpreting the tape label.

**Syntax** **VOLUME LCODE=***code-type*

**Parameter options***code-type*

ASCII

BCD

EBCDIC

H2BC

H6BCD

IBMBCD

PEBCDIC

NONE

Specifies that data is not to be translated by input processing.

## USER ID

Specifies a variable identifier you use in the CODE command to refer to a specific user-defined code translation table. The identifier id is required when multiple CODE commands are contained within the JDL.

**Default** The default is EBCDIC.

**Example** VOLUME LCODE=ASCII

**Considerations** You select NONE for XPPI jobs that contain metacodes.



**LPACK**

---

Specifies whether undefined labels are packed or unpacked.

<b>Syntax</b>	<b>VOLUME LPACK=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> NO Indicates that no unpacking operation is to be performed. YES Indicates that the labels are packed.
<b>Default</b>	The default is NO.
<b>Considerations</b>	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.

**MAXLAB**

---

Specifies the longest physical block that is treated as a label for undefined labels.

<b>Syntax</b>	<b>VOLUME MAXLAB=</b> <i>value</i>
<b>Parameter options</b>	<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.
<b>Default</b>	The default is 81.
<b>Example</b>	VOLUME MAXLAB=2
<b>Considerations</b>	Any data block whose length is between MINLAB and MAXLAB (inclusive) is treated as a label.

**MINLAB**

---

Specifies the smallest physical block that is treated as a label for undefined labels.

<b>Syntax</b>	<b>VOLUME MINLAB=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies in bytes the maximum length of the smallest physical block. The value can range from 1 to 4095 bytes. Make sure that the value you specify is less than the MAXLAB value.
<b>Default</b>	The default is 80.
<b>Example</b>	VOLUME MINLAB=2

**OPTIMIZE**

---

Enhances the performance of the data stream.

**Syntax** **VOLUME OPTIMIZE=***value*  
or  
**VOLUME OPTIMIZE=(***value,...***)**  
or  
**VOLUME OPTIMIZE=NONE**

**Parameter options**

*value*

**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.

**NONE**

Does not invoke the optimization parameter.

**Default**

The default is NONE.

**Example**

**VOLUME OPTIMIZE=(NCC,NDC,NPRE)**

**Considerations**

Maximum performance occurs when all the parameters are selected, but there is still significant improvement when you specify only NCCHECK and NDCHECK because input processing of a record and the subsequent data transfer of the next record occurs immediately.

**OSCHN**

---

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

**Syntax** **VOLUME OSCHN=***value*

**Parameter options**

*value*

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.

**Default**

The default is 9.

**OSHDP**

---

Specifies the number of header banner pages generated by an IBM OS Writer.

- Syntax** **VOLUME** *OSHDP=value*
- Options** *value*  
Specifies the number of header pages that precede the report.
- Default** The default is 0.

**PLABEL**

---

Specifies whether the tape labels are printed.

- Syntax** **VOLUME** *PLABEL=value*
- Parameter options** *value*  
NO  
Specifies that no tape label printing results.  
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.
- Default** The default is NO.

**RMULT**

---

Specifies a multiplication factor to apply to the record length extracted from a tape label to determine the true record length.

- Syntax** **VOLUME** *RMULT=value*
- Parameter options** *value*  
Specifies an integer from 1 to 15.
- Default** The default is 1.

**RSAT**

---

Specifies whether a complete pass is made 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=action*
- Parameter options** *action*  
REMOUNT  
Completes report copies. For each report copy, it is necessary to remount all of the tapes.  
SPLIT  
Indicates that only one pass is made over the input data and that the reports are in sections and require manual merging. Input processing is faster for SPLIT since only one pass is required over the input data.
- Default** The default is REMOUNT.

**Considerations** 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**

---

Specifies the set of type assignments to use for masked comparisons.

**Syntax** **VOLUME** *TCODE=code-type*  
or  
**VOLUME** *TCODE=id*

**Parameter options** *code-type*  
ASCII  
BCD  
EBCDIC  
PEBCDIC  
H2BCD  
H6BCD  
IBMCD.

By coding any one of the *code-types*, you are furnished with a set of character-to-type assignments.

*id*  
Specifies an identified label of a TCODE parameter that defines a set of type assignments.

**Default** The default is EBCDIC.

---

**UNPACK**

---

Specifies the unpacking method of the input data when required.

**Syntax** **VOLUME** *UNPACK=value*

**Parameter options** *value*  
NONE  
Does not perform the unpacking operation.  
T4X3  
Specifies unpacking routines in which 6-bit characters are extracted and restored as 8-bit bytes. T4X3 is used primarily for Honeywell 6000 tapes.  
T4X3H2  
Specifies unpacking routines in which 6-bit characters are extracted and restored as 8-bit bytes. 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.

**Default** The default is NONE.

**VCODE (XDDI)**

Specifies a table of character type codes to use in VALUE mode CRITERIA testing. This option accommodates the different presentations of numbers used in different countries, thereby ensuring accurate comparisons of numeric values. For more information on the PDL CRITERIA command, refer to the "Logical processing commands" chapter.

**Syntax**

**VOLUME** *VCODE*=(*translation-type*,*keyword*)

**Parameter options**

*translation-type*

Specifies a standard code translation set. The following options are available:

ASCII  
BCD  
EBCDIC  
H2BCD  
H6BCD  
IBMBCD  
PEBCDIC.

*keyword*

Specifies a standard character type code table. The options for this parameter are shown in table 2-2.

Table 2-2. **Standard character type code options**

Type	Thousands separator	Decimal point
VCODE0	Comma	Period
VCODE1	Period	Comma
VCODE2	Comma, space, apostrophe	Period
VCODE3	Period, space, apostrophe	Comma
VCODE4	Space	Period
VCODE5	Space	Comma
VCODE6	Period, space	Dollar sign
VCODE7	Comma, space, apostrophe	Period

**Default**

The default is EBCDIC,VCODE0.

**Examples**

The following command utilizes the character type code table applicable to the United Kingdom or the United States to process and compare numeric values through an ASCII code translation:

VOLUME CODE=ASCII,VCODE=(ASCII,VCODE0);

The following command utilizes the character type code table applicable to France to process and compare numeric values through an EBCDIC code translation:

VOLUME CODE=EBCDIC,VCODE=(EBCDIC,VCODE1);

The following are examples of the value 50,000 which can be interpreted by various VCODEs:

VCODE0 50000.00 or 50,000.00  
VCODE1 50000.00 or 50,000.00  
VCODE2 50000.00 or 50,000.00 or 50 000.00 or 50'000.00  
VCODE3 50000.00 or 50.000,00 or 50 000,00 or 50'000,00  
VCODE4 500000.00 or 50 000.00  
VCODE5 500000,00 or 50 000,00  
VCODE6 50000\$00 or 50.000\$00 or 50 000.00 or 50'000.00  
VCODE7 50000.00 or 50,000.00 or 50 000.00 or 50'000.00

The following are examples of the value PI which can be interpreted by various VCODEs:

VCODE0 3.141592654 or 3.141,592,654  
VCODE1 3,141592654 or 3,141.592.654  
VCODE2 3.141592654 or 3.141,592,654 or 3.141 592 654 or  
3.141'592'654  
VCODE3 3,141592654 or 3,141.592.654 3,141 592 654  
3,141'592'654  
VCODE4 3.141592654 or 3.141 592 654  
VCODE5 3,141592654 or 3,141 592 654  
VCODE6 3\$141592654 or 3\$141.592.654 or 3,141 592 654  
VCODE7 3.141592654 or 3.141,592,654 or 3.141 592 654 or  
3.141'592'65

## VOLUME example

---

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;

---

## 3. Print format commands

This chapter describes the PDL commands that control the print format of your job.

---

### Print format command functions

---

Print format commands serve many different functions. The following are the most common uses of print format commands:

- Specify the physical characteristics of a print job
- Define the placement of data on the page
- Identify system responses to error conditions.

---

#### ABNORMAL

---

Allows you to restrict certain operator functions and define system responses to error conditions.

---

#### ERROR

---

Defines the system response to abnormal conditions detected while processing or compiling DJDEs.

**Syntax**  
**Parameter options**

**ABNORMAL** *ERROR=action*

*action*

STOP

Stops processing the report. The following message is displayed and is followed by a DJDE error message:

```
OS2006 ENTER CONTINUE I OR ABORT I
```

ABORT

Displays a message for the abnormal condition. Aborts the report it is processing, not the entire job. Does not give you the option of continuing the job.

CONTINUE

Displays a message defining the abnormal condition. Continues processing the job without applying the offending DJDE parameter. Does not enable you to abort the job.

**Default**  
**Considerations**

The default is STOP.

If the system detects abnormal conditions while processing or compiling DJDEs, it sets up the DJDE parameter set containing the offending entry to print on the OPRINFO page and displays an error message. The system does not report errors detected while processing RTEXT. It does report graphics processing separately on the graphics exception page printed at the end of the report.

There are two ERROR parameter modes:

- Non-stop includes CONTINUE and ABORT
- Stop mode includes STOP.

In either mode, the appropriate message displays when any abnormal conditions, such as the following, are displayed:

- Syntax error in DJDE specifications
- Missing file referred to by a DJDE
- RTEXT error
- GRAPHIC error.

---

### OTEXT

Specifies whether the system stops or continues printing when it displays an OTEXT message during system recovery.

**Syntax**  
**Parameter options**

**ABNORMAL OTEXT=action**

*action*

NOWAIT

Displays the OTEXT message as the marker page prints without interrupting printing.

WAIT

Stops printing and displays the OTEXT message when a recovery marker page prints. This delay allows you to verify the output report in the vicinity of the marker page.

**Default**

The default is NOWAIT.

**Example**

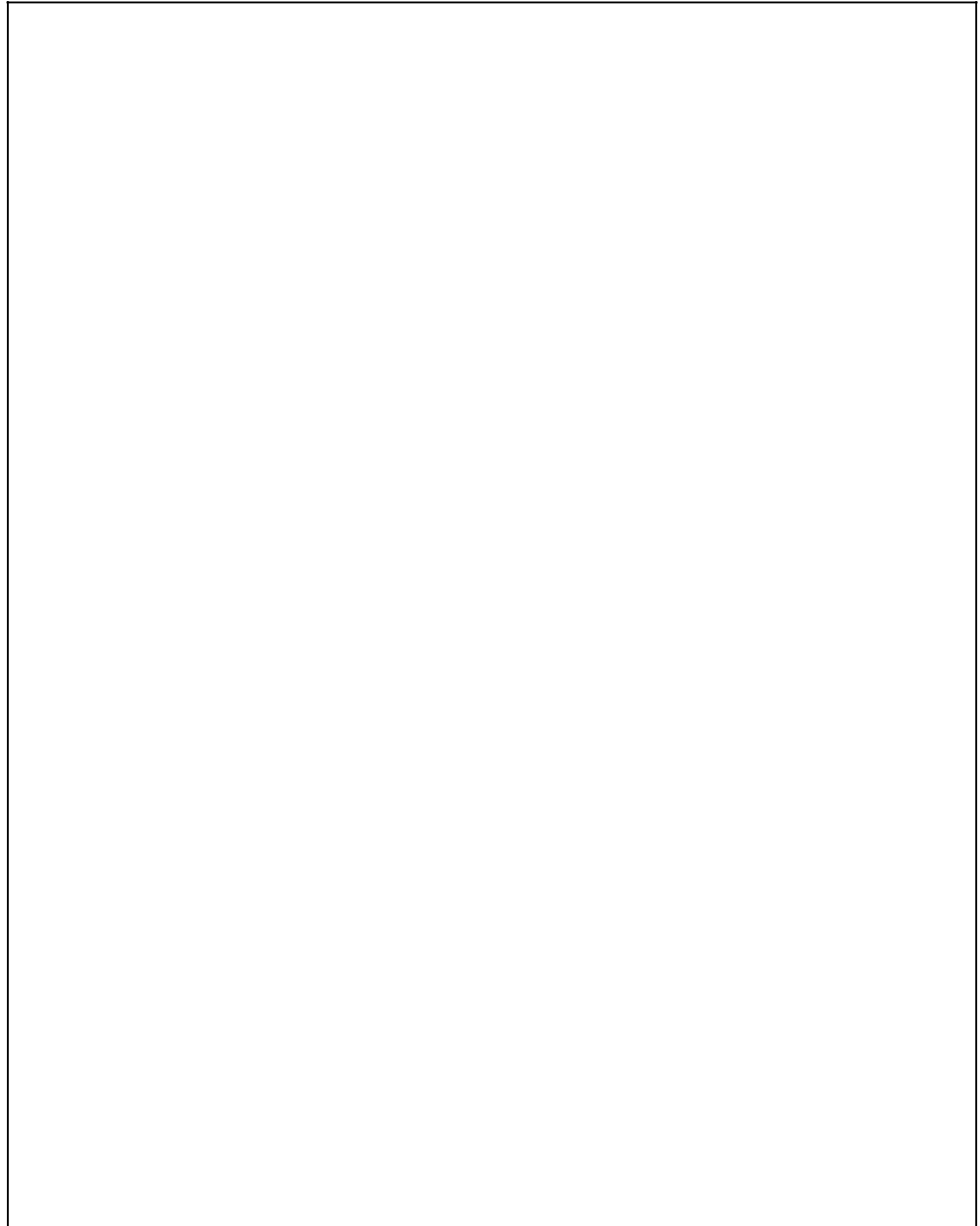
ABNORMAL OTEXT=WAIT,  
ERROR=ABORT;

**Considerations**

LPS automatic recovery procedures are based on valid checkpoint data and usually require no operator intervention. The system produces a marker page (shown in figure 3-1) and inserts it into the print job at the point where the crash occurred. It prints after the data that was being processed just before the crash occurred.



Figure 3-1. Sample marker page




---

**SECURITY**

Restricts certain operator functions.

**Syntax** **ABNORMAL SECURITY=***value*

**Parameter options**

*value*

NO

Specifies that no restrictions are in effect.

YES

Restricts the following commands for the duration of the startup JDE or JDL: SAMPLE (sample print), MOVE BLOCKS (block spacing of tape input), and SPACE PAGES (skipping over output pages).

**Default** The default is NO.

**Considerations** 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).

## ACCT

---

Allows you to track system print accounting statistics. Accounting information is automatically accumulated by the system on a report basis and also for overall system usage.

The ACCT command enables you to print an accounting page at the end of printing for each report. An accounting page is a summary that consists of a single page of information containing job setup information and counts of processing events.

If a job is aborted during input, the system prints an accounting page with as much accounting information as possible and delivers one copy to the sample tray and one to the output tray.

### DEPT

---

Establishes the name under which accounting statistics are accumulated for each report.

**Syntax**  
**Parameter options**

**ACCT DEPT=***sc*

*sc*

A string constant of up to 31 characters representing a department code or a name under which accounting information is maintained. In addition, the semicolon character (;) is also allowed for the department code. You must create the *sc* option name in the accounting file before you process a job.

**Default**

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.

**Example**

The following example illustrates the use of the PDL command ACCT and the system level ACCOUNT command. When the system level command 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.

This command:

```
ACCOUNT USER=TRAY, DEPT='OPERATIONS';
```

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.

**Considerations**

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 five duplex pages.

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 to the department name as specified on the ACCT command. The NODEPT entry must be deleted before any accounting data has been cleared.

## USER

Specifies the destination of the one-page accounting summary for each report. This page is always printed in the mode of the report.

### Syntax Parameter options

**ACCT** *USER=destination*

*destination*

**BIN**

Directs an accounting page to the output tray.

**BOTH**

Directs an accounting page to the output tray and the sample print tray.

**NONE**

Specifies that no accounting page is delivered.

**TRAY**

Directs an accounting page to the sample print tray.

### Default

The default is BIN.

### Considerations

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.

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 you of a possible report integrity problem. If the report is duplex and the first integrity problem occurs within the last five back sides, the report completion code on the forced accounting page is zero.

Jobs aborted by input as a result of job recovery still 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.

The REPORT USER system command is used to print a copy of accounting statistics. It prints out accounting data for each name entered into the accounting file by the ACCOUNT parameter.

DEPARTMENT or JDL entries are supplied by OSS (NODEPT and TASKS\$:JDL) and entered 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 or JDL entry.

Other entries, such as PAGES and LINES, are defined the same as those on the end-of-report accounting page.

## ACCT example

---

ACCT DEPT='TURTLE', USER=(BIN, TRAY);

After each report, one accounting page is sent to the output tray and one is sent to the sample tray. Accounting information for each job run under the ACCT command is accumulated in the system accounting log under the department name TURTLE. This assumes that you have entered the name TURTLE into the list of accounting names with the system level ACCOUNT parameter.

## ac:CME

---

Specifies font changes within variable data or to allow predefined static data to replace certain parts of report output on copies.

The copy modification feature (also referred to as spot-carbon) offers the ability to modify LPS output. It 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.

This feature is controlled through the use of the Copy Modification Entry (CME) command. The CME command defines 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.

An identifier of the type *ac* is specified when defining the CME. The command identifier *ac* may consist of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter. You reference *ac* with the MODIFY parameter of the OUTPUT command. The CME command must precede the OUTPUT command.

You may enter CME specifications in the short form to minimize the coding length. In short form, you give the first character of a parameter keyword, omit equal signs, and omit commas except where needed to avoid ambiguity.

CMEs are grouped into separate system disk command files of the type *xxxx.CME* instead of coded in the JSL. You reference CMEs with the OUTPUT command MODIFY parameter keyword.

## CONSTANT

---

Specifies the character string that is printed.

**Syntax**  
**Parameter options**

*ac:CME* *CONSTANT*=*sc*

*sc*

Represents the string constant to be printed. The width of the copy modification rectangle is determined by the number of characters specified by the *sc* option. More than one *sc* is allowed.

**Default**

There is no default for the CONSTANT parameter of the CME command. It must be specified unless all that is required is a font change in the variable data.

- Considerations** 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 the # are considered lowercase until another # is encountered. The sequence ## indicates that the character # is to be inserted and is not treated as toggling lowercase mode.
- 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.
- Whether an ink switch is applied to a CME string constant depends upon the order in which INK and CONSTANT are specified in the CME. If INK is specified before CONSTANT, the ink switch applies to the string constant. If INK is specified after CONSTANT, the ink 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 POS and CONSTANT parameters are specified in the CME. If POS is specified before CONSTANT, the string constant is printed at the specified position. If POS is specified after CONSTANT, the string constant is printed at the current position (position 1 if no other POS or CONSTANT parameter has been specified), and a subsequent CONSTANT or FONT parameter will take effect at the specified POS.
- When the host type is RSX11, CMEs are often applied twice per line because of the way the carriage return and 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.

---

## FONT

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 FONT=value*
- Parameter options** *value*  
 Ranges from 1 to *n*, where *n* 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.
- Default** The default is the current font (specified with the last FONT parameter; initially the value is 1).
- Considerations** A font specification applies to input variable data as well as static CME data. If a line number (LINE) and character position (POS) but no insertion text (CONSTANT) are specified, the font change specified applies to input variable data at the position specified.
- 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.

CMEs are not applied if OVERPRINT=MERGE and FONTINDEX are both specified.

**LINE**

---

Specifies the line range of the CME.

**Syntax** *ac:CME LINE=*n**  
 or  
*ac:CME LINE=(*n,m*)*

**Parameter options**

*n* The initial line number of the copy modification rectangle.

*m* 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 the *n* option.

**Default** There is no default.

**Considerations** LINE must always precede POSITION.

CME parameters are applied to all lines within the range whether or not variable data appears on those lines.

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;
```

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.

---

**POSITION**

---

	Specifies the initial character position of the CME in the print line.
<b>Syntax</b>	<i>ac:CME POSITION=value</i>
<b>Parameter options</b>	<i>value</i> An integer value between 1 (the first position of the print line) and the value specified by the length parameter option (right hand part) of the DATA= parameter keyword in the LINE command.
<b>Default</b>	The default is 1.
<b>Considerations</b>	The LINE parameter must always precede the POSITION parameter.

---

**EXPORT (XDDI only)**

---

A segment is a set of consecutively delivered sheets whose management as a set or segment has been specified by a PDL or DJDE command.

The EXPORT command enables specification of the following segment management capabilities:

- Division of reports into segments
- Generation of segment separator sheets
- Use of segment numbers on separator sheets to simplify the assembly of reports after finishing
- Segment recovery (from system errors or printer faults) to simplify assembly of reports after finishing.

Although the EXPORT command is primarily intended for use with the bypass transport option, it functions with any allowable destination (output device). If the OUTPUT DESTINATION specification is BIN, segments are offset relative to each other when delivered, and no other offsetting is processed.

**Interactions between EXPORT PDL statements and EXPORT DJDEs**

Take note of the following interactions between EXPORT PDL statements and EXPORT DJDEs and their effect on segment management.

If you specify a line and column number value for SNUMBER in PDL that causes the segment number string to print off the page, the segment number does not print. If the system encounters an invalid line or column specification in the PDL command and a valid specification in a DJDE, unpredictable results may occur.

If the system encounters a SPLIT=OFF DJDE, it disables segment management, including:

- SRECOVER defaults to PAGE recovery
- SEPARATORS defaults to NONE
- Segment numbering resets to PART 001 (which takes effect when segment numbering is next enabled).

You must use a DJDE SPLIT=(*min,max*) to reenale segment management processing after it has been disabled by the DJDE SPLIT=OFF command.

The system may end a segment under any of the following conditions:

- The maximum allowable page value is reached.
- The end-of-report condition is met, including the logical end-of-report specified in the RSTACK command.
- The end-of-data condition is met.
- SEPARATORS is specified in a DJDE.
- The system encounters a SPLIT=NOW DJDE.
- The system encounters a SPLIT=OFF DJDE.

You can reset the segment number to 1 by entering a SPLIT=OFF DJDE followed by a SPLIT=(*min,max*) DJDE.

Banner pages in an online environment are normally routed to the same output destination as the current job. You can redirect the banner page using the BANNER command, described in the "Logical processing commands" chapter.

If you have selected the SPLIT option for the RSAT command, a disk saturation condition may cause reports to be segmented incorrectly.

## SEPARATORS

---

Specifies that the printer software saves an internal copy of the first data page of the report and uses it to generate segment separator sheets.

**Syntax**

**EXPORT SEPARATORS=***value*  
 or  
**EXPORT SEPARATORS=(***BOTH,D***)**  
 or  
**EXPORT SEPARATORS=(***LAST,D***)**

**Parameter options**

*value*

**FIRST**

Specifies that the separator sheet be the first sheet of the second and subsequent segments (no sheet is generated for the first segment).

**LAST**

Specifies that the separator sheet be the last sheet of all segments. In an online environment, a true banner trailer page is used as the separator sheet in place of the saved separator page, if one has been specified.

**BOTH**

Specifies that the system prints a separator sheet both as the first and the last page of each segment.

**NONE**

Does not print a separator sheet.

**D**

Specifies that the system prints the separator sheet on the back of any LAST sheets. If the report is a simplex report, the system still processes the last page of the segment as a duplex page. The parentheses are required when using the D parameter option.

**Default**

The default is NONE.



---

## SNUMBER

Specifies that the system merges the current segment sequence number (beginning with 001) into any separator sheets produced.

The printed segment sequence number for all separators except a LAST separator in the last sequence is eight characters long. It has the following format:

PART *nnn*

The segment number string on a LAST separator in the last segment is fifteen characters long with the following format:

PART *nnn of nnn*

**Syntax** EXPORT *SNUMBER*=(*Inum*, *cnum*, *findex*)

**Parameter options** *Inum*

The line number on which the segment number string is to begin (integers only).

*cnum*

The column number on which the segment number string is to end (integers only).

*findex*

The font index of the font to use in printing the segment number string. If the font index is not specified, the system uses the first font in the font list of the separator page.

**Considerations** The system interprets the line and column numbers entered in the command using the metrics from the font index specified, or if none is specified, from the font used as a default.

If the line or column number you specify causes the segment number to print off the page, the segment number does not print. If the system encounters an invalid line or column specification in the PDL command and a valid specification in a DJDE, unpredictable results may occur.

---

## SPLIT

Specifies using segment management and specifies the conditions for ending segments.

You must use the EXPORT SPLIT parameter for other EXPORT parameters to have an effect.

**Syntax** EXPORT *SPLIT*=(*min*, *max*)

**Parameter options** *min,max*

The minimum and maximum number of pages allowed in a segment. These values must be specified together. The numbers must be integers between 1 and 32,767. The system includes separator sheets in its count.

**Considerations** The setting for minimum and maximum pages specified in this command may be overridden by the SPLIT=NOW DJDE. Refer to the "Print control (DJDE) commands" chapter for more information on the SPLIT DJDE command.

If the system encounters the end of the segment before the *min* value specified in this command, the system generates blank

pages or multiple LAST sheets (if you have specified LAST separators) to pad the segment.

### SRECOVER

---

Specifies the mode of segment recovery. This determines how the system handles printer faults and system errors.

**Syntax**  
**Parameter options**

**EXPORT** *SRECOVER=value*

*value*

PAGE

Specifies page recovery, which is the system standard method. When the system encounters a fault, it begins reprinting with the first sheet not successfully delivered.

SEGMENT

Specifies segment recovery. When the system encounters a fault, it reprints the entire segment currently printing. Because it is possible for the printer to process more than one segment concurrently in the paper path, the system may need to reprint more than one segment to effect full recovery of the incomplete segment.

ASK

Specifies that the system ask the operator to select page or segment recovery through a message on the user interface.

DEVICE

Specifies that the system should use the recovery method defined in the DFA specification. Used only when the Bypass Transport is the output destination, otherwise, the default page recovery is used.

**Default**

The default is PAGE.

### STIMING

---

Provided for possible compatibility with other Xerox software supporting third party finishing devices and is ignored by the Xerox 4050/4090/4450/4650 LPS. Device specific timing parameters are actually set by using the Finishing Configuration (FCG) Utility.

The STIMING parameter, although ignored by the Xerox 4050/4090/4450/4650 LPS, specifies that the printer accommodate the minimum timing requirements needed by finishing equipment interfaced with the bypass transport. When needed, the printer meets the minimum time requirements by inserting holes (skipped pitches) in the paper path.

STIMING is invoked only if DESTINATION=EXPORT is in effect.

**Syntax**  
**Parameter options**

**EXPORT** *STIMING=(INTERVAL,sec,DELAY,sec)*

INTERVAL

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**

Specifies the minimum allowable time between the end-of-segment signal for one segment and delivery of the initial sheet of the next segment through the bypass transport.

*sec*

Specifies the time interval as a decimal number to a tenth of a second.

**LINE**

Allows you to control the characteristics of the user portion of the input data record and how it is to be printed, such as margins, overprinting, and carriage control.

**DATA**

Specifies the location and length of the print line data within an input data record.

**Syntax**

**LINE DATA**=(*pdo*,*length*)

**Parameter options**

*pdo*

Print data offset. This 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.

*length*

Specifies the maximum length of printable data within each logical record. The maximum allowable value is 1,000.

**Default**

For offline systems, the default is (1,132). For online systems, the default is (0,150).

**Considerations**

For HOST=RSX11(FLX), the DATA parameter must be included in the JSL, since the default *pdo* value causes the first data byte to be skipped.

If you change the print line data to a value greater than the default value (150), change the LENGTH parameter of the RECORD command accordingly.

**FCB**

---

Suppresses or accepts Forms Control Buffer (FCB) processing.

**Syntax**

**LINE** *FCB=action*

**Parameter options**

*action*

PROCESS

Accepts FCB processing. The VFU parameter defines the vertical upspacing if the FCB is ignored. For online systems only.

IGNORE

Suppresses the processing of the host-transmitted FCB.

**Default**

The default is PROCESS.

**FONTINDEX**

---

Specifies that a field within a data record defines the index to a specific font to be used for that line.

**Syntax**

**LINE** *FONTINDEX=offset*

or

**LINE** *FONTINDEX=(offset,init-val,bit-opt)*

or

**LINE** *FONTINDEX=NONE*

**Parameter options**

*offset*

Indicates the byte offset in the data record where the font index number is to be found.

*init-val*

Can be 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.

*bit-opt*

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.

NONE

Specifies that there is no font index.

**Default**

The default is NONE.

**Considerations**

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 FONTS specification. Also, internal table structures limit the number of fonts that can be invoked on a single page to 94 fonts.

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 blank characters in the previous record. Character spacing values are adjusted; thus proportionally spaced or different size fonts may be used, and the LPS performs the character placement.

Blank characters in the previous record are replaced only when they are represented by an EBCDIC X'40' or an ASCII X'20'. A binary character representation other than an EBCDIC X'40' or an ASCII X'20', which produces 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 X'40' or ASCII X'20') 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.

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 used 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 in 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 must be present in all records since it controls line spacing and overrides line spacing at page transitions.

It is advisable to put the FONTINDEX byte in the beginning of a variable record to conserve tape space.

**MARGIN**

---

Specifies the left margin on a physical page.

**Syntax** **LINE MARGIN**=*value*  
or  
**LINE MARGIN**=(*value,units*)

**Parameter options**

*value*

The form *nnn.mm* (a positive decimal number with up to two digits to the right of the decimal point), which is the distance from the left margin.

*units*

Indicates whether a *value* is specified in inches (IN), centimeters (CM), or character positions (POS). A *value* must be specified as an integer (*nnn*), if *units* is specified in character positions.

**Default** The default for *value* is 1. The default for *units* is POS.

**OVERPRINT**

---

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**=(*over-opt,disp*)

**Parameter options**

*over-opt*

PRINT

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

Ignores all overprint lines.

MERGE

Specifies the same instructions as the PRINT option, except when used with FONTINDEX or CME processing.

PRINT2

Up to two consecutive lines are printed per line: one line and one overprint. Other overprints for the line are ignored.

*disp*

Options are DISP and NODISP. These are nonfunctional parameter options that are preserved for compatibility with other Xerox laser printing systems. The number of overprint lines is always printed on the accounting page.

**Default** The default is (PRINT,NODISP).

---

**PCC**

---

Specifies the position (and possible translation) of the Printer Carriage Control (PCC) field.

**Syntax** **LINE** *PCC=(offset,trans-type)*

**Parameter options**

*offset*

Specifies the byte offset of the user portion of the record within each logical record to the PCC field.

*trans-type*

Specifies whether the PCC 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

Prohibits translation.

**Default** The default is (0,NOTRAN).

---

**PCCTYPE**

---

Specifies a carriage control set used in printing a job.

**Syntax** **LINE** *PCCTYPE=pcc-name*

**Parameter options**

*pcc-name*

ANSI  
B2500  
B2700  
B3500  
B3700  
B4700  
B6700  
H6000  
H2000  
IBM1401  
IBM1403  
IBM3211  
NCR  
UNIVAC  
US70  
XEROX  
*id*  
USER  
NONE

Creation of a user-defined PCC table referenced by either an identifier *id* or the USER parameter option is defined using the PCC parameter keyword.

**Default** For online systems, the default (and the only possible parameter) is IBM3211. For offline systems, the default is ANSI. For HOST=ICL2900, the PCCTYPE is NONE.

**Considerations** The INITIAL parameter for any selected PCCTYPE (except ANSI or HOST=RSX11, USER, and *id*) is TOF. For ANSI or HOST=RSX11, the INITIAL parameter is BOF. For USER or *id* the INITIAL parameter is set by the user in the PCC command. The ADVTAPE parameter (refer to the PCC parameter earlier in this

chapter) for any selected PCCTYPE (except IBM1403, IBM3211, USER, and *id*) is YES. For IBM1403 and IBM3211, ADVTAPE is set to NO. For USER or *id*, the ADVTAPE is set by the user in the PCC parameter.

The PCCTYPE=IBM3211 is not intended to be used when processing offline jobs. The LPS online interface performs checks for bad or NO-OP parameters; however, these checks are not done in the offline mode. The NO-OP or bad parameters are treated as undefined carriage control parameters and default to the carriage control parameter of print-and-space-1-line.

---

### UCSB

Accepts or suppresses the host-transmitted Universal Character Set Buffer (UCSB).

**Syntax**  
**Parameter options**

**LINE UCSB=action**

*action*

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.

IGNORE

Suppresses the host transmitted UCSB. If IGNORE is coded, the CODE parameter of the VOLUME command defines the character translation. For online systems only.

**Default** The default is PROCESS.

---

### VFU

Specifies the vertical format table you are using.

**Syntax**  
**Parameter options**

**LINE VFU=options**

*options*

NONE

Indicates that any skip-to-channel parameter is to be replaced by a carriage control of print and space 1.

*vfu-id*

Refers to the command identifier of the VFU table, which must precede this reference to it. The VFU table defines print line positions corresponding to skip-to-channel parameters for the job to be processed.

**Default** The default is NONE.



---

**LINE example**

**Example 1**     LINE     DATA=(1, 132), PCC=(0, NOTRAN),  
   PCCTYPE=IBM1403, FONTINDEX=133;

Defines the characteristics of the following record structure.

Figure 3-2.     **Sample record structure**



The print-data-offset option 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 one byte. The print-length parameter (the number of characters in the longest print line in the record) is defined as 132 bytes. 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 3-3.     **Sample print-data-offset option of the DATA parameter**

---

**MESSAGE**


---

Allows you to send messages to the operator during input or output processing.

---

**ITEXT**


---

Allows you to send a text message to the operator during input processing.

**Syntax** **MESSAGE ITEXT=***sc*  
or  
**MESSAGE ITEXT=(***sc,passnum***)**  
or  
**MESSAGE ITEXT=NONE**

**Parameter options***sc*

Specifies a text message of up to 80 characters.

*passnum*

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.

NONE

Specifies that no text message is output.

**Default**

The default is NONE.

**Considerations**

Hexadecimal strings used in a MESSAGE command are interpreted as standard EBCDIC and are translated to extended ASCII.

---

**OTEXT**


---

Sends a text message to the operator during job printing.

**Syntax** **MESSAGE OTEXT=***sc*  
or  
**MESSAGE OTEXT=(***sc,passnum,WAIT***)**  
or  
**MESSAGE OTEXT=(***sc,END,WAIT***)**  
or  
**MESSAGE OTEXT=NONE**

**Parameter options***sc*

Specifies a text message of up to 80 characters (maximum of 400 characters per report).

*passnum*

Specifies the copy to which the text applies. Multiple *sc* messages, one per *passnum*, may be specified in a JSL. The message is output to the operator prior to the beginning of printing the specified report ply (pass). 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. The maximum number of OTEXT characters that can be used in a report with GHO processing specified is 320 characters.

NONE

Specifies that no text message is to be output to the operator during job printing.

**Default**

The default is NONE.

**Considerations**

Hexadecimal strings used in a MESSAGE command are interpreted as standard EBCDIC and are translated to extended ASCII.

**OUTPUT**

Allows you to control the organization and format of a report.

The following parameters require specific action or system response:

- COLLATE
- DUPLEX
- FACEUP
- NTO1
- STAPLE.

These restrictions follow a specific hierarchy by which one command parameter forces changes in the other command specifications, as shown in table 3-1.

Table 3-1. **Hierarchy of OUTPUT command parameter restrictions**

If ...	Then ...			
	COLLATE =	DUPLEX =	FACEUP =	NTO1 =
FACEUP = YES	—	NO	YES	—
NTO1 = YES	YES	<i>specified option</i>	YES	YES
STAPLE = YES	YES	<i>specified option</i>	YES	YES

**NTO1, FACEUP, and accounting or Interpress page generation**

If finishing enhancement device software is applied, all Interpress error pages are redirected to the sample tray to prevent them from becoming part of the finished job output. The NTO1 and FACEUP parameters can help you determine which system generated accounting sheets or Interpress error pages correspond to each job. These relationships are shown in table 3-2.

Table 3-2. Accounting and Interpress sheet generation in relation to NTO1 and FACEUP parameters

NTO1	FACEUP	First Out Sheet	Last Out Sheet	Faceup or Facedown
NO	YES	Interpress	Accounting	Faceup
YES	NO	Interpress	Accounting	Faceup
NO	YES	Accounting	Interpress	Facedown
YES	YES	Interpress	Accounting	Faceup

**DUPLEX=YES** Any job that runs in simplex mode runs in duplex mode.

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.
- Odd-numbered page results in one additional hole in the paper path and no aborted pages to the sample tray.

**NTO1, STAPLE** If the default stock order is sysgened to be 1TON, the NTO1 and STAPLE parameters specified in a JDE or JDL cannot be overridden by DJDEs.

**DUPLEX=NO, ACCT USER=TRAY, BFORM ignored, RPAGE SIDE=NUBACK, forced to NUFRONT**

Upon the start of a report, the input task checks if transparency is specified by an operator command or in a JSL. Any operator input overrides the JSL specifications. If transparency is specified, the input task forces these parameters.

**CYCLEFORMS, FORMS** Both may be used within a report; however, only the last parameter specified is in effect at any given time. If a multiple copy report specifies CYCLEFORMS, a FORMS DJDE encountered in the middle of a report overrides the CYCLEFORMS parameter from that point in every copy.

**RAUX, FEED** If they specify two different cluster names, RAUX has higher priority.

**SYSPAGE parameter of STOCKSET command**

- System pages are checked for ordered stocks through this 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 specified cluster-name.

- All applications that are printed on tab stocks 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 31 (decimal). It also ensures the value of POS is greater than or equal to 1 and less than or equal to MOD.

**Overprint ratio**

An overprint ratio is the maximum number of variable data and forms characters that may be intersected by a single scan line. If this ratio is exceeded, the page does not print and the LPS indicates a local density error. Each type of laser printing system has a specific overprint ratio which is determined by a number of factors, such as engine speed and character resolution. Thus, a job containing high character content, such as form characters, rules, and shading, for example, may print successfully on one LPS but fail to print on another, or may print at one resolution (300 or 600) but fail to print at the other.

If a local density problem occurs, it can be corrected in one of the following ways:

- Redesign the job to reduce the concentration of characters, thus avoiding the local density problem.
- Invoke the OUTPUT command with the DENSITY=FIX parameter in the JDL to direct the system to recover from the imaging error.

**BFORM**

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 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=form-id*  
or  
**OUTPUT** *BFORM=(form-id,init,copies)*  
or  
**OUTPUT** *BFORM=NONE*

**Parameter options**

*form-id*

Specifies a 1 to 6 character filename (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language (FSL) source file.

*init*

If the *copies* parameter is not defined, the last (or only) form specified will apply to all copies beginning with copy number *init*. If the form is not the last one specified, copies defaults to 1. If neither *init* nor *copies* is specified, the form applies to all copies of the report. Note that *init* copies is only valid offline.

*copies*

Specifies the number of plies (passes) to which a specified form applies.

NONE

No form is added to the associated report page of variable data.

**Default** The default is NONE.

---

**COLLATE**

Specifies whether pages are collated or uncollated.

**Syntax** **OUTPUT COLLATE=***value*

**Parameter options**

*value*

YES

Collates output pages.

NO

Does not collate output pages.

**Default** The default is YES.

**Considerations**

If DUPLEX=YES, PDL ignores COLLATE=NO. If NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES and includes a warning in the JSL. Once you have specified multiple copies, the COLLATE parameter cannot be changed again within the report.

---

**COPIES**

Specifies the number of report copies you want printed.

**Syntax** **OUTPUT COPIES=***value*

**Parameter options**

*value*

An integer value with range 1 to 32767.

**Default** The default is 1.

**Considerations**

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.

---

**COVER**

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=***value*

or

**OUTPUT COVER=(***value,SEP***)**

<b>Parameter options</b>	<p><i>value</i></p> <p><b>FRONT</b> Specifies that a cover page is to be picked (from the auxilliary tray) at the front of each copy and is the first page of the copy.</p> <p><b>BACK</b> Specifies that a cover page is to be picked at the end of each copy. No report data is printed on back covers.</p> <p><b>BOTH</b> 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.</p> <p><b>NONE</b> Specifies that no cover pages are to be picked.</p> <p><b>SEP</b> 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.</p>
<b>Default</b>	The default is NONE.
<b>Considerations</b>	<p>The FEED MAIN operator command 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.</p> <p>If RTEXT is specified along 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 the MESSAGE and ROUTE commands for further details on RTEXT.)</p>
<hr/> <b>CYCLEFORMS</b> <hr/>	
<b>Syntax</b>	<p>Associates a set of forms with report pages in a cyclical fashion.</p> <p><b>OUTPUT CYCLEFORMS=(<i>form-id</i>,...)</b> or <b>OUTPUT CYCLEFORMS=NONE</b></p>
<b>Parameter options</b>	<p><i>form-id</i> Specifies a 1 to 6 character filename (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.</p> <p><b>NONE</b> No form is added to the associated report page of variable data.</p>
<b>Default</b>	The default is NONE.
<b>Considerations</b>	<p>If multiple CYCLEFORMS left parts in the same OUTPUT command are specified, a single list consisting of all forms specified in the OUTPUT command is created.</p> <p>The order of the list is the order in which the names are encountered. The number of <i>form-ids</i> is limited only by the number of forms retained on the disk.</p> <p>Specifying CYCLEFORMS (or multiple forms in the FORMS parameter) in a JSL causes the value specified by the FORMS</p>

operator command to be overridden, if necessary. This occurs when the number of forms specified in the CYCLEFORMS or FORMS parameters is larger than the FORMS operator command value.

**Example** OUTPUT CYCLEFORMS=(*form<sub>1</sub>*,*form<sub>2</sub>*,*form<sub>3</sub>*)

---

**DENSITY**

---

Disables IG local density functionality of continuous printing for specific jobs.

**Syntax** OUTPUT DENSITY=*value*

**Parameter options** *value*

DEFAULT

Directs output to respond to local density events according to the default for the printer type on which output is running.

FIX

Directs output processing to intervene in the event of IG local density and to perform continuous printing recovery.

NOFIX

Directs output processing to handle local density events in the customary fashion, such as by aborting the page.

**Default** The default is DEFAULT.

---

**DESTINATION**

---

Specifies the destination of the printed output. only.

**Syntax** OUTPUT DESTINATION=*value*

**Parameter options** *value*

TRAY

Sends output to the sample tray. Use this option for transparencies, as they can be sent only to the sample tray.

BIN

Sends output to the output bin installed on your system.

EXPORT

Sends output to the bypass transport. The EXPORT option takes effect only if the operator command options SELECT 1 or SELECT E are specified. (This option is available with XDDI only.) For more information on the SELECT command, refer to the *Xerox 4050/4090/4450/4650 LPS Command Reference* or the *Xerox 4050/4090/4450/4650 LPS Operator Guide*.

**Default** The default is output to stacker trays or bins.

---

**DUPLEX**

---

Specifies whether printing is to occur in duplex or simplex.

**Syntax** OUTPUT DUPLEX=*value*

**Parameter options** *value*

NO



Specifies simplex printing (prints on one side of a sheet of paper).

YES

Selects duplex printing (prints on both sides of a sheet).

**Default** The default is NO.

**Considerations** If you specify the PAPERSIZE=FORM and DUPLEX=YES parameter statements, the actual paper size used for printing is based on the size of the back side page. For example, if the front side references a 8.5 by 11 inch paper size and the back side references a 8.5 by 14 inch paper size, this duplex page is printed using 8.5 by 14 inch paper size.

---

## FACEUP

Specifies faceup delivery of pages.

**Syntax** **OUTPUT** *FACEUP*=*value*

**Parameter options** *value*

NO

Specifies sheets are delivered facedown to the stacker tray.

YES

Specifies that the sheets are delivered to the tray faceup. Specify FACEUP YES for labels.

**Default** The default is NO.

---

## FEED

Controls the stock on which the page is printed.

**Syntax** **OUTPUT** *FEED*=*value*  
or  
**OUTPUT** *FEED*=*stock-reference*  
or  
**OUTPUT** *FEED*=*stock-name*

**Parameter options** *value*

OPR

MAIN

AUX

Provides compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN.

*stock-reference*

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 you to change stocks associated with a job without altering the stock references in the data application.

*stock-name*

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.)

**Default** The default is OPR.

**FORMAT**

---

Specifies a Page Descriptor Entry (PDE), listed in table 3-3, to use 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=pde-id*

**Parameter options**

*pde-id*

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.

**Default**

The default is FMT1.

**Considerations**

Standard *pde-ids*, such as FMT1 and FMT2, are defined in table 3-3. These standard *pde-ids* 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 via DJDEs.

To ensure reliability, key in the entire parameter keyword **FORMAT**. Do not abbreviate.

Table 3-3. Standard LPS print formats

PDE id	Lines	Columns	LPI	CPI	Point size (approx.)	Page size and orientation *	BEGIN values	Default font id
FMT1	66	132	8.1	13.6	9	11 x 8.5	(.18,.66)	L0112B
FMT2	66	150	8.1	15	9	11 x 8.5	(.18,.50)	L0212A
FMT3	88	132	10.7	13.6	7	11 x 8.5	(.14,.66)	L0312A
FMT4	88	150	10.7	15	7	11 x 8.5	(.14,.50)	L0412A
FMT5	49	100	6	10	12	11 x 8.5	(.17,.50)	L0512A
FMT6	80	100	8.1	13.6	9	8.5 x 11	(.57,.58)	P0612A
FMT7	60	90	6	12	12	8.5 x 11	(.50,.50)	P07TYA
FMT8	60	75	6	10	12	8.5 x 11	(.50,.50)	P0812A
FMT9	60	200	10.0	20.0	7	11 x 8.5	(.25,.25)	L0912A
FMT10	132	132	12.5	17.6	6	8.5 x 11	(.22,.51)	P1012A
FMT11	132	150	12.5	20.0	6	8.5 x 11	(.22,.50)	P1112A
FMT12	66	172	8.1	13.6	9	14 x 8.5	(.18,.66)	L0112B
FMT13	104	100	8.1	13.6	9	8.5 x 14	(.57,.58)	P0612A
FMT1A	66	132	8.3	12.5	9	11.69 x 8.27	(.18,.57)	R112BL**
FMT2A	66	150	8.3	14.3	9	11.69 x 8.27	(.18,.60)	R212BL**
FMT3A	88	132	11.1	12.5	7	11.69 x 8.27	(.18,.57)	R312BL**
FMT4A	88	150	11.1	14.3	7	11.69 x 8.27	(.18,.60)	R412BL**
FMT5A	48	100	6	10	12	11.69 x 8.27	(.22,.85)	R512BL**
FMT6A	80	100	8.1	113.6	9	8.27 x 11.69	(.91,.46)	R612BP**
FMT7A	60	90	6	12	12	8.27 x 11.69	(.85,.39)	R7TIBP**
FMT8A	60	75	6	10	12	8.27 x 11.69	(.85,.39)	R812BP**
FMT9A	80	200	10.0	20.0	7	11.69 x 8.27	(.14,.85)	R912BL**
FMT10A	132	132	12.5	17.6	6	8.27 x 11.69	(.57,.39)	RA12BP**
FMT11A	132	150	12.5	20.0	6	8.27 x 11.69	(.57,.39)	RB12BP**

\* Dimensions are in inches. Landscape orientations are 11 x 8.5, 14 x 8.5, and 11.69 x 8.27. Portrait orientations are 8.5 x 11, 8.5 x 14, and 8.27 x 11.69.

\*\* A4 European international formats.

**FORMS**

---

Associates forms with the report copies. Different forms may be associated with different copies of a report by the use of multiple FORMS left parts in the same OUTPUT command.

**Syntax**     **OUTPUT FORMS=***form-id*  
                  or  
                  **OUTPUT FORMS=(***form-id,init,copies***)**  
                  or  
                  **OUTPUT FORMS=NONE**

**Parameter options**

*form-id*

Specifies a 1 to 6 character filename (may be numeric, alpha, or alphanumeric) that exists on disk. This file is created by compiling a Forms Description Language source file (FSL file), with the FDL system task.source file (FSL file) with the FDL system task.

*init*

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 *init*. If the form is not the last one specified, *copies* defaults to 1. If neither *init* nor *copies* is specified, the form applies to all copies of the report.

*copies*

The number of copies of the report you want to print.

NONE

No form is added to the associated report page of variable data.

**Default**

The default is NONE.

**Considerations**

To ensure reliability, enter the entire parameter keyword FORMS. Do not abbreviate to FOR because the system interprets it as FORMATS.

Specifying multiple forms in the FORMS parameter (or CYCLEFORMS) in a JSL causes the value specified by the FORMS operator command to be overridden, if necessary. This occurs when the number of forms specified in the FORMS or CYCLEFORMS parameters is larger than the FORMS operator command value.

**Example**

OUTPUT FORMS=(FORM3,3,2);

**GRAPHICS**

---

Specifies how to process graphics in a job.

**Syntax**     **OUTPUT GRAPHICS=***value*  
                  or  
                  **OUTPUT GRAPHICS=(***value,NOSUB***)**

**Parameter options**

*value*

NO

Indicates there are no graphics in this job.

YES

Indicates there are graphics in this job.

**MOVE**

Copies all referenced graphic disk files into the print file.

**BATCH**

Specifies 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.

**Default**

The default is NO.

**Considerations**

If you are using the IMAGE parameter in conjunction with GRAPHICS, list the GRAPHICS parameter first in your JSL, as these two parameters are order dependent.

**Example**

```
OUTPUT GRAPHICS=BATCH,
      RESOLUTION=300;

OUTPUT GRAPHICS=MODE,
      RESOLUTION=300,PURGE=NO;
```

**IMAGE**

Specifies the initial graphic imaging parameters to be used for batch mode processing only.

**Syntax**

**OUTPUT** *IMAGE*=(*vpos units,hpos units,n/d*)

**Parameter options***vpos*

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.

*units*

CM (centimeters)  
IN (inches)  
UN (user-defined unit)  
DOTS  
XDOTS

If the UN parameter is used to define positioning, the UNITS parameter must precede this command in the parameter line.

*hpos*

Specifies the horizontal position of the left edge of the graphic, as an offset relative to 0,0 on the current physical page. The *units* of the specification is the same as for *vpos*.

*n/d*

Specifies the reference scale factor.

**Default**

The default is top left corner, scaled at 1/1; measurement is in inches.

**Considerations**

You create, edit, and compile an image specifying XDOTS on an LPS with version 3 software.

**MODIFY**

---

Specifies the CMEs to be associated with report copies.

**Syntax**     **OUTPUT MODIFY=cme-id**  
 or  
**OUTPUT MODIFY=(cme-id,init,copies)**  
 or  
**OUTPUT MODIFY=NONE**

**Parameter options**

*cme-id*  
 The command identifier of a CME command. 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. If the CME command is within the JSL, it must precede a reference to its identifier by the MODIFY parameter.

*init*  
 Specifies the initial ply (pass) to which the associated CME is to be applied.

*copies*  
 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 *init*.

NONE  
 Processes data without modification.

**Default**     The default is NONE.

**Considerations**     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, 871-CM, or online processing is specified.

**NTO1**

---

Specifies, on a report basis, whether all copies of a particular report are printed last page to first (N-1), rather than the typical method (and the default) of printing a report in first-to-last sequence.

**Syntax**     **OUTPUT NTO1=value**  
 or  
**OUTPUT NTO1=n**

**Parameter options**

*value*  
 NO  
 Prints first page to last (1TON). NTO1=n allows you to specify the threshold beyond which input overrides the desire to print a report in N-to-1 mode, and print 1-N instead. (When NTO1=n is specified, NTO1=YES is assumed.)

YES

Specifies last page to first. To print N-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.

*n*

Designates the physical page count within a report that, once encountered by input and before encountering the end of the report, results in LPS override to 1-to-N and immediate cycle-up of the engine. The default value of *n* is 50 physical sheets, which is the nominal capacity of the stitcher for a single stitch.

**Default**

The default is NO.

**Considerations**

NT01=YES, PDL overrides COLLATE=NO with COLLATE=YES, and notes this action with a warning in the JSL listing.

The entire report must be stored in the print file and the system must know which is the last page before it can begin printing. Therefore, a limit, referred to as the N-to-1 threshold, is imposed on the size of the report. This limit can be the default value of 50 or the *n* value you specify. If the report exceeds this limit, the entire report prints in 1-to-N order.

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 forces FACEUP=YES to be invoked.

**NUMBER**

Specifies page numbering on the output pages of a report.

**Syntax**

**OUTPUT NUMBER=(*pnum*,*lnum*,*cnum*,*findex*)**

or

**OUTPUT NUMBER=NO**

**Parameter options***pnum*

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 becomes positive. The maximum page number that will be printed is 2,147,483,647.

*lnum*

Specifies an integer line number on which the page number is to be placed.

*cnum*

Specifies an integer ending column number for the page number sequence.

*findex*

A font index in 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.

<b>Default</b>	The default is NO.
<b>Considerations</b>	<p>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.</p> <p>If both BFORM and NUMBER are specified, BFORM pages are numbered.</p>
<b>Example</b>	OUTPUT NUMBER=(1,1,40,1);
<hr/> <b>OFFSET</b> <hr/>	
Specifies offset control on a report basis.	
<b>Syntax</b>	<b>OUTPUT</b> <i>OFFSET=value</i>
<b>Parameter options</b>	<p><i>value</i></p> <p>ALL Results in an offset of each copy of each report.</p> <p>FIRST Specifies that an offset is to occur only on the first copy of a report.</p> <p>NONE Specifies that there is to be no offset at any time.</p>
<b>Default</b>	The default is ALL.
<b>Considerations</b>	The OFFSET control of FIRST, ALL, or NONE may be modified by the ROFFSET command. 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.
<hr/> <b>PAPERSIZE</b> <hr/>	
Identifies the paper size to be used for printing the job.	
<b>Syntax</b>	<b>OUTPUT</b> <i>PAPERSIZE=value</i> or <b>OUTPUT</b> <i>PAPERSIZE=(x,y)</i>
<b>Parameter options</b>	<p><i>value</i></p> <p>A4 8.27 by 11.69 inches/210 mm by 296 mm</p> <p>USLEGAL 8.5 by 14 inches/216 mm by 356 mm</p> <p>USLETTER 8.5 by 11 inches/216 mm by 279 mm</p> <p>(<i>x,y</i>) A variable paper size selected at system generation. The <i>x</i> is the short edge of the sheet and the <i>y</i> is the long edge. The <i>x</i> and <i>y</i> variables may be specified in either inches (IN) or centimeters (CM).</p>
<b>Default</b>	The default is the paper size selected at system generation.
<b>Considerations</b>	When centimeters (CM) are used to specify paper size, the maximum allowable value is 32.767 cm.



You cannot change paper size directly through DJDE commands on a page basis. Paper sizes can be changed on a report basis using a DJDE JDE or JDL that calls out the desired paper size.

With the FORM option of the PAPERSIZE parameter forms of different sizes for the same print job sent using DJDE, the system matches the paper size of each page to the paper size specified for the forms. The system allows forms of different sizes through DJDEs, and matches the physical paper size to the paper size specified on the form.

Whether you use the *value* parameter option or the (x,y) method for specifying paper size in your JSL, you must use the same method in your FSL.

When DUPLEX=YES and two FORMS DJDEs with two different size forms are specified for the same physical page (one for each side of the page), the larger paper size is used.

Note that the system identifies A4 paper as smaller than 8.5 by 11 inch paper because it is narrower.

---

## PURGE

Specifies whether the system should delete graphic disk files at the end of a report.

**Syntax**  
**Parameter options**

**OUTPUT** *PURGE=value*

*value*

YES

Deletes all IMG graphics files created or replaced by document interleaved graphic processing for this report at the end of the report printing.

NO

Does not delete graphic disk files.

**Default**  
**Considerations**

The default is YES.

A DJDE SAVE specification may be used to override the purging of specific files.

---

## RESOLUTION

Defines the output coordinate system (300 or 600 spi).

**Syntax**  
**Parameter options**

**OUTPUT** *RESOLUTION=spi*

*spi*

300

Prints output, including fonts, forms, images, and logos, at 300 spi.

600

Refers to all fonts, forms, images, and logos in FN6, FR6, LG6, and IM6 directories. On systems printing at 600 spi, references in the JSL to 1 IN, 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.

**Default**

The default is the parameter specified at system generation.

**SHIFT**

---

Specifies that the image of the form and data on a page is to be shifted. This parameter can be used to shift the image of the data for purposes of three-hole drilling or binding.

**Syntax** **OUTPUT SHIFT=value**

or  
**OUTPUT SHIFT=(v1,v2)**

**Parameter options**

*value*

NO

Specifies that no shift occurs.

YES

Specifies that a shift occurs. A standard size shift occurs if YES is specified ( $v_1 = 75$  and  $v_2 = -75$ ).

$v_1$

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.

**Default**

The default is NO.

**Considerations**

When duplex shift is used, the left margin must be at least 0.25 inches (75 dots) wide. Care should be taken when using SHIFT to ensure that the data is not shifted past the margin and off the page.

**SIZING**

---

Specifies how the system matches the paper sizes in trays when the LPS is unattended.

**Syntax** **OUTPUT SIZING=value**

**Parameter options**

*value*

SEMIAUTO

The standard mode of operation.

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. Called nonstop mode because the LPS does not cycle down. The system prints reports only if the exact paper size is available. If the exact paper size is not available, the system aborts the report.

**Default**

The default is SEMIAUTO.

**Considerations**

A paper size is equal in size to another paper size if neither dimension differs by more than 17/1600 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.

---

## STAPLE

Selects stitching of copy sets with an LPS configured with the stitcher/stacker.

**Syntax**  
**Parameter options**

**OUTPUT** *STAPLE=value*

*value*

NO

Specifies no stitching.

YES

Specifies stitching and implies NTO1=YES, FACEUP=YES, and COLLATE=YES, but not vice versa. Individual sheets cannot be offset when coded in the JSL. To maintain consistent faceup orientation as stitched and unstitched reports are delivered to the stacker, unstitched sets may be printed NTO1, provided that the defined override conditions are not encountered. To stitch copy sets properly, sheets must be delivered to the finisher staging tray faceup.

**Default**  
**Considerations**

The default is NO.

If NTO1=NO and STAPLE=YES, the system overrides NTO1=NO with NTO1=YES and notes this action with a warning in the JSL listing. When an entire set is delivered to the stitcher compiler tray, output issues a parameter to stitch the set.

For portrait sheets, the staple is driven into the upper left corner, which is normal. However, the same corner that is the upper left corner for the portrait orientation is the upper right for landscape orientation. For landscape reports, the staple may appear to be in the wrong corner.

---

## STOCKS

Defines the stockset and its associated stock.

**Syntax**  
**Parameter options**

**OUTPUT** *STOCKS=stockset-name*

*stockset-name*

Defines the stockset and its associated stock to use 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.

**Default**  
**Considerations**

There is no default.

Whenever a new stockset is chosen at the 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.

**UNITS**

Defines the dot size of a unit used for specifying the position of a graphic when referenced by the IMAGE parameter keyword on the OUTPUT command or a DJDE IMAGE, ALTER, or GRAPHIC parameter.

<b>Syntax</b>	<b>OUTPUT UNITS=</b> <i>size</i>
<b>Parameter options</b>	<i>size</i> 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.
<b>Default</b>	There is no default.

**OUTPUT examples**

<b>Example 1</b>	OUTPUT	PAPERSIZE=USLETTER, DUPLEX=NO, NUMBER=(1,1,80), FORMAT=SMPPOE;
<b>Example 2</b>	OUTPUT	GRAPHICS=MOVE, UNITS=150, CYCLEFORMS=NONE, FORMAT=PDE1, RESOLUTION=300;
<b>Example 3</b>	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 GBAR form appears on each data page of the report.

**ac:PDE**

Specifies a Page Descriptor Entry (PDE) that defines customized 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 use.

You use the PDE command when you want to specify a format that is different than any of the standard LPS formats listed in table 3-3. You specify the standard LPS formats with the FORMAT parameter of the OUTPUT command.

You must specify an identifier of the type *ac* when defining the PDE, and reference it with the FORMAT parameter of the OUTPUT command. The command identifier *ac* may consist of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter. 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 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.

### Multiple logical pages on 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 move to the next logical page. Note the following:

- Each online banner page is positioned as the first logical page of a new physical sheet (if BANNER 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 the OUTPUT command NUMBER parameter, 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.
- Overlapping BEGINS may cause line density exceeded failures.
- 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.
- When applied, the following DJDEs move the current page position to the first logical page of the front side of a new physical sheet:
  - BFORM
  - COPIES
  - DUPLEX
  - JDL
  - JDE
  - OTEXT
  - RTEXT.

- When applied, the following DJDEs move the current page position to the first logical page of the next physical side:
  - FORMAT
  - FORMS.

---

## BEGIN

---

Specifies the location of the starting print line of a logical page for graphics.

**Syntax**  
**Parameter options**

*ac:PDE BEGIN=(vpos,hpos)*

*vpos*

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 IN. You can specify a decimal number with up to three digits to the right of the decimal point, such as 0.563 IN and 2.356 CM.

*hpos*

Specifies the horizontal position of the first character of the first print line on the logical page (in IN or CM as for *vpos*). All specifications are rounded to the nearest dot (1/300 of an inch) for positioning of the logical page.

**Default**

The default is (.18 IN,.66 IN).

**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

---

Specifies the fonts to use in printing variable input data and CME data.

**Syntax**  
**Parameter options**

*ac:PDE FONTS=(f<sub>1</sub>,f<sub>2</sub>,...)*

or

*ac:PDE FONTS=((f<sub>1</sub>,s<sub>1</sub>),(f<sub>2</sub>,s<sub>2</sub>),(...))*

or

*ac:PDE FONTS=((f<sub>1</sub>,s<sub>1</sub> units),(f<sub>2</sub>,s<sub>2</sub> units),(...))*

*f*

Specifies the name of the font. It is a 1 to 6 alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on system disk. Up to 128 fonts can be used with the font indexing capability.

*s*

Specifies an optional override line-spacing value. Maximum value for *s* is 30 lpi; minimum value is 10 dots.

*units*

LPI (lines per inch)  
 DOTS (dots per print line) to be associated with the font  
 XDOTS (1/600 unit of measurement).

**Default**

If *units* are not specified, LPI is the default.

**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.

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 that 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 cache where information about the fonts is stored.

If the patch to enable the use of Merger 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.

## PMODE

Specifies the printing mode for each physical sheet.

<b>Syntax</b>	<i>ac:PDE PMODE=orientation</i>
<b>Parameter options</b>	<i>orientation</i> 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.
<b>Default</b>	The default is LANDSCAPE.

## ac:PDE examples

```

Example 1  PDE1:      PDE   PMODE=LANDSCAPE,
                BEGIN=(.861, .7)
                FONTS=(L0112B, L01BOB);
                PDE2:      PDE   PMODE=PORTRAIT,
                BEGIN=(1.3, .37),
                FONTS=((P08TYA, 6.8), (P08OAA, 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 two 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;  
 This is an example of two BEGINS. Two logical pages are defined on one physical page.

**Example 3** PDE999: PDE PMODE=PORTRAIT, BEGIN=(.10, 0),  
 FONTS=(P071TA, P08OAA);  
 END;  
 In this example, a PDE may be compiled separately from a JDL. After compilation, a PDE object file is created in the PDE directory that may then be referenced by 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).

**ac:ROUTE**

---

Prints 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.

**RFORM**

---

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** *ac:ROUTE RFORM=NONE*  
*or*  
*ac:ROUTE RFORM=form-id*

**Parameter options** NONE  
 Specifies that no form is to be printed.  
*form-id*  
 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. The RFORM parameter is not allowed in a ROUTE command that is being defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file.

**Default** The default is NONE.

**RTEXT**

---

Specifies text to be printed on a separate page preceding a report (or copy ply).

**Syntax** *ac:ROUTE RTEXT=sc*  
*or*  
*ac:ROUTE RTEXT=rtext-id*  
*or*  
*ac:ROUTE RTEXT=(sc,value,line,col,fontindex)*  
*or*  
*ac:ROUTE RTEXT=NONE*



<b>Parameter options</b>	<p><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 <i>fontindex</i>).</p> <p><i>rtext-id</i> References a cataloged file of RTEXT parameters previously compiled by PDL. 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.</p> <p><i>value</i> An integer number or the <i>value</i> ALL. An integer number specifies the pass (copy ply) to which the text applies.</p> <p><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 <i>passnum</i>.</p> <p><i>col</i> Specifies the column number at which the first character of a block of RTEXT messages is to be printed (default is column 1).</p> <p><i>fontindex</i> Identifies the index (starting with 1) of the PDE font with which the text is printed.</p> <p>NONE Specifies that no text is to be printed on a separate page.</p>
<b>Default</b>	The default is NONE.
<b>Considerations</b>	<p>The font index is associated only with a particular string. To print an entire multiple-line 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).</p> <p>The font index is associated only with a particular string. To print an entire multiple-line 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).</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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</p>

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);
```

## **ac:STOCKSET**

---

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.

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. The command identifier *ac* may consist of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter.

### **ASSIGN**

---

Lists stock names and associates stock references with stock names.

**Syntax** *ac:STOCKSET ASSIGN=stock-descriptor*  
or

*ac:STOCKSET ASSIGN=(stock-descriptor<sub>1</sub>,stock-descriptor<sub>2</sub>,...)*

**Parameter options**

*stock-descriptor*

Associates stocks with the identified stockset, where the *stock-descriptor=stock-name* only or *stock-name* and *stock-reference*.

**Default**

There is no default.

### **INIFEED**

---

Specifies the stock to use in the absence of any OUTPUT command FEED parameter specifications. If not specified, the INIFEED parameter defaults to the first stock name specified in the ASSIGN parameter.

**Syntax** *ac:STOCKSET INIFEED=stock-name*  
or

*ac:STOCKSET INIFEED=stock-reference*

**Parameter options**

*stock-name*

Bypasses the reference feature, but still requires that the *stock-name* is specified as present in the current STOCKSET. OUTPUT FEED=*stock-reference* should always be used.

*stock-reference*

Specifies the stock assigned to a *stock-name* by the STOCKSET command in effect at the time the page is printed. INIFEED=*stock-reference* allows you to change paper stocks associated with a job, without altering the stock references in the data application itself.

**Default** The default is the first *stock-name*.

**SYSPAGE**

Specifies the stock to use for system-generated pages, such as the DJDE OPRINO, PLABEL, exception, and accounting pages.

**Syntax**

*ac*:STOCKSET SYSPAGE=*value*

or

*ac*:STOCKSET SYSPAGE=*stock-name*

or

*ac*:STOCKSET SYSPAGE=*stock-reference*

**Parameter options**

*value*

MAIN or AUX

Provides compatibility to existing applications. OUTPUT FEED=OPR is equivalent to FEED=MAIN.

*stock-name*

Bypasses the reference feature, but still requires that the *stock-name* is specified as present in the current STOCKSET. OUTPUT FEED=*stock-reference* should always be used.

*stock-reference*

Specifies the stock assigned to a *stock-name* by the STOCKSET command in effect at the time the page is printed. OUTPUT FEED=*stock-reference* allows you to change paper stocks associated with a job, without altering the stock references in the data application itself.

**Default** There is no default.

**Considerations**

System-generated pages (such as PLABEL or OPRINFO) are formatted using 8.5 by 11 inch paper, unless the system has been sysgened for A4 paper; then system-generated pages are formatted using A4 paper. System-generated pages are printed using the SYSPAGE parameter that is in effect. If no STOCKSET is in effect, cluster MAIN is used unless overridden by an operator command.

An attempt is made 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.

## ac:STOCKSET example

---

The STOCKSET command is similar to the PDE, CME, and IDR commands in that it can be compiled separately. For each STOCKSET the system encounters before an initial JDL or SYSTEM command, you create files of the type .STK. The .STK file can then be referenced in multiple JSL source files.

```
BILLS:  STOCKSET;
        ASSIGN=('F1302', COVER),
        ASSIGN=('F1415', (BODY, SUMMARY)),
        ASSIGN=((('F6204', LATE), 9'F9999', DISCON)),
        INIFEED=COVER, /*INIFEED='F1302' WORKS AS
        WELL*/
        SYSPAGE=COVER;
```

In this example, 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 option of the FEED parameter at the start of a 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'. The use of the stock reference is recommended over the use of stock names.

## ac:VFU

---

Assigns 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 (TOF) and the bottom-of-form (BOF). Top-of-form indicates the number of lines from the top (as defined by the PDE command BEGIN values) of an output page to the first print line on the page.

TOF defines the *first* line on which printing appears. BOF 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 pre-job page alignment and for page bottom-of-form overflow processing. For all PCCTYPES except ANSI and user-defined PCCs, the pre-job 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.

If you attempt to print a line lower than the current line number or exceed the BOF, the print line skips to the next page and line spacing is continued from the top-of-form line number of the next page. You can print on the page even if you have exceeded the BOF value if you have Honeywell 2000 (H2000) carriage control or Xerox carriage control processing.

You specify an identifier of the type *ac* when defining the VFU table and reference it in the VFU command of the LINE command. The VFU command must precede the LINE command. The command identifier *ac* may consist of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter.

---

## ASSIGN

---

Specifies the output line-to-channel assignments.

### Syntax

*ac*:VFU ASSIGN=(*channo*,*lineno*)

or

*ac*:VFU ASSIGN=(*channo*,(*lineno*<sub>1</sub>,*lineno*<sub>2</sub>,...))

### Parameter options

*channo*

Identifies the number of the channel being assigned. It is an integer in the range 0 to 15. You can 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.

*lineno*

Identifies 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.

### Default

There is no default.

### Considerations

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 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.

**BOF**

---

Assigns the bottom-of-form line number.

**Syntax**  
**Parameter options**

*ac:VFU BOF=value*  
*value*

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.

**Default**

The default is 66.

**Considerations**

If you specify a line number that is greater than the current line number or exceeds BOF (i.e., exceeds the maximum value of 255), the print line skips to the next page and line spacing is continued from the top-of-form line number of the next page. BOF is ignored if you have Honeywell 2000 (H2000) or Xerox carriage control processing.

If carriage control action to perform at BOF is specified as OVR (the default), the system spaces from TOF a number of lines equal to the difference between its location and the new BOF.

If the TOF parameter is specified as a value greater than or equal to the BOF value, the TOF value is set to the value of 1.

**TOF**

---

Specifies the number of lines from the top of the output page to the first print line on the page (top-of-form).

**Syntax**  
**Parameter options**

*ac:VFU TOF=value*  
*value*

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.

**Default**

The default is 1.

**Considerations**

If the TOF parameter is specified as a value greater than or equal to the BOF value, the TOF value is set to the value of 1.

**VFU example**

---

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 LPS 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 page 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.





---

## 4. Logical processing commands

The logical processing commands allow you to specify special functions performed on either a record, a set of records, or on a block basis.

---

### BANNER

Allows you to define the banner page detection test online.

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 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 they are separated from each other logically but not physically (with tape marks, 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 that must satisfy the test criteria before end-of-report occurs.

---

### HCOUNT

Specifies the total number of consecutive banner pages.

**Syntax**  
**Parameter options**

**BANNER** *HCOUNT=value*

*value*

Specifies the total number of consecutive header banner pages which must be detected to satisfy the test expression.

**Default**  
**Considerations**

The default is 0.

Refer to "Considerations" in the TCOUNT parameter section of this chapter for information about how the HCOUNT and TCOUNT parameters work under certain conditions.

**HJOBNO**

---

Specifies that for each report, a subfield of the first record satisfying the banner selection criteria is displayed as "Customer ID" in response to the operator JOBS command. The Customer ID represents the job number or job ID.

**Syntax** **BANNER HJOBNO=(*offset,length*)**  
or  
**BANNER HJOBNO=NONE**

**Parameter options** *offset*  
Specifies the offset in bytes (relative to zero) from the start of the user portion of the record to the subfield within the record.

*length*  
Specifies the number of bytes in the subfield (0 to 6).

NONE  
Specifies that no HOST job number is to be selected.

**Default** The default is NONE.

**HRPTNA**

---

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. For jobs that specify trailers only with HCOUNT=0, the report name is associated with the report that precedes the trailer banner page.

**Syntax** **BANNER HRPTNA=(*offset,length*)**  
or  
**BANNER HRPTNA=NONE**

**Parameter options** *offset*  
Specifies the offset in bytes (relative to zero) from the start of the user portion of the record to the subfield within the record.

*length*  
Specifies the number of bytes in the subfield (0 to 16).

NONE  
Specifies that no report name is to be selected.

**Default** The default is NONE.

**TCOUNT**

---

Specifies the total number of consecutive trailer banner pages needed for banner page detection.

**Syntax** **BANNER TCOUNT=*value***

**Parameter options** *value*  
For online jobs, specifies the total number of consecutive trailer banner pages which must be detected to satisfy the test expression.

**Default** The default is 0.

**Considerations**

Table 4-1 shows the effect of the TCOUNT and HCOUNT parameters under the listed conditions.

Table 4-1. **TCOUNT and HCOUNT conditions and results**

Condition	Parameter	Result
LPS positions to TOF after end-of-report processing.	TCOUNT=0	The page after the first header is positioned at TOF.
	TCOUNT 0	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.	TCOUNT 0 and HCOUNT 0	LPS assumes that the report has fewer header pages than were specified and begins looking for trailer pages.

**TEST**

Defines the test expression for detection of a banner page for either change mode or constant mode criteria for online jobs.

**Syntax**

**BANNER TEST=***test-exp*

**Parameter options**

*test-exp*

If a *test-exp* is satisfied, the page containing the record tested is considered a banner page. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**

There is no default.

**TYPE**

---

Specifies the use of selected data pages as banner pages when reports do not contain this information.

**Syntax**  
**Parameter options**

**BANNER** *TYPE=value*

*value*

**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.

**DATA**

For online jobs, use when reports do not contain user-specified banner pages. This parameter allows you to treat selected data pages 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.

**Default** The default is BANNER.

---

**BDELETE**

---

Allows you to define the block deletion test. You can also use this command to selectively delete specialized blocks, such as control blocks and unsupported labels that are on the data tape but are not to be printed.

**TEST**

---

Deletes printing any interspersed blocks within an offline report or file.

**Syntax**  
**Parameter options**

**BDELETE** *TEST=test-exp*

*test-exp*

Defines the test expression for selecting blocks for printing. The block is deleted for printing if a *test-exp* is satisfied. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default** There is no default.

**Considerations**

Block deletion is performed before the extraction of the records from the block. If a block is deleted from 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 CRITERIA CONSTANT or CHANGE parameters, specify offsets to subfields of a block in bytes, relative to zero from the start of block to the beginning of subfield.

When you use XDDI Release 1.0, you can only use the TEST parameter with the CRITERIA command in the CONSTANT mode.

## BSELECT

Allows you to define the block select test. You can also use this command to selectively delete specialized blocks such as control blocks and unsupported labels, that are on the data tape but are not to be printed.

### TEST

Selects any interspersed blocks within an offline report or file for printing.

**Syntax**  
**Parameter options**

**BSELECT** *TEST=test-exp*

*test-exp*

Defines the test expression for selecting blocks for printing. The block is selected for printing if a *test-exp* is satisfied. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**  
**Considerations**

There is no default.

Block selection is performed before the extraction of the records from the block. If a block is 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 CRITERIA CONSTANT or CHANGE parameters, specify offsets to subfields of a block in bytes, relative to zero from the start of block to the beginning of subfield.

When you use XDDI, you can only use the TEST parameter with the CRITERIA command in the CONSTANT mode.

### BSELECT example

The following commands illustrate the use of BSELECT to process interspersed reports on a block basis, as shown in figure 4-1.

```
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 4-1. Sample BSELECT and BDELETE command usage

---

**ac:CRITERIA**

---

Allows you to define logical processing text specifications.

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 of the type *ac* that can be specified in any of the logical processing commands as a TEST parameter. The command identifier *ac* consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter.

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 with the following results:

- 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.

The CRITERIA tables may specify either change mode or constant mode functions. There are no restrictions on their usage or combination. However, there are two special cases:

- When the record or block is too short to include the field being tested
- When the line being tested contains no record (has been skipped).

If the test specifies a constant mode function, the CRITERIA fails.

If the test specifies a change mode function, the CRITERIA fails because no change has occurred but the value for the LAST option of the CHANGE parameter is unchanged for comparison with the next record.

An 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 contained criteria 1 and line 2 contains criteria 2, that is, criteria 1 and criteria 2 are not equal to LAST.

## CHANGE

Defines test specifications for a logical processing function with change mode criteria. Note that change mode criteria are not valid for logical block processing.

### Syntax

*ac*:**CRITERIA CHANGE**=(*offset,length,NE,LAST*)

### Parameter options

#### *offset*

Specifies the offset in bytes (relative to zero) from the start of the user portion of the record to the control field within the record.

#### *length*

Specifies 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. If a control field of the current record 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 that 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 the test is met.

### Default

There is no default.

### Considerations

You must specify the length and location of a control field in each record. When the content of the control field of one record differs from the content of the control field of the previous record, the CRITERIA command is true.

## CONSTANT

---

Defines test specifications for a logical processing function with constant mode criteria in block processing.

**Syntax**

*ac:CRITERIA CONSTANT=(offset,length,equal,tab-id)*

**Parameter options***offset*

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 portion of the record to the field in the record being compared.

*length*

The length in bytes of the test field. Its range is 1 to 255.

*equal*

NE

Indicates not equal to.

EQ

Indicates equal to.

*tab-id*

The identifier of a TABLE command.

**Default**

There is no default.

**Considerations**

You must specify the length and contents of a fixed field within a user record or block. Each user record or block is examined at the specified location to determine if the constant is present (the identifier *table-id* defines the table containing the constant). If present, the CRITERIA command is true. If not, the command is false.

## LINENUM

---

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**

*ac:CRITERIA LINENUM=(init,count)*

**Parameter options***init*

An integer specifying the number, on each page, of the beginning line for which the CRITERIA command is evaluated.

*count*

An integer specifying the consecutive number of lines (starting with *init*) for which the CRITERIA command is evaluated.

**Default**

The default is all lines.



## VALUE

Defines test specifications for a logical processing function with value mode criteria. Note that value mode criteria are not valid for logical block processing.

Use the CRITERIA VALUE parameter to compare two numeric values. The system returns a true value if the variable text data satisfies the test criteria.

### Syntax

*ac*:**CRITERIA VALUE**=(*offset,length,operator,tab-id*)

### Parameter options

#### *offset*

Specifies the offset in bytes (relative to zero) from the beginning of the user portion of the record to the beginning of the test field.

#### *length*

Specifies the length in bytes of the test field (from 1 to 255). Note that the length may differ from the length of the associated TABLE CONSTANT string(s).

#### *operator*

Specifies the comparison operation. Available options are:

EQ	Equal
NE	Not equal
LT	Less than
GT	Greater than
LE	Less than or equal to
GE	Greater than or equal to.

#### *tab-id*

The identifier of a TABLE statement.

### Considerations

A test criteria specifying the EQ operator may return a true result even though the character strings being compared are not identically equal. For example, '0000' is equal to '0','0','\$0.00', and so forth.

A numeric character string will be compared with the constant character string only if it is properly constituted. The following rules will be observed by Input processing when evaluating a numeric character string:

- A properly constituted numeric character string should consist of a single sequence of numeric characters which may be interspersed only with characters from a rigidly defined set of "separator" characters which are allowed in the format of a decimal number. The following strings: '1, 2, 3,', '957N4218', and '00 -123', are examples of improperly constituted numeric character strings.
- All numeric character strings will be evaluated as decimal numbers with a decimal point separating the integer portion of the number from the fractional portion of the number. Due to multinational differences in decimal number formats, the character(s) that is (are) interpreted to be the decimal point, and the character(s) that is (are) interpreted to be the "thousandths" separator will be taken from the system defined VCODE table specified in the VOLUME statement. The VCODE command is used to specify the character translation code and/or the character type code, and must be specified if the VCODE command defaults are not applicable.

- Non-numeric characters, such as currency symbols, positive and negative number designators, and alphabetic text may only precede, follow, or enclose the numeric character string. If the appropriate VCODE table is specified: '\$(1,500.00)', '-1.500,00 DM', 'kr -1 500,00' are valid numeric character strings.
- A numeric character string will be evaluated as a negative number if a minus sign (-) either precedes or follows the string, or if the string is enclosed in a single set of parentheses. Because of the simplified procedure used to determine negative numbers, occurrences of more than one opening parenthesis preceding the string, occurrences of more than one closing parenthesis following the string, or occurrences of more than one minus sign will invalidate the string.
- Leading zeros in a numeric character string do not affect the value of the string, and trailing zeros in the fractional part of a numeric character string likewise do not affect the value of the string. The decimal point character separates the integer part of a number from the fractional part. If a decimal point is not present, there is an implied decimal point at the end of an integer number. The thousandths separator is allowed within a numeric character string only if it is placed between groups of three digits going away from the decimal point. A decimal point or a thousandths separator may appear repeatedly outside the numeric character string. In VCODE0, the following are examples of valid numeric character strings:  
'50,000', '0,000,50', ',,42,', '1.000,00', '.....5'.

The CONSTANT specified in the TABLE statement that is associated with a CRITERIA VALUE statement may not be specified together with the MASK command.

When more than one constant character string is specified in the TABLE statement and if the operator is:

- EQ, the system tests the variable data against all the values in the TABLE CONSTANT statement. If any of the values are equal, the system returns a true value.
- NE, LT, GT, LE, or GE, the system tests the variable data against only the first value in the TABLE CONSTANT statement.

### Performance Considerations

CRITERIA VALUE affects performance in relation to the number of characters per page involved, which is a function of the number of tests performed, the number of character columns in each test and the number of lines in which the criteria is evaluated.

To reduce the impact of CRITERIA VALUE processing on the throughput of your applications, you may consider the following:

- Restrict the range of lines tested via the LINENUM parameter to avoid the time wasted testing lines which are not supposed to meet the criteria. For a forms application with an address field at the top, and mailing information at the bottom, you might use the LINENUM to avoid tests on the data associated with the address label and mailing info.
- Restrict the columns tested in the CRITERIA VALUE command to the shortest possible length to avoid testing white space. At the same time be cautious ensuring that all possible values are covered. Do not restrict the columns to hold only

\$999,999 if there is a chance that \$1,000,000 might occur on a rare occasion.

- For applications which either CRITERIA VALUE or CRITERIA CONSTANT can be used, choose CRITERIA CONSTANT, as it has less performance impact.
- When coding numeric values into TABLE CONSTANTS, for maximum performance, specify the shortest possible numeric string. Omit superfluous leading zeros, trailing zeros, white space, and thousands separators. For example, '50000' is faster to evaluate than '050,000.00', since the former expresses the same value using only half as many characters as the latter.

## LMODIFY

This command allows you to add highlight color to black-only applications. It is ignored by the Xerox 4050/4090/4450/4650 LPS.

### TEST

#### Syntax

**LMODIFY** *TEST=test-exp*,*INK=ink-index*,*SELECT=LINE*

or

**LMODIFY** *TEST=test-exp*,*INK=ink-index*,*SELECT=(offset,length)*

#### Parameter options

*test-exp*

Specifies the test to be performed using either one or two criteria.

*ink-index*

Specifies the ink to be used to print the text data. The *ink-index* is an index number which refers to the current ink list (ILIST). If the INK command option is not specified, the currently applicable ink would be applied.

*select-op*

Specifies the part of the line to be printed with the selected ink. The options for *select-opt* are as follows:

#### LINE

Prints the entire line using the selected ink.

*(offset,length)*

Specifies an offset value in bytes (relative to zero) from the beginning of the print line for *length* characters to be printed using the selected ink.

#### Example

If two test criteria are used in the *test-exp* parameter, they are linked by an AND or OR operator.

test-exp: = *cri-id*

test-exp: = (*cri-id1* AND *cri-id2*)

test-exp: = (*cri-id1* OR *cri-id2*)

The logical OR function is also implied when multiple character strings exist in the TABLE CONSTANT statement.

#### Considerations

The LMODIFY command acts upon individual print lines. The offset values in all of the test criteria are measured in bytes (relative to zero) from the beginning of the print line to the beginning of the test field. It should be noted that this is a marked departure from previously established usages of the offset as a measure in bytes (relative to zero) from the beginning

of the user portion of the record. The print data offset in the DATA parameter of the LINE command can affect the relationship between the beginning of the input data record and the beginning of the print line.

Copy modification entries (CMEs), or FONTINDEX (and optionally INKINDEX) in conjunction with OVERPRINT=MERGE will override LMODIFY.

## RAUX

---

Allows you to define a select page from an auxiliary tray.

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 should be used with this parameter to prevent the auxiliary tray from being automatically selected when the main tray is empty.

This command can also be used in combination with the OUTPUT command FEED parameter and the DJDE FEED.

## TEST

---

Specifies the test expression for selection of paper from the auxiliary tray.

**Syntax**  
**Parameter options**

**RAUX TEST=***test-exp*

*test-exp*

If a *test-exp* is satisfied, a page is selected from the auxiliary tray. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**  
**Considerations**

There is no default.

If a record intended to satisfy the criteria for RAUX is suspended by RSUSPEND, that record is not checked for the RAUX criteria. Note that the RAUX criteria does not take effect when printing is resumed (offline only).

Record selection or deletion is performed prior to RAUX processing. If a record satisfying the RAUX test criteria was previously not selected for or deleted from printing, the RAUX does not function (offline only).

Since DJDE processing occurs prior to RAUX processing, DJDE records are not checked for satisfying the RAUX criteria (offline only).

---

**RAUX example**

---

```
T1:  TABLE      CONSTANT='CUSTOMER COPY';
C1:  CRITERIA    CONSTANT=(20, 13, EQ, T1);
      RAUX        TEST=C1;
```

When processed as part of the input data stream, this RAUX command causes the page it is a part of to be printed on a sheet of paper fed from the auxiliary tray. The Xs are depicted in the example to illustrate the specified offset of 20 character positions.

---

**RDELETE**

---

Allows you to delete interspersed records for printing within one report or file. You can also use this command to selectively delete specialized records, such as control records and offset records, that reside on data tape but are not to be printed.

**TEST**

---

Defines a test expression for deletion of records from printing.

**Syntax**

**RDELETE** *TEST=test-exp*

**Parameter options**

*test-exp*

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. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**

There is no default.

**Considerations**

When defining CRITERIA CONSTANT or CHANGE parameters, specify offsets to the subfields of the records in bytes (relative to zero) from the start of the user portion of the record to the beginning of the subfield.

Record 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 deleted from printing, the RAUX, suspend, or resume do not function.

---

**RDELETE example**

---

The following example illustrates the use of RDELETE to process interspersed reports on a record basis.

```
T1:  TABLE      CONSTANT=('EFGH');
C1:  CRITERIA    CONSTANT=(104, 4, EQ, T1);
      RDELETE     TEST=C1;
```

If the contents of the 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 as shown in figure 4-2, where records 2 and 6 would not be printed.

Figure 4-2. **Sample RDELETE command usage**

---

## RFEED

---

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.

---

### TEST

---

Specifies paper fed from different clusters if certain criteria are met on a record basis.

<b>Syntax</b>	<b>RFEED TEST=(<i>test-exp</i>,<i>clu-def</i>)</b>
<b>Parameter options</b>	<i>text-exp</i> References a criteria table to be used. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of <i>test-exp</i> .  <i>clu-def</i> Defines the cluster to use for the page, either a <i>cluster-name</i> or a <i>cluster-reference</i> .
<b>Default</b>	There is no default.
<b>Considerations</b>	Satisfaction of the RFEED criteria will cause the current page to feed from the cluster defined in the RFEED command.  If there are multiple RFEED criteria satisfied for a given page, the last criterion will override any previous ones.  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.

**ROFFSET**

Provides the capability for online and offline users to initiate a page offset in the stacker tray under control of the input data (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 example, the report is not terminated and multiple copies are not 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 that satisfies the ROFFSET test is printed on a page of the output, that page is the offset sheet.
- If the record that 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.

**PASSES**

Specifies a page offset for reports.

**Syntax**  
**Parameter options**

**ROFFSET** *PASSES=value*

*value*

ALL

Specifies that the satisfied criteria causes an offset on all passes of a collated print run.

FIRST

Specifies that the satisfied criteria causes an offset only on the first pass of a collated print run.

**Default**  
**Considerations**

The default is ALL.

If ROFFSET is specified for an uncollated job, an offset occurs on only the first copy of the offset page.

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.

**TEST**

Defines a test expression for offsetting pages to the stacker tray.

**Syntax**

**ROFFSET** *TEST=test-exp*

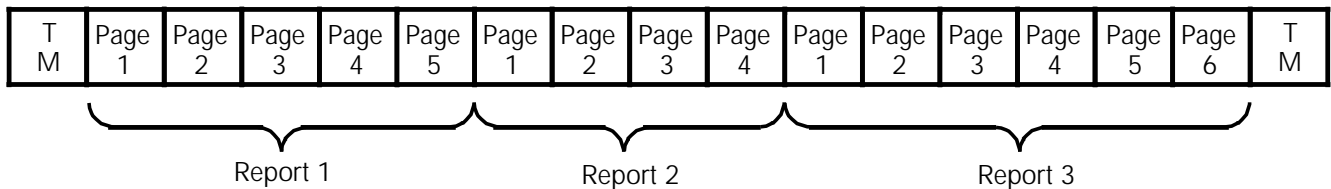
<b>Parameter options</b>	<p><i>test-exp</i></p> <p>If a <i>test-exp</i> is satisfied, the record causes an offset in the stacker tray. If a <i>test-exp</i> is satisfied in duplex mode, the logical page on which the record occurs is the first logical page of a new sheet. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of <i>test-exp</i>.</p>
<b>Default</b>	There is no default.
<b>Considerations</b>	<p>The ROFFSET feature prints the record satisfying the test expression according to the normal job parameters.</p> <p>If a record satisfying the ROFFSET test criteria is not selected for or deleted from printing, the offset indication is maintained and used to cause the offset on the next record printed.</p>

### ROFFSET example

A file has multiple reports without any delimiters separating the reports, as illustrated in figure 4-3. 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 4-3. Sample ROFFSET command parameters



### RPAGE

Allows you to reposition the current or next logical page.

#### SIDE

Defines the side of the physical sheet on which the repositioned logical page is to appear.

<b>Syntax</b>	<b>RPAGE SIDE=side,offset</b>
<b>Parameter options</b>	<i>side</i>

In duplex (with no BFORM), *side* produces the following results:  
**NUFRONT**  
 Specifies the logical page is positioned as the first logical page on a new sheet.



**BACK**

Specifies the logical page is positioned as the first logical page of the next available back.

**NUBACK**

Specifies the logical page is positioned as the first logical page on the back of a new sheet.

**NEXT**

Specifies the 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.

The logical page is moved unless it is already properly positioned as the first logical page on the specified side.

*offset***NOFFSET**

No offset occurs.

**OFFSET**

The sheet on which the repositioned logical page occurs is offset in the output stacker.

**Default**

The default is NUFROnt,NOFFSET. 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.

**Example** Figure 4-4 illustrates SIDE parameter options.

Figure 4-4. **SIDE parameter options**

---

**TEST**

---

Defines the test expression for detection of an RPAGE record.

**Syntax**

**RPAGE TEST=***test-exp*

**Parameter options**

*test-exp*

The logical page on which this record normally prints is called the current logical page. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**

There is no default.

**Considerations**

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 (offline only).

Record selection or 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 (offline only).

Since DJDE processing occurs prior to RPAGE processing, DJDE records are not checked for satisfying the RPAGE criteria (offline only).

---

**WHEN**

---

Defines the portion of the current logical page to be positioned to the first logical page of a new location.

**Syntax**

**RPAGE WHEN=***value*

**Parameter options**

*value*

TOP

The current logical page is repositioned.

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.

**Default**

The default is TOP.

**Considerations**

Figure 4-5 illustrates WHEN parameter options.

Figure 4-5. **WHEN** parameter options

**RRESUME**

Allows you to delete groups of records from printing that are distinguishable at the start and end, but whose intermediate records may not be unique or distinguishable. The command can specify the full range of tests as described previously for the other logical processing commands.

When specifying the RRESUME command, you can specify whether 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.

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.

**BEGIN**

Specifies on which record printing resumes.

<b>Syntax</b>	<b>RRESUME</b> <i>BEGIN=value</i>
<b>Parameter options</b>	<p><i>value</i></p> <p>NEXT If NEXT is coded, the record satisfying <i>test-exp</i> is not printed, and printing begins with the next record.</p> <p>CURRENT Specifies whether printing resumes on the current or next record for offline jobs. If CURRENT is coded, the record is printed.</p>
<b>Default</b>	The default is NEXT.

**TEST**

Defines the test expression for the resumption of printing following print suppression. If the *test-exp* is satisfied, the record is used to resume printing following print suppression.

<b>Syntax</b>	<b>RRESUME</b> <i>TEST=test-exp</i>
<b>Parameter options</b>	<p><i>test-exp</i></p> <p>If the <i>test-exp</i> is satisfied, the record is used to resume printing following print suppression. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of <i>test-exp</i>.</p>
<b>Default</b>	There is no default.
<b>Considerations</b>	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 RRESUME processing. If a record satisfying the RRESUME test criteria was not selected for or deleted from printing previously, it does not cause either the suspend or resume function.

The records after print resumption should have compatible Printer Carriage Control (PCC) characters. No additional carriage control characters are inserted by the system during the print suppression.

DJDE records are processed, not ignored, even when detected before the corresponding RRESUME record.

Consider the following when using RRESUME with the RSTACK command:

- A record satisfying the RSTACK test can still be found and terminates the report, even if record printing is suspended at the time. Also, DJDE records are not processed even if record printing has been suspended.
- 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.

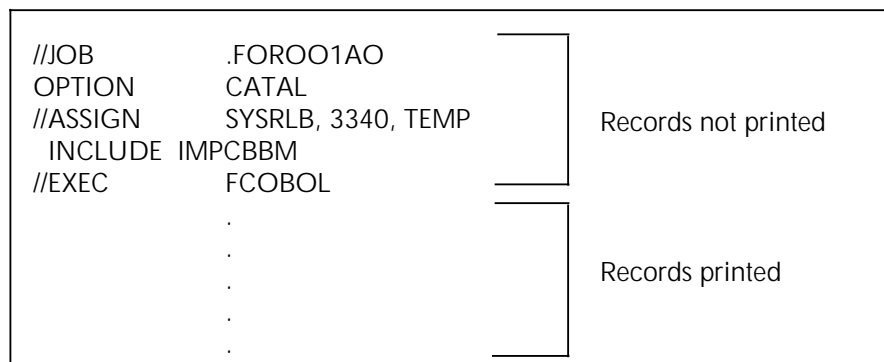
Use of the LINENUM parameter in the CRITERIA command is not recommended when using RRESUME. Since the carriage control characters are not processed during the print suppression, the line number used by the system is that which existed when the suppression started. This could result in the test criteria being either unexpectedly satisfied or never satisfied as a function of the line number where the print suppression started.

### RRESUME example

```
T1:  TABLE      CONSTANT=('//JOB', '//EXEC');
T2:  TABLE      CONSTANT='EOJ';
T3:  TABLE      CONSTANT=('//EXEC');
C1:  CRITERIA    CONSTANT=(1, 6, EQ, T1);
C2:  CRITERIA    CONSTANT= (1, 4, EQ, T2);
C3:  CRITERIA    CONSTANT=(1, 6, EQ, T3);
      RSUSPEND   TEST=(C1, OR, C2),
                        BEGIN=CURRENT;
      RRESUME    TEST=(C3, OR, C2),
                        BEGIN=NEXT;
```

Printing of input data records in figure 4-6 is suspended (from record //JOB to //EXEC) by the preceding RSUSPEND command. Printing resumes after the //EXEC data record (as per the RRESUME command above).

Figure 4-6. **Sample RRESUME and RSUSPEND data**



---

**RSELECT**

---

Allows you to select interspersed records for printing within one report or file. You can also use this command to selectively delete specialized records, such as control records and offset records, that reside on data tape but are not to be printed.

**TEST**

---

Defines a test expression for the selection of records for printing.

**Syntax**  
**Parameter options**

**RSELECT** *TEST=test-exp*

*test-exp*

If a *test-exp* is satisfied, the record is selected for printing. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**  
**Considerations**

There is no default.

When defining CRITERIA CONSTANT or CHANGE parameters, specify offsets to the subfields of the records in bytes (relative to zero) from the start of the user portion of the record to the beginning of the subfield.

Record selection is performed offline prior to RAUX and suspend or resume. If a record satisfying either the RAUX suspend or resume test criteria was previously not selected for printing, the RAUX, suspend, or resume do not function.

---

**RSTACK**

---

Allows you to define a series of reports in a single file.

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 a normal end of report. 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.

---

**ACCTINFO**

---

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.

<b>Syntax</b>	<b>RSTACK ACCTINFO=(<i>offset,length</i>)</b>
<b>Parameter options</b>	<i>offset</i> Specified in bytes (relative to zero) from the start of the user portion of the record to the subfield within the record. <i>length</i> The number of bytes in the subfield (1 to 64).
<b>Default</b>	There is no default.

---

**DELIMITER**

---

Specifies whether all or single records are part of a subsequent report.

<b>Syntax</b>	<b>RSTACK DELIMITER=<i>value</i></b>
<b>Parameter options</b>	<i>value</i> 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.
<b>Default</b>	The default is YES.
<b>Considerations</b>	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.



---

**HRPTNA**

---

Specifies that a subfield of the first record is being displayed as REPORT NAME on the LPS console in response to the operator JOBS command or <STATUS> key.

**Syntax** **RSTACK** *HRPTNA*=(*offset,length*)  
or  
**RSTACK** *HRPTNA*=NONE

**Parameter options**

*offset*

Specifies in bytes (relative to zero) from the start of the user portion of the record to the subfield within the record.

*length*

The number of bytes in the subfield (1 to 16).

NONE

Specifies that no report name is selected.

**Default**

The default is NONE.

---

**PRINT**

---

Allows you to specify, if DELIMITER=YES is coded, if the report delimiters are to be printed, and if so, the output destination of the printed delimiters.

**Syntax** **RSTACK** *PRINT*=*value*

**Parameter options**

*value*

NONE

Specifies that report delimiters are not 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.

BIN

Specifies that report delimiters are printed and the output is delivered to the output stacker tray.

BOTH

Specifies that report delimiters are printed and the output is delivered to both the sample print tray and output stacker tray. The parameter BOTH in duplex is treated as BIN.

TRAY

Specifies that report delimiters are printed and the output is delivered to the sample print tray.

**Default**

The default is NONE.

**TEST**

---

Defines a test expression for end-of-report conditions.

**Syntax**

**RSTACK** *TEST=test-exp*

**Parameter options**

*test-exp*

Defines a test expression for end-of-report conditions for either change mode or constant criteria. If a *test-exp* is satisfied, the record specifies an end-of-report condition. Refer to the "Test expression definition" section of the "Overview" chapter for more information on the format and syntax of *test-exp*.

**Default**

There is no default.

**Considerations**

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 that 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.

**RSTACK example**

---

```
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 figure 4-7. The delimiter record is not printed with the report.

Figure 4-7. **Sample RSTACK command**

**RSUSPEND**

Allows you to delete groups of records from printing that are distinguishable at the start and end, but whose intermediate records may not be unique or distinguishable. Each of the commands can specify the full range of tests as described previously for the other logical processing commands.

When specifying the RSUSPEND command, you can also specify whether suspension 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.

Upon encountering a record that 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.

Consider the following when using RSUSPEND with the RSTACK command:

- 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.
- 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.

**BEGIN**

Specifies on which record printing is suspended.

<b>Syntax</b>	<b>RSUSPEND</b> <i>BEGIN=value</i>
<b>Parameter options</b>	<i>value</i> NEXT If NEXT is coded, the record satisfying <i>test-exp</i> is not printed, and printing is suppressed beginning with the the next record.  CURRENT Specifies whether printing resumes on the current or next record for offline jobs. The record satisfying <i>test-exp</i> does not print.
<b>Default</b>	The default is NEXT.

**TEST**

---

Defines a test expression for the suspension of printing following print suppression. If the *test-exp* is satisfied, the record is used to resume printing following print suppression.

**Syntax****RSUSPEND** *TEST=test-exp***Parameter options***test-exp*

If the *test-exp* is satisfied, the record is used to resume printing following print suppression. Refer to the "Test expression definition" section of the "Overview" chapter for more information on format and syntax of *test-exp*.

**Default**

There is no default.

**Considerations**

Make sure that if an RSUSPEND command is coded, an RRESUME command is also 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.

If a data record satisfying the test expression in the RSUSPEND is encountered, printing is suspended.

Record selection or deletion is performed before suspend processing. If a record satisfying the suspend 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 should have compatible Printer Carriage Control (PCC) characters. No additional carriage control characters are inserted by the system during the print suppression.

DJDE records are processed, not ignored, even when detected after an RSUSPEND record.

**ac:TABLE**

---

Allows you 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 TABLE command requires an identifier of the type *ac* that can be specified in any of the logical processing commands as a TEST parameter. The command identifier *ac* consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be a letter.

---

## CONSTANT

Specifies the content of one or more string constants.

<b>Syntax</b>	<i>ac</i> :TABLE CONSTANT=( <i>sc</i> <sub>1</sub> , <i>sc</i> <sub>2</sub> ,...)
<b>Parameter options</b>	<p><i>sc</i></p> <p>Specifies string 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.</p>

**Default** There is no default.

**Considerations** The default character set is EBCDIC. Character strings must be preceded by a letter and a single quote, except EBCDIC, for which the E is optional:

EBCDIC: TABLE CONSTANT=(E'REPORT');

or

TABLE CONSTANT=('REPORT');

ASCII: TABLE CONSTANT=(A'REPORT');

Octal: TABLE CONSTANT=(O'REPORT');

Hexadecimal: TABLE CONSTANT=(X'REPORT');

Refer to the "String constants" section in the "Overview" chapter for additional information.

---

## MASK

Defines mask characters used within string constants (specified by the CONSTANT parameter) and indicates which character positions have special type testing and what that testing is.

<b>Syntax</b>	<i>ac</i> :TABLE MASK=( <i>ignore-char</i> , <i>charspeci</i> <sub>1</sub> , <i>charspeci</i> <sub>2</sub> ,...)
<b>Parameter options</b>	<p><i>ignore-char</i></p> <p>Specifies a character to appear in those positions within a string constant which are not to be compared.</p>

*charspeci*

Specifies a character that appears in those positions within a string constant where a test for type *i* (*i*=index number) characters is to be performed. A *charspec* is of the form: '*char*' or ('*char*','*char*,...' *char*'). Mask characters should be defined using the same string type as in the string specified in the CONSTANT parameter.

**Default** There is no default.

**ac:TABLE example**

**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: xx9x, 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 specify the format x999x9 within the input data stream would be the following:

```
T1:  TABLE  MASK=('?', '%', '@');
```

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. The mask character-to-type associations made for this TABLE command are shown in table 4-2.

Table 4-2. **Example 2 mask character-to-type associations**

Character	Type	Meaning
?	None	Make no comparison
%	1	Standard default, any numeric (0-9)
@	2	Standard default, any alphabetic (A-Z, a-z)

**Example 3** T2: TABLE MASK=('?', '%', '@'), CONSTANT=('A7%%@%');

In this example, no mask character (as specified in the MASK parameter) is found when the first two character positions of the CONSTANT are checked. For those two positions, exact character matches between the input data string characters and the CONSTANT parameter characters are required. In this example, only data strings that begin with A7 can pass the entire test.

**Example 4** T2: TABLE MASK=('\*', ':', '!'),  
CONSTANT=('DATE : : / : : / : : ID ! ! !');

This example sets up the mask character-to-type associations shown in table 4-3.

Table 4-3. Example 4 mask 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)





---

## 5. Print control (DJDE) commands

Dynamic Job Descriptor Entries (DJDEs) are parameter keywords embedded within the input data stream. You use them to modify a printing environment that was established by a Job Descriptor Entry (JDE). 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 Job Descriptor Entry (JDE) notifies the system that DJDE records are included in the input data stream.

---

### ALTER

Specifies the new imaging parameters for graphics previously referenced using the hold (automatic reimaging) parameter. ALTER is a record-oriented DJDE relating to graphics.

#### Syntax

**ALTER**=(*name,vpos,hpos,n/d*)

#### Parameter options

##### *name*

Identifies a previously called out graphic (refer to DJDE IMAGE parameter) that 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.

##### *vpos*

Specifies the vertical position of the top edge of the graphic, relative to 0,0 on the current physical page. Available *vpos* options are: UN (user-defined units), CM (centimeters), IN (inches specified as a decimal number with up to three digits to the right of the decimal point), DOTS, or XDOTS. 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. If no units are specified, inches are assumed.

##### *hpos*

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) for *hpos* is the same as for *vpos*.

##### *n/d*

Specifies the reference scale factor.

#### Default

There is no default.

#### Example

```
ALTER=(BUS,5,5,1);  
ALTER=(IMG 1,3,5),END;
```

## ASSIGN

---

		Assigns a VFU channel to a page line number or set of line numbers. ASSIGN is a record-oriented DJDE.
<b>Syntax</b>	<b>ASSIGN=(channo,lineno,...)</b>	
<b>Parameter options</b>	<i>channo</i>	Specifies the number of the channel being assigned. It is an integer in the range of 0 to 15. You 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>	Specifies 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.
<b>Default</b>		There is no default.
<b>Considerations</b>		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.

## BATCH

---

		Permits normal processing on online banner pages for batch mode jobs by delimiting batch mode graphic data. BATCH is a record-oriented DJDE.
<b>Syntax</b>	<b>BATCH=action</b>	
<b>Parameter options</b>	<i>action</i>	
	START	The first graphic (or IMAGE DJDE) must be preceded immediately by a BATCH=START DJDE. A BATCH=START DJDE may be followed immediately (no intervening records) by an IMAGE DJDE. This parameter, when specified, must be the only parameter in the DJDE packet.
	END	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.
<b>Default</b>		There is no default.
<b>Considerations</b>		You must mark precisely the beginning and ending of online batch mode data.

**BEGIN**

	Specifies the location of the starting print line of a logical page for graphics. BEGIN is a page-oriented DJDE.
<b>Syntax</b>	<b>BEGIN</b> =( <i>vpos</i> , <i>hpos</i> )
<b>Parameter options</b>	<p><i>vpos</i> Specifies the vertical position of the first character of the first print line on the logical page. It is a decimal number with up to three digits to the right of the decimal point, for example, 0.563 IN and 2.35 CM are all legal specifications. The <i>vpos</i> units are specified inches (IN), or centimeters (CM).</p> <p><i>hpos</i> Specifies the horizontal position of the first character of the first print line on the logical page. All specifications are rounded to the nearest dot (1/300 of an inch) for positioning of the logical page. The <i>hpos</i> units are specified inches (IN), or centimeters (CM).</p>
<b>Default</b>	The default is (.18IN,.66IN). The default unit specification is inches (IN).
<b>Considerations</b>	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**

	Specifies that a form is printed on the back side of a printed page (must have DUPLEX=YES). BFORM is a page-oriented DJDE. It takes effect as soon as the data is read into the system. BFORM parameters are the same as for the OUTPUT command.
<b>Syntax</b>	<b>BFORM</b> =NONE or <b>BFORM</b> = <i>form-id</i> or <b>BFORM</b> =( <i>form-id</i> , <i>init</i> , <i>copies</i> )
<b>Parameter options</b>	<p>NONE Does not add a form to the associated report page of variable data.</p> <p><i>form-id</i> Specifies a filename which exists on disk. This file is created by compiling an FSL source file with the FDL system task.</p> <p><i>init</i> Specifies the beginning ply number to which a specified form applies. This defaults to the first or next copy.</p> <p><i>copies</i> Specifies the number of plies (passes) to which a specified form applies.</p>
<b>Default</b>	The default is NONE.
<b>Example</b>	<b>BFORM</b> =(FORM 1,2),END;

## BOF

---

		Specifies the bottom-of-form (BOF) line number. BOF is a record-oriented DJDE.
<b>Syntax</b>	<b>BOF=</b> <i>value</i>	
<b>Parameter options</b>	<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). The bottom-of-form specification is independent of the channel assignments. Bottom-of-form should be greater than or equal to the largest line number assigned to a channel. It takes effect at the next record following a DJDE END command. The maximum value for BOF is 255.
<b>Default</b>		The default is 66.
<b>Considerations</b>		<p>If the TOF DJDE is specified as a value greater than or equal to the BOF value, the TOF value is set to the value of 1.</p> <p>If bottom-of-form 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.</p> <p>If you specify a line number that is greater than the current line number or exceeds BOF (i.e., exceeds the maximum value of 255), the print line skips to the next page and line spacing is continued from the top-of-form line number of the next page. BOF is ignored if you have Honeywell 2000 (H2000) or Xerox carriage control processing.</p> <p>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.</p>

## C text

---

		Specifies comment text in the DJDE record. <i>C text</i> is a record-oriented DJDE.
<b>Syntax</b>	<b>C</b> <i>text</i>	
<b>Parameter options</b>	<i>text</i>	Allows you to include comment text 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.
<b>Default</b>		There is no default.

## CANCEL

---

		Cancels the hold (automatic reimaging) specified by the H option of an IMAGE, GRAPHIC, or LOGO parameter, and is effective on the current page. CANCEL is a record-oriented DJDE.
<b>Syntax</b>	<b>CANCEL=</b> ( <i>name</i> <sub>1</sub> , <i>name</i> <sub>2</sub> ,...) or <b>CANCEL=</b> <i>ALL</i>	

<b>Parameter options</b>	<i>name</i> Specifies the graphics or logos to be canceled. Since no file type is given, if a graphic and a logo have the same name, both are canceled.
	ALL Specifies that all graphics or logos being held are to be canceled. You may also cancel PDL-invoked logos in this manner.
<b>Default</b>	There is no default.

**COLLATE**

	Specifies whether the pages of a job are collated (placed into sets) or uncollated (placed together). COLLATE is a page-oriented DJDE.
<b>Syntax</b>	<b>COLLATE=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> YES Specifies that report copies are printed in collated mode. It is enabled at the next page boundary but must occur prior to the first data record of the report.
	NO Specifies that report copies are to be printed in collated mode.
<b>Default</b>	The default is YES.

**COPIES**

	Specifies the number of copies or sets of the pages to produce at the next page boundary. COPIES is a page-oriented DJDE.
<b>Syntax</b>	<b>COPIES=</b> <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies the number of copies or sets of the pages 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 <i>value</i> is set to zero, this copy count remains in effect until a new copy count or end of report is encountered. DJDEs encountered while <i>value</i> is zero in effect are processed normally.
<b>Default</b>	The default is 1.
<b>Considerations</b>	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 too quickly.  Follow these steps to avoid the problem:  1. Set the number of copies of the first page to 2 or higher. 2. Include a copy-sensitive Copy Modification Entry (CME).

3. Use the COPIES parameter in the START command to override the DJDE parameter, COPIES=.

## DATA

---

		Specifies the location and length of printable data within an input record. DATA is a record-oriented DJDE relating to graphics.
<b>Syntax</b>	<b>DATA=(<i>pdo</i>,<i>length</i>)</b>	
<b>Parameter options</b>	<i>pdo</i>	Print data offset. Specifies 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.
<b>Default</b>		For offline systems, the default is (1,132). For online systems, the default is (0,150).
<b>Considerations</b>		For HOST=RSX11(FLX), the DATA parameter must be included in the user's JSL since the default <i>pdo</i> value causes the first data byte to be skipped.

## DEPT

---

		Specifies the accumulation of accounting statics for reports on a name basis. DEPT is a page-oriented DJDE.
<b>Syntax</b>	<b>DEPT=<i>sc</i></b>	
<b>Parameter options</b>	<i>sc</i>	Specifies the accumulation of accounting statistics for reports on a name basis. It takes effect at the next page boundary.
<b>Default</b>		The default is the JDL name.
<b>Considerations</b>		If there is more than one DEPT DJDE in a report, only the last DEPT name is charged for all copies of the entire report.

## DESTINATION

---

		Specifies the destination for the printed output. DESTINATION is a page-oriented DJDE.
<b>Syntax</b>	<b>DESTINATION=<i>value</i></b>	
<b>Parameter options</b>	<i>value</i>	
	BIN	Directs output to the current active bin.
	TRAY	Directs output to the sample tray.
	EXPORT	Directs output to the Bypass Transport. This option is only valid if the operator has entered the SELECT 1 or SELECT E command at the user interface.

---

**DUPLEX**

---

Specifies whether both sides of a piece of paper are printed (duplex) or only the top side (simplex). DUPLEX is a page-oriented DJDE.

<b>Syntax</b>	<b>DUPLEX=value</b>
<b>Parameter options</b>	<i>value</i> NO Specifies that printing is to occur on a single side of a page. It takes effect on the next page boundary. YES Specifies that printing is to occur on both sides of a page. It takes effect on the next page boundary.
<b>Default</b>	The default is NO.

---

**END**

---

Specifies the end of DJDE information.

<b>Syntax</b>	<b>END;</b>
<b>Parameter options</b>	None.
<b>Default</b>	There is no default.
<b>Considerations</b>	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.

---

**FEED**

---

Controls the stock (type of paper) on which the page image prints. FEED is a page-oriented DJDE.

<b>Syntax</b>	<b>FEED=value</b>
<b>Parameter options</b>	<i>value</i> OPR, MAIN, or AUX Provides compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN. FEED MAIN should be entered before paper feeding using DJDEs.  <i>stock-reference</i> Allows users to change stocks associated with a job without altering the stock references in the data application. References the stock assigned to a <i>stock-name</i> by the STOCKSET command in effect at the time the page is printed.  <i>stock-name</i> Bypasses the reference feature, but still requires that the <i>stock-name</i> is specified as present in the current STOCKSET. (FEED= <i>cluster-reference</i> should always be used.)
<b>Default</b>	The default is OPR.

## FILE

---

DJDE FILE provides the capability of loading card image or LPS-labeled files to the system disk while a print job is in progress. You can invoke a print job solely for the purpose of downloading files. In other situations, you can interleave files with variable text data so that files are available when referenced during the print job.

**Syntax** FILE=(*file-name,file-type,f,s,n*)

**Parameter options**

*file-name*

Specifies a 1 to 6 character name used to identify the disk file.

*file-type*

Specifies a 3 character file type symbol.

*f*

Specifies one of the following file input format parameters:  
C=card-image format  
L=LPS-labeled tape format.

*s*

Specifies one of the following file storage parameters:  
D=delete after report printed  
P=permanent.

*n*

Specifies the maximum number of card images.

**Default**

The default is (*file-name,file-type,L,P,120*)

**Considerations**

FILE is a record-oriented DJDE. You can include FILE with other record-oriented or page-oriented DJDE commands in a DJDE packet. A DJDE packet must immediately precede the data for a file or batch of files for it to copy to the LPS disk. The DJDE packet must include FILE. FILE signals the start of the file data.

Although intended primarily for use with online systems, FILE is not restricted to online use.

The input task handles the creation and deletion of files that 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 completes report setup for the report that follows.

DJDE FILE does not allow a file replacement or deletion transaction to occur if the file is secured and the logon level is not 5. Refer to the *Xerox 4050/4090/4450/4650 System Administration Guide* for further information.

**Processing card-image format files**

When you process card-image format files, the system checks FILE 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 prints on an OPRINFO page with the message:

```
FILE PROCESSING ERROR - ILLEGAL DESTINATION FILE TYPE.
```

Input then proceeds to read and discard the extent of the file.

- If you improperly specify the file name, the preceding DJDE packet is printed on an OPRINFO page 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 predicted by the parameter specifying the maximum number of card images, the extra records are read and discarded. The preceding DJDE packet prints an OPRINFO page with the message:

```
FILE xxx.yyy OVERFLOW - EXCESS CARD IMAGES LOST.
```

- If the file cannot be created because of insufficient space on the disk, the preceding DJDE packet prints an OPRINFO page with the message:

```
NO SPACE ON DISK FOR FILE xxx.yyy.
```

Input then proceeds to read and discard the file.

When the system reads a card-image file, it ignores the PCC byte. The system uses block and record delimiters 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. The system can perform character translation, depending on the JDE in effect.

Any DJDE record serves to terminate card-image record processing for the file being downloaded, as follows:

- If a DJDE comes upon one or more card-image records that are written in the file, the DJDE truncates to an appropriate size and is closed. If OPRINFO=YES, the message:

```
FILE xxx.yyy CREATED
```

is printed on an OPRINFO page.

- If a DJDE comes before any card-image records are filed, the preceding DJDE packet prints on an OPRINFO page with the message:

```
FILE xxx.yyy ERROR - NO CARD IMAGE DATA.
```

The null file is deleted.

File processing mode is concluded, and print mode is resumed with the DJDE record unless the DJDE record contains another FILE parameter.

### Processing LPS-labeled files

For LPS-labeled files, the DJDE FILE parameter keyword may be specified as FILE=(). If the *file-name* or *file-type* specified in the DJDE FILE differs from that in the label record, the file is renamed as specified in the DJDE FILE. The renaming feature for LPS-labeled files allows six restricted types of files to be downloaded as different file types. In this case, the file name and file type are obtained from the label record that precedes the file data.

When you process LPS-labeled files, the system checks FILE for valid parameters, and reads the beginning of the LPS label record. 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 first label record does not meet the requirements of a label record, the preceding DJDE packet prints on an OPRINFO page with the message:

```
FILE PROCESSING ERROR - NO LABEL RECORD.
```

Input then proceeds to process the data variable text data.

If the destination file type is OSD, SAF, SYS, \$Y\$, 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 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 system disk, and file space becomes free after output prints a previous report, input displays the following message and waits for the report to complete before trying again to create the file:

```
DISK FULL INPUT WAITING FOR OUTPUT TO COMPLETE
```

If the file cannot be created due to insufficient space on the system disk, and there is no possibility of disk file space becoming available, the preceding DJDE packet prints on an OPRINFO page with the message:

```
NO SPACE ON DISK FOR FILE xxx . yyy.
```

Input then proceeds to read and discard the extent of the file.

The system reads and stores the files data records 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 OPRINFO=YES and the file transfers successfully conclude, the file name and file type of each file transferred list on the OPRINFO page with a brief message indicating whether the file was newly created or replaces an existing file.

### **Delimiting records for LPS-labeled files**

To prevent loss of data through trailing blank suppression procedures on the host spooler, you must delimit the data records for LPS-labeled files by appending an extra nonblank character at the end of each record. You must use the same nonblank character with all records processed in the file processing mode.

If the nonblank character that is appended at the end of the records is not consistently present throughout the extent of a file, the preceding DJDE packet prints an OPRINFO page with the message:

```
FILE xxx . yyy PROCESSING ABORTED DUE TO DATA ERROR.
```

Input then deletes the partial file the system created and proceeds to process the data as variable text data.

---

**FONTINDEX**


---

	Specifies a particular location in the input record where an index to the specified font is stored. It takes effect on the next logical page boundary. FONTINDEX is a page-oriented DJDE.
<b>Syntax</b>	<b>FONTINDEX</b> =( <i>offset,init-val,bit-opt</i> ) or <b>FONTINDEX</b> = <i>offset</i> or <b>FONTINDEX</b> =NONE
<b>Parameter options</b>	<p><i>offset</i> Indicates the byte offset in the data record where the font index number is to be found.</p> <p><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 <i>bit-opt</i> parameter is used, the <i>init-val</i> 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.</p> <p><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. These low-order bits specify an index value into the font list of the current PDE.</p> <p>NONE Specifies that there is no font index.</p>
<b>Default</b>	The default for <i>init-val</i> is ONE. The default value for <i>bit-opt</i> is 4.
<b>Considerations</b>	Since there is ambiguity between the FONTINDEX and FONTS parameters, if either is abbreviated to the first three letters, the parameter defaults to FONTINDEX.

---

**FONTS**


---

	Specifies the fonts used in input data or variable (CME) data. The FONTS parameter keyword within a DJDE takes effect on the next page boundary. FONTS is a page-oriented DJDE.
<b>Syntax</b>	<b>FONTS</b> =( <i>f<sub>1</sub>,f<sub>2</sub>,...</i> ) or <b>FONTS</b> =( <i>(f<sub>1</sub>,s<sub>1</sub>),(f<sub>2</sub>,s<sub>2</sub>),...</i> ) or <b>FONTS</b> =( <i>(f<sub>1</sub>,s<sub>1</sub> units),(f<sub>2</sub>,s<sub>2</sub> units),...</i> )
<b>Parameter options</b>	<p><i>f</i> Each font index 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 parameter keywords must be used.</p>

**S**  
 Specifies an optional override line spacing value specifying either lines per inch or dots per line to be associated with the font. The maximum value for *s* is 30 LPI; the minimum value is 10 XDOTS or DOTS.

*units*  
 Available *units* are: LPI (lines per inch), XDOTS, and DOTS.

**Default**  
**Considerations**

If *units* is not specified, LPI is 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.

An XDOT is a unit of measurement, which represents 1/600th of an inch. A form specifying XDOTS can be created, edited, and compiled on any LPS with version 3 software, but the form will not print on your Xerox 4050 or 4090 LPS, which is 300 spi.

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 (DJDE) 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 DJDE task sends the error message:

0S6700 SYNTAX ERROR IN DJDE.

## FORMAT

---

Specifies that a new Page Descriptor Entry (PDE) is used for formatting control. **FORMAT** is a page-oriented DJDE.

**Syntax** **FORMAT**=*pde-id*

**Parameter options** *pde-id*  
 Refers to a separately cataloged file in the PDE library on disk.

**Default** There is no default.

**Considerations** Parameters of the selected PDE, such as **BEGIN**, **FONTS**, and **PMODE**, may also be modified on an individual basis.

## FORMS

---

Specifies the form to be merged on the printed pages. **FORMS** is a page-oriented DJDE.

**Syntax** **FORMS**=*form-id*  
 or  
**FORMS**=(*form-id,init,copies*)  
 or  
**FORMS**=NONE

**Parameter options** *form-id*  
 Specifies a 1 to 6 character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling an FDL source file (FSL file).

	<i>init</i>	Specifies the beginning copy number to which a specified form applies. This defaults to the first or next copy. If the <i>copies</i> 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 you want to print.
	NONE	No form is added to the associated report page of variable data.
<b>Default</b>		There is no default.
<b>Considerations</b>		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, the PAPERSIZE parameter of the OUTPUT command cannot be used. It is important to change the image size when paper size will be smaller than the current image so that all data will be printed on the page.
<b>Example</b>		FORMS=(FORM3,3,2);

---

## GRAPHICS

		Specifies that the DJDE is a graphic sentinel. A sentinel 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 text data has been previously encountered, the graphic is document interleaved and copied to an IMG-type disk file.
		If text data has been previously encountered, the graphic is page-interleaved (block mode) and the 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 that 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.
		GRAPHICS is a record-oriented DJDE.
<b>Syntax</b>		<b>GRAPHICS</b> = <i>name</i> or <b>GRAPHICS</b> =( <i>name,vpos,hpos,H,n/d</i> )
<b>Parameter options</b>	<i>name</i>	Identifies the graphic. If the document is interleaved, it becomes the filename with which the IMG file is written. If the page is interleaved, it is used to associate the graphic with other DJDE references.

*vpos*

Specifies the vertical position of the top edge of the graphic, relative to 0,0 on the current physical page. *vpos* is specified in user-defined units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS.

*hpos*

Specifies the horizontal position of the left edge of the graphic relative to 0,0 on the current physical page. *hpos* is specified in user-defined units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS.

H

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.

*n/d*

Specifies the reference scale factor.

**Default**

There is no default.

**Considerations**

Only *name* may be used if the graphic is document interleaved or 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.

An XDOT is a unit of measurement, which represents 1/600th of an inch. Jobs specifying XDOTS can be created, edited, and compiled on any LPS with version 3 software, but the form will not print on your Xerox 4050 or 4090 LPS, which is 300 spi.

## IMAGE

---

Defines the new imaging parameters for a graphic in batch mode. IMAGE is a record-oriented DJDE relating to graphics.

**Syntax**

**IMAGE=(*img-name*,*vpos*,*hpos*)**

**Parameter options**

*img-name*

Specifies 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 *img-name* parameter may be omitted in batch mode; however, the comma before *vpos* must be coded, since it is a positional parameter.

*vpos*

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. *vpos* is specified in user-defined units (UN), centimeters (CM), inches (IN), or DOTS.

*hpos*

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 *vpos*.

<b>Default</b>	There is no default.
<b>Considerations</b>	If user-defined units (UN) are specified for <i>hpos</i> and <i>vpos</i> , the user-defined positioning unit must previously be defined by the UNITS parameter of the OUTPUT command of the current JDE or JDL. If no units are specified, inches are assumed.

**ITEXT**

	Displays a text message to operators during the input process. It takes effect on the next page boundary. ITEXT is a page-oriented DJDE.
<b>Syntax</b>	ITEXT= <i>sc</i> or ITEXT=( <i>sc,passnum</i> ) or ITEXT=NONE
<b>Parameter options</b>	<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 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.  NONE Specifies that no message is to be displayed.
<b>Default</b>	The default is NONE.

**JDE**

	Specifies the JDE to be used within the selected JDL at the next page boundary. JDE is a page-oriented DJDE.
<b>Syntax</b>	JDE= <i>jde-id</i>
<b>Parameter options</b>	<i>jde-id</i> Specifies the JDE to be used within the selected JDL at the next page boundary.
<b>Default</b>	There is no default.
<b>Considerations</b>	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.

## JDL

---

	Specifies the name of the JDL to be invoked at the next page boundary. JDL is a page-oriented DJDE.
<b>Syntax</b>	<b>JDL=<i>jdl-id</i></b>
<b>Parameter options</b>	<i>jdl-id</i> Specifies the name of the JDL to be invoked at the next page boundary.
<b>Default</b>	The default is DFAULT.
<b>Considerations</b>	The <i>jdl-id</i> 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.

## MARGIN

---

	Specifies the left printing margin within each logical page. It takes effect at the logical page following a DJDE END parameter keyword. MARGIN is a page-oriented DJDE.
<b>Syntax</b>	<b>MARGIN=<i>value</i></b> or <b>MARGIN=(<i>value,units</i>)</b>
<b>Parameter options</b>	<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>units</i> Indicates whether a <i>value</i> is specified in inches (IN), centimeters (CM), or character positions (POS).
<b>Default</b>	The default for <i>value</i> is 1.
<b>Considerations</b>	A <i>value</i> must be specified as an integer ( <i>nnn</i> ), if <i>units</i> is specified as character positions.

## MODIFY

---

	Specifies the Copy Modification Entry (CME) to be used on the page. MODIFY is a page-oriented DJDE.
<b>Syntax</b>	<b>MODIFY=<i>cme-id</i></b> or <b>MODIFY=(<i>cme-id,init,copies</i>)</b> or <b>MODIFY=NONE</b>
<b>Parameter options</b>	<i>cme-id</i> Refers to a file which is cataloged separately in the CME directory.  <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 <i>copies</i> 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.
<b>Default</b>		The default is NONE.
<b>Considerations</b>		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. If the CME command is within the JSL, it must precede a reference to its identifier by the OUTPUT command MODIFY parameter.

**NUMBER**

		Specifies page numbering control. It takes effect at the next logical physical page boundary. NUMBER is a page-oriented DJDE.
<b>Syntax</b>		<b>NUMBER=(<i>pnum</i>,<i>lnum</i>,<i>cnum</i>,<i>findex</i>)</b> or <b>NUMBER=NO</b>
<b>Parameter options</b>	<i>pnum</i>	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 is positive. The maximum page number that will be printed is 2,147,483,647.
	<i>lnum</i>	Specifies an integer line number on which the page number is to be placed.
	<i>cnum</i>	Specifies an integer ending column number for the page number sequence.
	<i>findex</i>	A font index in 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.
<b>Considerations</b>		None.
<b>Default</b>		The default is NO.
<b>Example</b>		NUMBER=(1, 1, 40, 1)

## OTEXT

---

		Displays a text message to the operator while the job is printing. OTEXT is a page-oriented DJDE.
<b>Syntax</b>		<p><b>OTEXT=</b><i>sc</i></p> <p>or</p> <p><b>OTEXT=</b>(<i>sc,passnum,WAIT</i>)</p> <p>or</p> <p><b>OTEXT=</b>(<i>sc,END,WAIT</i>)</p> <p>or</p> <p><b>OTEXT=</b>NONE</p>
<b>Parameter options</b>		<p><i>sc</i></p> <p>Specifies a text message of up to 80 characters (maximum of 400 characters per report).</p> <p><i>passnum</i></p> <p>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 <i>passnum</i> is specified, the text is output once at the beginning of printing the entire report.</p> <p><b>WAIT</b></p> <p>Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.</p> <p><b>END</b></p> <p>Specifies that the text is displayed after the last copy of the report is printed.</p> <p><b>NONE</b></p> <p>Specifies that no text message is displayed while the job is printing.</p>
<b>Default</b>		There is no default.
<b>Considerations</b>		The maximum number of OTEXT characters which can be used in a report with GHO processing specified is 320 characters.

## OVERPRINT

---

		Provides instructions to the system when overprint lines occur. Overprint lines are lines whose carriage control specifies printing with no line spacing since the last printed line. OVERPRINT is a record-oriented DJDE.
<b>Syntax</b>		<b>OVERPRINT=</b> ( <i>over-opt,disp</i> )
<b>Parameter options</b>		<p><i>over-opt</i></p> <p><b>PRINT</b></p> <p>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.</p> <p><b>IGNORE</b></p> <p>Specifies that all overprint lines are ignored.</p>

**MERGE**

Specifies the same instructions as the PRINT parameter option, except when used with FONTINDEX or CME processing. Refer to the "Considerations" section of the FONTINDEX parameter keyword of the LINE command in the "Print format commands" chapter for more information.

**PRINT2**

Specifies that up to two consecutive lines are printed per line: one line and one overprint. Other overprints for the line are ignored.

*disp*

Options are DISP and NODISP. These are nonfunctional parameter options which are reserved for compatibility with other Xerox laser printing systems.

**Default** The default is (PRINT,NODISP).  
**Considerations** The number of overprint lines is always printed on the accounting page.

**PMODE**

Specifies the printing mode for each page. PMODE is a page-oriented DJDE.

**Syntax** **PMODE=orientation**  
**Parameter options** *orientation*  
 LANDSCAPE  
 Prints parallel to the long edge of the paper.

PORTRAIT  
 Prints parallel to the narrow edge of the paper.

**Default** The default is LANDSCAPE.  
**Considerations** If PMODE is changed using a DJDE command, the BEGIN command should be specified again (even if the same horizontal and vertical position has been specified in the previous BEGIN parameter).

**RFORM**

Specifies whether a form is printed on all RTEXT pages. It takes effect on the next RTEXT page. RFORM is a page-oriented DJDE.

**Syntax** **RFORM=form-id**  
 or  
**RFORM=NONE**  
**Parameter options** *form-id*

Specifies the name of a file cataloged in the FRM directory.

NONE  
 Specifies that no form is to be printed.

**Default** The default is NONE.  
**Considerations** The RFORM parameter keyword is not allowed in a ROUTE command which is defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file.

## RTEXT

---

	Prints text on a separate page preceding a report. RTEXT is a page-oriented DJDE.
<b>Syntax</b>	<b>RTEXT=</b> <i>sc</i> or <b>RTEXT=</b> ( <i>sc,passnum,line,column,fontindex</i> ) or <b>RTEXT=</b> ( <i>sc,ALL,line,column,fontindex</i> ) or <b>RTEXT=</b> <i>rtext-id</i> or <b>RTEXT=</b> NONE
<b>Parameter options</b>	<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 <i>fontindex</i> ).  <i>passnum</i> An integer number which specifies the pass (copy number) to which the text applies.  ALL Prints a message on all copies to which the text applies.  <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 <i>passnum</i> .  <i>column</i> Specifies the column number at which the first character of a block of RTEXT messages is to be printed.  <i>fontindex</i> Identifies the index (starting with 1) of the PDE font with which the text is printed.  <i>rtext-id</i> References a cataloged file of RTEXT parameters that were compiled by PDL.  NONE Specifies that no text is to be printed on a separate page.
<b>Default</b>	There is no default.
<b>Considerations</b>	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.

**SAVE**

	Updates image (.IMG) files during the current report and saves them after the report is printed. SAVE is a record-oriented DJDE relating to graphics.
<b>Syntax</b>	<b>SAVE=(file-name,...)</b> or <b>SAVE=ALL</b>
<b>Parameter options</b>	<i>file-name</i> Specifies the name of an IMG file or a file name mask whose form and interpretation are as described for the operator SAMPLE command.  <b>ALL</b> Saves all document interleaved files that are created by this report processing, overriding an explicit (or default) specification in the job's OUTPUT command.
<b>Default</b>	There is no default.
<b>Considerations</b>	This DJDE must occur prior to or at the end of the report.

**SEPARATORS**

	Specifies that the system saves an internal copy of the next page in the report and uses it to generate segment separator sheets. If separators are already in effect when the system encounters this DJDE, it overrides the current specifications for all subsequent separators.
<b>Syntax</b>	<b>SEPARATORS=value</b> or <b>SEPARATORS=(BOTH,D)</b> or <b>SEPARATORS=(LAST,D)</b>
<b>Parameter options</b>	<b>FIRST</b> Specifies that the separator sheet be the first sheet of the second and subsequent segments (no sheet is generated for the first segment).  <b>LAST</b> Specifies that the separator sheet be the last sheet of all segments. In an online environment, a true banner trailer page is used as the separator sheet, in place of the saved separator page, if one has been specified.  <b>BOTH</b> Specifies that the system prints a separator sheet both as the first and the last page of each segment.  <b>NONE</b> Specifies that the system not print a separator sheet. However, the system continues to increment the segment numbering. If separator sheets are called for again later in the job, the updated number is used.  <b>D</b> Specifies that the system prints the separator sheet on the back of any LAST sheets. If the report is a simplex report, the system still processes the last page of the segment as a

**Considerations** duplex page. The parentheses are required when using the D parameter option.  
 If segment management is disabled using the SPLIT=OFF DJDE, SEPARATORS defaults to NONE.

## SHIFT

---

Shifts images on the page for binding purposes. SHIFT is a page-oriented DJDE.

**Syntax** **SHIFT**=(*v*<sub>1</sub>, *v*<sub>2</sub>)  
 or  
**SHIFT**=*value*

**Parameter options**

*v*<sub>1</sub>  
 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*<sub>1</sub> is -75 to +75.

*v*<sub>2</sub>  
 A value in dots for the amount of shift on the even (back) side of a duplex page. Range for *v*<sub>2</sub> is -75 to +75.

*value*

NO  
 Specifies that no shift occurs (*v*<sub>1</sub>=0 and *v*<sub>2</sub>=0).

YES  
 Specifies that a shift occurs. A standard size shift occurs if YES is specified (*v*<sub>1</sub> =75 and *v*<sub>2</sub> =-75).

**Default** The default is NO.

**Considerations** An LPS configured with 600 spi automatically doubles dot values to xdots for positioning.

**SIDE**

Repositions the new logical page to the first logical page of the given side of a physical sheet of paper. It takes effect at a logical page boundary. The DJDE SIDE 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 it. The DJDE SIDE overrides both the RPAGE command SIDE parameter and the ROFFSET command.

SIDE is a page-oriented DJDE.

**Syntax**

**SIDE**=(*side-opt*, *offset-opt*)

**Parameter options***side-opt*

In duplex (with no BFORM), *side-opt* produces the following results:

## NUFRONT

Positions a logical page as the first logical page on a new sheet.

## BACK

Positions a logical page as the first logical page of the next available back.

## NUBACK

Positions a logical page as the first logical page on the back of a new sheet.

## NEXT

Positions a logical page as the first logical page on the next available side, the back of the current sheet or the front of the next sheet. The logical page is moved unless it is already properly positioned as the first logical page on the specified side.

*offset-opt*

## OFFSET

Specifies the sheet on which the repositioned logical page occurs is offset in the output stacker.

## NOFFSET

Specifies that no offset occurs.

**Default**

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.

**Considerations**

It is important to ensure that the data does not pass the margin and go off the page.

## SNUMBER

---

Specifies that the system prints the segment number on the separator page.

SNUMBER can be modified as a DJDE under the following conditions:

1. Both SEPARATORS and SNUMBER are previously defined in either a JDE or as DJDEs.
2. If a SEPARATORS parameter is not specified in the EXPORT command, it must be specified as a DJDE, along with the SNUMBER DJDE.

The printed segment sequence number for all separators except a LAST separator in the last sequence is eight characters long. It has the following format:

PART *nnn*

The segment number string on a LAST separator in the last segment is fifteen characters long having the following format:

PART *nnn of nnn*

**Syntax**

**EXPORT SNUMBER**=(*Inum*, *cnum*, *findex*)

**Parameter options**

*Inum*

The line number on which the segment number string is to begin (integers only).

*cnum*

The column number on which the segment number string is to end (integers only).

*findex*

The font index of the font to use in printing the segment number string. If the font index is not specified, the system uses the first font in the font list on the separator page.

**Considerations**

The system interprets the line and column numbers entered in the command using the metrics for the font index specified, or if none is specified, the font used as a default.

If the line or column number you specify causes the segment number to print off the page, the segment number does not print. If the system encounters an invalid line or column specification in the PDL command and a valid specification in a DJDE, unpredictable results may occur.



**SPLIT**

Specifies that you want to use segment management and specifies the conditions for ending segments.

You must use the EXPORT SPLIT parameter in order for other EXPORT parameters to have an effect.

**Syntax**  
**Parameter options**

**EXPORT SPLIT=***value*

*value*

**NOW**

Specifies that the system mark the current or next available page (a back side of a page if printing duplex) as the end of the segment. In order for SPLIT=NOW to take effect, you must include the PDL SPLIT=(*min,max*) statement in the JSL.

*min,max*

The minimum and maximum number of pages allowed in a segment. These values must be specified together. The numbers must be integers between 1 and 32,767. The system includes separator sheets in its count.

If segment management is already in effect (for example, through a PDL statement or previous DJDE), the system tests the current segment against the new max value and if greater, it immediately declares an end-of-segment.

If the system encounters the end of the segment before the *min* value specified in this command, the system generates either blank pages or multiple LAST sheets (if you have specified LAST separators) to pad the segment.

**OFF**

Terminates segment management if it is in effect. If segment management is not in effect, the system ignores the command.

This parameter has the following effects:

- The system immediately declares an end-of-segment
- SRECOVER defaults to PAGE recovery
- SEPARATORS defaults to NONE
- Segment numbering resets to PART 001 (which takes effect when segment numbering is next enabled).
- You must use a DJDE SPLIT=(*min,max*) to reenble segment management processing after it has been disabled by the DJDE SPLIT=OFF command.

## SRECOVER

---

	Specifies the mode of segment recovery. This determines how the system handles printer faults and system errors.
<b>Syntax</b>	<b>EXPORT SRECOVER=(<i>value</i>)</b>
<b>Parameter options</b>	<p><i>value</i></p> <p><u>PAGE</u> Specifies page recovery, which is the system standard method. When the system encounters a fault, it begins reprinting with the first sheet not successfully delivered.</p> <p>SEGMENT Specifies segment recovery. When the system encounters a fault, it reprints the entire segment currently printing. Because it is possible for the printer to process more than one segment concurrently in the paper path, the system may need to reprint more than one segment to effect full recovery of the incomplete segment.</p> <p>ASK Specifies that the system ask the operator to select page or segment recovery through a message on the user interface.</p> <p>DEVICE Specifies that the system follow the recovery method outlined in the DFA specification. Used only when the Bypass Transport is the output destination, otherwise, the default page recovery is used.</p>
<b>Considerations</b>	When used as a DJDE, you must specify the SRECOVER statement prior to the first data record in the data stream of a report, otherwise, it is ignored.

---

**STOCKS**


---

	Redefines the parameters of a STOCKSET command (the types of paper to use) for the page. STOCKS is a page-oriented DJDE.
<b>Syntax</b>	<b>STOCKS</b> = <i>stockset-name</i>
<b>Parameter options</b>	<i>stockset-name</i> This option defines the STOCKSET and the associated stock to be used in a report. If the <i>stockset-name</i> 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.
<b>Default</b>	There is no default.
<b>Considerations</b>	Whenever a new stockset is chosen, that is, at the start of a 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.

---

**TOF**


---

	Specifies the number of lines from the top of the output page to the first print line on the page (top-of-form). TOF is a record-oriented DJDE.
<b>Syntax</b>	<b>TOF</b> = <i>value</i>
<b>Parameter options</b>	<i>value</i> Specifies the number of lines from the top of the output page to the first print line on the page (bottom of form).
<b>Default</b>	The default is 1.
<b>Considerations</b>	If the DJDE TOF value is specified as greater than or equal to the DJDE BOF value, the TOF value is set to 1.  The DJDE TOF 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 a value of 1 is processed and the page transition has already occurred, the DJDE TOF value is too late and does not take effect until the next page transition.  As with all record-oriented DJDEs, TOF takes effect immediately at the next record following the last record of the set of DJDE records of which they are a part, i.e., after a DJDE END. Keep in mind that a compiled record-oriented DJDE, however, takes effect immediately, and this would allow the system to apply TOF to the desired page.



This chapter describes the various formats for graphics processing.

---

## Processing modes

---

Graphics are handled in one of the following modes:

- Batch
- Block
- Move
- Random.

You select the mode depending on how the graphics are made available for report processing, your performance requirements, and disk resources.

---

### Batch mode

---

In batch mode, the input stream contains graphics that can 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 way 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. You can invoke this feature by using the GRAPHICS parameter of the OUTPUT command.

---

### Block mode

---

In block mode, a graphic is obtained from the input data stream immediately following the text data to be imaged on the same page. (For more information, refer to the "Page interleaved" section in this chapter.) The graphic is copied directly to the print file adjacent to the formatted text information of the page. 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 page of a report that includes block mode graphics may also include random mode or move mode graphics.

## Move mode

---

In move mode, as in random mode, a graphic is obtained from a named disk file. In move mode, as in block mode, the graphic is copied into the print file and used. You invoke this feature using the GRAPHICS parameter of the OUTPUT command.

Under some circumstances, move mode graphics are overridden and treated as random mode. In particular, random mode is invoked if, for any page, there is exactly one graphic and that graphic is on a disk drive other than the print file disk drive.

## 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 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.

---

## Tape formats

---

You can use four types of tape formats for graphics on your laser printing system:

- Noninterleaved
- Document-interleaved
- Page-interleaved
- Batch mode.

## Noninterleaved

---

To transfer .IMG files from tape to an LPS system disk independent of printing, you can use the COPY TAPE LABEL *file-name* command.

## Document-interleaved

---

In this format, the data stream of a report 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 you have explicitly specified otherwise.

The rules for formatting a document-interleaved graphic report tape file are as follows.

---

### **Tape format**

Tape format may be any legal fixed or variable length block or record format that is otherwise supported by the LPS:

RECORD STRUCTURE=F or FB or V or VB

It may not be an undefined block or record format:

RECORD STRUCTURE=U or UB

Packed data formats are not supported.

---

### **Block and record constants**

The system supports the use of block and record constants in an appropriately structured file (type F, FB, V, or VB) containing interleaved graphics. 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 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.

---

### **GRAPHIC DJDEs**

Each graphic must be preceded immediately by a GRAPHIC DJDE.

---

### **Record setup**

You get improved tape-to-disk processing time if the graphic data portion of each record is set up as follows:

- Integer multiple of 512 bytes
- 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, they are copied one at a time directly to the print file adjacent to the formatted textual data of the page. If a page-interleaved graphic is not referenced, it is skipped.

The rules for page-interleaved graphics are as follows.

### **Tape format**

---

Graphics must occur on the tape immediately following the last text to appear on the same physical page. Tape format may be any legal fixed or variable length block or record format that is otherwise supported by the LPS:

RECORD STRUCTURE=F or FB or V or VB

It may not be an undefined block or record format:

RECORD STRUCTURE=U or UB

Packed data formats are not supported.

### **Block and record constants**

---

The system supports the use of block and record constants in an appropriately structured file (type F, FB, V, or VB) containing interleaved graphics. 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 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.

### **GRAPHIC DJDEs**

---

Each graphic must be preceded immediately by a GRAPHIC DJDE.

### **Record setup**

---

You get improved tape-to-disk processing time if the graphic data portion of each record is set up as follows:

- Integer multiple of 512 bytes
- 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.



---

## Batch mode

---

In this mode, a report contains only graphics image done per page and an optional IMAGE DJDE preceding each graphic. These graphics are copied to the print file and processed as block mode graphics.

Note that block or record delimiters are not permitted.

Except for the graphics themselves, the input stream may contain only optional DJDE IMAGE records. When a DJDE IMAGE record is encountered, it defines the scaling and positioning parameters for subsequent graphics.

The rules for batch mode graphics are in the following sections.

---

### Tape format

---

Tape format may be any legal fixed or variable length block or record format that is otherwise supported by the LPS:

RECORD STRUCTURE=F or FB or V or VB

It may not be an undefined block or record format:

RECORD STRUCTURE=U or UB

Packed data formats are not supported.

---

### Record setup

---

You get improved tape-to-disk processing time if the graphic data portion of each record is set up as follows:

- Integer multiple of 512 bytes
- 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.

---

## Online formats

---

You can use one of the following types of online formats on your laser printing system:

- Noninterleaved
- Document and page-interleaved
- Batch mode.

---

### Noninterleaved

---

HOSTCOPY is the mechanism for transferring noninterleaved graphics using 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 described in the following sections.

### DJDE GRAPHIC records

---

Each graphic must be preceded immediately by a DJDE GRAPHIC record.

### Record length

---

There is no change in the maximum record length (214 for optimize mode; 2140 for non-optimize mode) when processing interleaved graphic records. However, every record of an online interleaved graphic must have an appended terminating character. This character must be the same for all records of the graphic. If the system encounters an interleaved graphic record that does not have the same terminating character as the first record of the graphic, it will allow you to abort or continue the report. If you continue the report, the graphic is unprintable and the results may be unpredictable.

### Banner pages

---

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 rules for online batch mode graphics are indicated below.

### Batch stream delimiters

---

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.

### DJDE IMAGE records

---

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.

### Maximum record length

---

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. This character must be the same for all records of the graphic. If a batch mode graphic record is encountered that does not have the same terminating character as the first record of the graphic, the system allows you to abort or continue the report. If you continue the report, 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 and using a JDE with an OUTPUT command specifying graphics with PURGE=NO. This function is also available for offline use.

### Management of image files

---

To produce print files consisting of interleaved text and image data, the LPS *file-id* must be saved in ASCII in the LPS 128-byte file label.

---

### PDL command options

---

Several PDL commands are available for graphic handling. These include four JDE OUTPUT commands and six DJDEs.

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.

#### OUTPUT command parameters for graphics

---

##### GRAPHICS

Specifies that graphics may be used in current job.

##### IMAGE

Specifies initial scaling and positioning parameters for batch mode.

##### PURGE

Specifies the default disposition of document-interleaved graphics.

##### UNITS

Specifies user-selected units for positioning graphics.

### DJDE definitions for graphics

---

**ALTER**

Specifies new imaging parameters for graphics previously referenced using a HOLD statement.

**BATCH**

Marks the beginning and end of online batch mode graphic data.

**CANCEL**

Terminates a HOLD statement.

**GRAPHICS**

Precedes an interleaved graphic in the data stream.

**IMAGE**

Specifies the name, scaling, and positioning of a printed graphic.

**SAVE**

Identifies the document-interleaved graphics not to be deleted (purged).

---

## Performance considerations

---

Several factors may impact the processing and printing of graphics, including random mode access, online printing, and document-interleaved filing.

- 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 less chance there is of completing the disk reads in one page setup time.
- 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.
- The time required to create (or replace) and subsequently purge document-interleaved graphics is significant.

---

## Graphic features restrictions

---

This section summarizes graphic restrictions. These restrictions are grouped according to whether they affect graphic features. The restrictions are applicable without RIP and Interpress.

- Tape formats supported by interleaved graphic processing are restricted to LPS-compatible fixed or variable records structures:

RECORD STRUCTURE=F or FB or V or VB

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, however, a restriction on the total number of graphic bits (overlapped or not) that may be specified on a given scan line.
- The maximum amount of graphic data that can be imaged on a simplex page is limited by the size of graphics memory. Throughput reduces if more than one megabit of graphic data is imaged on a page.
- There is no allowance or "tolerance" available for the paper size in a JDE for a graphic to be off the page.
- Online banner pages (header or trailer) may not include graphic references.
- A JDE that is selected using 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 DJDE BATCH=START until the DJDE BATCH=END.
- 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 command 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 that the RPAGE command 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 that 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.



---

## 7. Paper stocks and clusters

The cluster printing feature allows you to designate multiple feed trays as a single logical group, known as a cluster. Each cluster handles one type of paper stock. The LPS switches automatically between the paper trays within the cluster to allow continuous printing. You can group paper trays with different sizes of paper together to allow the LPS to determine what size paper to use for a print job. The system also provides error checking to ensure that jobs are being printed on the correct paper stock.

---

### Programmer and operator tasks

---

As the programmer, you group any combination of trays into a cluster. The trays are grouped together because they hold the same stock or they hold stock that differs only in size. Specify the paper stock by any combination of weight, color, size, and so on.

Use clusters to include instructions within the report that identify the following:

- Clusters needed by the report (that is, the "stockset")
- Particular cluster to use to print each page
- Recommended trays for loading the stock.

An operator uses clusters to decrease how often stocks are loaded, and verifies job and cluster status, using the CLU commands. The cluster definitions specify which stock to load in which tray for queued jobs.

When cluster trays are loaded without entering the CLU commands, the system assumes that the trays are full and that no cluster changes occurred.

---

### The CLUSTER.LIB file

---

You store the defined clusters in the CLUSTR.LIB file using the cluster (CLP or CLU) commands. The CLUSTR.LIB file is the cluster database for the LPS. Each cluster definition is a record in the file. The system creates the file when a print job is run and no CLUSTR.LIB exists.

CLUSTER.LIB 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.

You cannot edit the CLUSTER.LIB. It is automatically updated by the CLU and CLP commands that enter clusters in the database or modify cluster definitions.

CLUSTER.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 that are not in the second system are deleted from the cluster definitions which reference them.

## Paper stock commands and DJDEs

You use the PDL commands and parameters listed in table 7-1 to specify paper stocks. Refer to the "Print format commands" chapter for more information on these PDL commands.

Table 7-1. PDL commands specifying paper stock

Command	Parameter	Options
OUTPUT	FEED =	AUX MAIN OPR <i>stock-name</i> <i>stock-reference</i>
	STOCKS =	<i>stockset-name</i>
STOCKSET	ASSIGN =	<i>stock-descriptor</i> ( <i>stock-descriptor</i> <sub>1</sub> , <i>stock-descriptor</i> <sub>2</sub> ,...)
	INIFEED =	<i>stock-name</i> <i>stock-reference</i>
	SYSPAGE =	<i>stock-name</i> <i>stock-reference</i>

You use the DJDEs listed in table 7-2 to modify paper stock specifications. Refer to the "Print control (DJDE) commands" chapter for more information on these DJDEs.

Table 7-2. DJDEs modifying paper stock

DJDE	Options
FEED =	AUX MAIN OPR <i>stock-name</i> <i>stock-reference</i>
STOCKS =	<i>stockset-name</i>



The default CLUSTER.LIB contains definitions for the MAIN, AUX, and AUTO clusters. You can store these clusters as they are, or modify them before you use them. The system defines the trays as follows:

- MAIN represents tray 1
- AUX represents tray 2
- AUTO represents trays 1, 2, 3, 4.

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

Do not use FEED=MAIN and FEED=AUX in conjunction with stocksets.

---

## Cluster names and references

---

When creating clusters, 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 parameter 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 may cycle down when a report calls a stockset with more clusters than the number of feeder trays on the system.

---

## Defining stocksets

---

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.

---

## RAUX options

---

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.

## Creating clusters

Follow these steps to create clusters:

1. Determine the stocks that your application uses.
2. Create a .JSL file and compile it to produce an .STK file.
3. Define a logical mnemonic name, a cluster reference, for each stock. Refer to the example in step 4 where it uses COVER.
4. Associate each cluster reference with a cluster name that will be easily recognized by the operator as a particular stock. Refer to the following example where CV102 is used:

```
BILLS: STOCKSET      ASSIGN=('CV102,' COVER),
                    ASSIGN=('BL231,' BILPAG),
                    ASSIGN=('SM233,' SUMMARY);
```

5. Use the cluster reference rather than the cluster name in the .JSL files and DJDEs. This way you can make cluster stock changes by modifying the cluster name in the STOCKSET command and recompiling it. You also avoid modifying the data stream or the application. The cluster reference in JDL and PDE files remains valid.
6. Use the FEED=DJDE option in the JSL file to specify the stock you want to use, 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;
```

You must identify the active cluster in the specified cluster.

The OUTPUT statement requires the STOCKS= option; the FEED= option is optional. The INFEEED= option of the stockset takes effect when the no FEED= option is coded. If the stockset has no INFEEED= option, the system uses the first cluster you specify in the stockset. The following example produces the same results as the previous example:

```
BILLS:STOCKSET      ASSIGN=('CV102,' COVER),
                    ASSIGN=('BL231,' BILPAG),
                    ASSIGN=('SM233,' SUMMARY),
                    INFEEED=COVER;

OUTPUT              STOCKS=BILLS;
```

The STOCKS=DJDE makes the stockset INIFEED= option take effect, thereby canceling the previous FEED= option.

The STOCKSET statement has an optional SYSPAGE= option. It feeds the system pages, such as the PLABEL, OPRINFO, rollover marker pages, and accounting pages, from the specified cluster. For example:

```
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 INIFEED FEED= in JSL, or the FEED= DJDE. A stockset change causes the LPS output task to make additional checks when a page is ready to print. These checks can result in a printer cycledown if any cluster in the new stockset presents a tray overlap or sizing problem. Thus, you should keep stockset changes within applications to a minimum. For example:

```
DJDE STOCKS=BILLS1, END;
...data...
DJDE STOCKS=BILLS1, END;
...data...
DJDE STOCKS=BILLS2, END;
...data...
```

This data stream has three STOCKSET= callouts, but only one stockset change. Report boundaries are not treated as stockset changes unless use of the stockset specified by Job Descriptor Entry (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 prints using the JDE/JDL of START. If it does not specify OUTPUT STOCKS=BILLS1, the stockset changes.

If the stockset changes, but the second stockset consists of the same clusters as the first stockset, no cycledown occurs since, for printing purposes, the stockset has not changed.

### Using clusters with ordered or tab stocks

Ordered stocks consist of any repeating sequence of paper stock types stacked in a tray. These clusters function in the same way that other clusters function. A special case of ordered stock is a tab cluster: the tab positions of the tab stock are stacked in a repeating sequence.

---

## Using clusters in print jobs

---

You can specify clusters on your applications in the following ways:

- Simple applications, which do not change stock types
- OTEXT applications, which change stock types but do not use stockset commands to do so
- STOCKSET applications, which use the STOCKSET command to change stock types.

### Simple and OTEXT applications

---

These applications do not provide for the use of different stocks. If the application is online, 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.

If your site runs simple, OTEXT, and stockset jobs, then it is useful to maintain a cluster for each stock loaded using cluster or stockset definitions and changing MAIN and AUX to logicals for each cluster as required by the jobs.

### STOCKSET applications

---

Each page has a particular stockset and a particular cluster associated with it. The cluster is always verified because it determines from which trays a page may feed. The stockset is verified only if the current page calls a different stockset than the previous called page.

Stockset changes require the cluster checks listed in table 7-3.

Table 7-3. Cluster checks for stockset changes

Condition	Verifying question
Missing	Is the cluster defined in the CLUSTER.LIB?
No trays	Does the cluster have current trays?
Tray overlap	Do any of the clusters have the same current trays?
Sizing error	Are all trays the same size? If not, is autosize set?

A cluster without these errors has a status of "in use." 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 a status of "in error" to the stockset.

As jobs print, the output task uses the cluster application specified 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, FEED=DJDE, then the printer cycles down and displays the following:

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

If there is an error in any clusters in the stockset, the system gives a status of "in error" to the stockset. Tray Overlap and Sizing Error are fatal errors. The printer does not cycle down if it is active, and does not cycle up if it is inactive. The printer cycles up only if you remedy the problem. Tray Overlap and Sizing Error are the only errors that the printer detects.

Missing and No Trays are not fatal errors. The printer cycles down only if a report in the job calls a report in the job.

The system displays the following errors only the first time they are encountered in the stockset:

```
Sizing error when paper is too large for data (requires OUTPUT
SIZING = SEMIAUTO) Data off page error Operator FEED command
overrides.
```

As jobs print, the output task uses the cluster that the print job specifies to choose a tray for each page. Occasionally, it provides a cluster status display. If you enter FEED commands that override the data, for example, FEED=DJDE, then the printer cycles down and displays the following message:

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

The system issues the confirmation once per stockset. To respond, enter **FEED**, which allows the LPS to select the feeder tray, or enter **CONTINUE O**. If stockset or cluster problems are found, the output task informs you with the appropriate messages and blinking status values for the problem clusters. Once you correct the error or decide to override the status, enter **CONTINUE O**.

## Mixing applications

---

When a site runs simple and OTEXT applications with stockset applications, the LPS executes the following two additional stockset checks to help ensure the correct paper is loaded:

- If a report has no stockset information, a stockset change is assumed.
- 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, make sure 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.

## A. PDL command summary

This appendix lists all PDL commands, as shown in table A-1. It also specifies the laser printing system or software versions that support them.

The term "ALL" indicates that the command or parameter can be used by all Xerox laser printing systems (8700/9700, 8790/9790, 4050, 4090, 4135, 4450, 4650, 4850, and 4890) operating with version 3 software or higher.

You can achieve compatibility between the Xerox 4050/4090/4450/4650 LPS and the Xerox 4850 Highlight Color LPS. Refer to the "4850 Highlight Color and 4135 LPS compatibility" appendix for further information.

The symbol *ac* is used to represent command identifiers. The *ac* identifier consists of one to six alphanumeric characters (A through Z and 0 through 9). At least one of the characters must be a letter.

Command or parameter availability for offline or online use is indicated by either a "Y" for yes or an "N" for no.

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-1. PDL commands

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	N	Y	—
	TCOUNT	0	All	N	Y	—
	TEST	—	All	N	Y	—
	TYPE	BANNER	All	N	Y	—
BDELETE	TEST	—	All	Y	N	—

Table A-1. PDL commands (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
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	—
	INKS	—	4850, 4890	Y	Y	—
	LINE POSITION	—	All	Y	Y	—
	POSITION	1	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	—
	VALUE		4090	Y	Y	—
END	—	—	All	Y	Y	—
EXPORT	SEPARATORS	NONE	4135, 4850, 4890, 4090	Y	Y	Page
	SNUMBER	—	4135, 4850, 4890, 4090	Y	Y	Page
	SPLIT	—	4135, 4850, 4890, 4090	Y	Y	Page
	SRECOVER	PAGE	4135, 4850, 4890, 4090	Y	Y	Page
	STIMING	—	4135, 4850, 4890, 4090	Y	Y	Page
FILE	—	—	All	N	Y	Record
IDEN	OFFSET	0	All	Y	Y	—
	OPRINFO	NO	All	Y	Y	—
	PREFIX	—	All	Y	Y	—
	SKIP	1	All	Y	Y	—
<i>ac</i> :IDR	ICATALOG	DFAULT	4850, 4890	Y	Y	Page
	ILIST	—	4850, 4890	Y	Y	Page
	PALETTES	DFAULT	4850, 4890	Y	Y	Page
<i>dd</i> :JDE	INCLUDE	—	All	Y	Y	Page
<i>dd</i> :JDL	—	—	All	Y	Y	Page
<i>dd</i> :JOB	INCLUDE	—	All	Y	Y	—



Table A-1. PDL commands (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
LINE	DATA	offline=1,132 online=0,150	All All	Y Y	Y Y	Record Record
	FCB	PROCESS	All	N	Y	—
	FONTINDEX	NONE	All	Y	Y	Page
	INKINDEX	NONE	4850, 4890	Y	Y	Page
	MARGIN	1,POS	All	Y	Y	Page
	OVERPRINT	PRINT,NODISP	All	Y	Y	Record
	PCC	0,NOTRAN	All	Y	N	—
	PCCTYPE	offline=ANSI online=IBM3211	All	Y	Y	—
	UCSB	PROCESS	All	N	Y	—
	VFU	NONE	All	Y	Y	—
LMODIFY	TEST	NONE	4090	Y	Y	—
MESSAGE	ITEXT	NONE	All	Y	Y	Page
	OTEXT	NONE	All	Y	Y	Page
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	Page
	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	—
	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	Record
	IRESULT	sysgened default	4850, 4890	Y	Y	Page
	LOGO	IN	4890	Y	Y	Record
	MODIFY	NONE	All	Y	Y	Page
	NTO1	NO	All	Y	Y	—
	NUMBER	NO	All	Y	Y	Page
	OFFSET	ALL	All	Y	Y	—

Table A-1. PDL commands (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
	OSTK	RES MOD and POS default=1; SIZE=.5 inch	4135	Y	Y	—
	PAPERSIZE	sysgened default	All	Y	Y	—
	PURGE	YES	All	Y	Y	—
	RESOLUTION	sysgened default	All	Y	Y	—
	SHIFT	NO	All	Y	Y	Page
	SF1FUNCTION	NO	4135	Y	Y	Page
	SIZING	—	All	Y	Y	—
	STAPLE	NO	All	Y	Y	—
	STOCKS	—	All	Y	Y	Record
	SYSPPR	—	4135	Y	Y	—
	TMODE	depends on papersize; default value is inches	4135	Y	Y	Page
	TRANS	—	4135	Y	Y	Page
	UNITS	—	All	Y	Y	—
	XMP	DEFAULT	4850, 4890	Y	Y	Page
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	N	—
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	Y	N	—
RFEED	TEST	no default for <i>clu-def</i>	4050, 4090, 4650 (V3.8), 4850 (V4.0), 4890, 4135	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

Table A-1. PDL commands (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
RPAGE	SIDE	NUFRONT, NOFFSET	All	Y	Y	Page
	TEST	—	All	Y	Y	—
	WHEN	TOP	All	Y	Y	—
RRESUME	BEGIN	NEXT	All	Y	N	—
	TEST	—	All	Y	N	—
RSELECT	TEST	—	All	Y	N	—
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	N	—
	TEST	—	All	Y	N	—
SEFFNT	MAP	—	4135	Y	Y	Page
	SEFMAP	—	4135	Y	Y	Page
ac:STOCKSET	ASSIGN	—	All	Y	Y	—
	INIFEED	<i>first stock-name</i>	All	Y	Y	—
	SYSPAGE	—	All	Y	Y	—
ac:SYSTEM		—	All	Y	Y	—
ac:TABLE	CONSTANT	—	All	Y	Y	—
	MASK	—	All	Y	Y	—
ac:TCODE	DEFAULT	—	All	Y	Y	—
	TASSIGN	—	All	Y	Y	—
	TRESET	—	All	Y	Y	—
ac: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, NOCHECKSUM, NOBREAKPAGE, SIMPLE, INTEGRAL, PERFORMANCE, RIP	All	Y	N	—

Table A-1. PDL commands (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
	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	—
	VCODE	EBCDIC, VCODE0	4090 (XDDI), 4850/4890 (V4.1)	Y	N	—

## B. Character code assignments

This appendix shows the various character code assignments.

Table B-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.  
†Corresponds to BCD code set used by IBM users and defined by PDL command CODE = IBMBCD.

Table B-2. **Honeywell 200/2000 BCD code set**

Table B-3. **Honeywell 6000 BCD code set**

Table B-4. **Field data translation**

Table B-5. **UNIVAC ASCII character set**



Table B-6. **Standard ASCII character set**

Table B-7. **Standard EBCDIC character set**

Table B-8. **Xerox EBCDIC to extended ASCII hexadecimal translation values**

---

## C.

## Offline specifications

This appendix shows offline specifications.

Figure C-1. **Input unpacking examples**

Table C-1. **Valid host computer and label specifications**

Host Job Description Library (JDL) source files supplied on Operating System Software (OSS) tape are summarized in table C-2.

Table C-2. **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="0"> <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 Fielddata</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 Fielddata	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 Fielddata																
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																
IBMCA	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-formatted 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 in the JSL directory. Use the Editor PRINT parameter or the PDL compiler to obtain a printout of the JDL you want.

---

## D. Online specifications

This appendix lists the online specifications for the following:

- Hexadecimal and binary codes for FCB
- UCSB byte locations
- Characters associated with the UCSB field
- 4245 emulated command codes.

---

### Hexadecimal and binary codes for FCB

---

The Forms Control Buffer (FCB) defines the channel positions and forms length. It is analogous to a carriage return on an impact printer. Each byte in the buffer corresponds to one line on the form.

Table D-1. Hexadecimal codes for FCB

Location	Function
01 to 0C	Channels 1 to 12
00	No channel assignment
1n	Last line, where n may be zero or a channel code

Indexing (shifting of print position one) may be initiated when the FCB is loaded. This is defined by a one-byte code, the Print Position Indexing (PPI) byte, preceding the forms definition data. The PPI byte should be omitted from the FCB load data if no indexing is required. The maximum shift is 30 print positions.

Table D-2. Binary codes for FCB

Location	Function
100N <i>nnnn</i>	Shift right <i>n</i> - 1 positions
010N <i>nnnn</i>	Shift left <i>n</i> - 1 positions (first <i>n</i> - 1 bytes of each record is not printed by the LPS)

Online input processing modifies the size and address of the print line based on the PPI byte before processing the DATA command of the LINE statement. If the resulting size of the print line is zero or negative (after PPI modification), only PCC action takes place.

---

## Function of UCSB byte locations

---

The correlation between location in the associative field hexadecimal values, characters, and the associative bits that determine the character printability are listed in tables D-2 through D-4.

Table D-3. **Function of UCSB byte locations**

Location	Function
1-432	Train image field; used in responding to a CHECK READ command that is not preceded by a DIAGNOSTIC GATE command.
433-447	Reserved field; ignored by the LPS.
448-511	Associative field; bits 0-3 of each location are used to define printable characters.
512	Reserved field; ignored by the LPS.



---

## Characters associated with UCSB field

---

Table D-4. Characters associated with UCSB field

**Note:** Characters in the “Char” column are shown for illustration only. The actual characters printed are a function of the font specified in the PDE statement.

Table D-4. **Characters associated with UCSB field**  
(continued)

**Note:** Characters in the “Char” column are shown for illustration only. The actual characters printed are a function of the font specified in the PDE statement.

---

**3211/4245 emulated command codes**


---

Table D-5. 3211/4245 emulated command codes

Hex	Function	Hex	Function	Hex	Function
01	Write without spacing	0B	Space 1 line immediate	00	Test I/O
09	Write and space 1 line	13	Space 2 lines immediate	02	Read Print Line Buffer (PLB)
11	Write and space 2 lines	1B	Space 3 lines immediate	03	No-op
19	Write and space 3 lines	83	Skip immediate to channel 0	04	Sense
89	Write and skip to channel 1	8B	Skip immediate to channel 1	05	Diagnostic Write
91	Write and skip to channel 2	93	Skip immediate to channel 2	06	Check Read
99	Write and skip to channel 3	9B	Skip immediate to channel 3	07	Diagnostic Gate
A1	Write and skip to channel 4	A3	Skip immediate to channel 4	0A	Read UCSB
A9	Write and skip to channel 5	AB	Skip immediate to channel 5	12	Read FCB
B1	Write and skip to channel 6	B3	Skip immediate to channel 6	23	Unfold
B9	Write and skip to channel 7	BB	Skip immediate to channel 7	43	Fold
C1	Write and skip to channel 8	C3	Skip immediate to channel 8	63	Load FCB
C9	Write and skip to channel 9	CB	Skip immediate to channel 9	65	Raise cover (functionally ignored)
D1	Write and skip to channel 10	D3	Skip immediate to channel 10	73	Block data check
D9	Write and skip to channel 11	DB	Skip immediate to channel 11	7B	Allow data check
E1	Write and skip to channel 12	E3	Skip immediate to channel 12	FB	Load UCSB
				E4	Sense ID (4245 only)
				87	Clear Printer (4245 only)



---

## E. 4850 Highlight Color and 4135 LPS compatibility

---

### Software compatibility with the 4850 and 4135 LPS

---

Version 3.8 software and XDDI, which may be loaded on your 4050, 4090, or 4650 laser printing system has two primary functions:

- Compatibility with the 4850 HighLight Color LPS
- Limited compatibility with the 4135 LPS.

These functions are illustrated in figure E-1 and explained in the following sections. Applications created on 4050, 4090, and 4650 systems with version 3.5 software are also shown.

---

### Printing color jobs

---

If your 4050, 4090, or 4650 LPS runs software version V3.8 or XDDI, print jobs you create with color commands print in black and shades of gray. Remember the following when printing color jobs on your LPS:

- You must recompile forms that use color commands.
- You must print the job you created with color commands to verify that the colors meet your requirements. Applications build color sets differently, resulting in color variation on the LPS. If the print job does not meet your color specifications, adjust your color command selection, and retry the print job.
- If your color print job calls for a solid color to print on a black background, the LPS prints the entire job as solid black.

Figure E-1. **Software version 3.8 and XDDI compatibility**

---

## Xerox 4850 HighLight Color LPS

---

If you have a 4850 HighLight Color LPS, version 3.8 software on your 4050, 4090, or 4650 LPS provides you with two types of compatibility:

- Running 4850-created applications on your 4050, 4090, or 4650 LPS, which print in monochrome black
- Creating highlight color applications on your 4050, 4090, or 4650 LPS using ink specifications which can then be compiled and run on the 4850 LPS in the specified highlight color.

Version 3.8 software is designed primarily to enable your 4050, 4090, or 4650 laser printing systems to serve as a backup for your 4850 to run color jobs in monochrome black as the need arises.

If you have a 4850 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 4050, 4090, or 4650 laser printing system will compile, process, and print these jobs, ignoring the ink specified and using the default black ink.

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

### Printing 4850 LPS forms

---

The 4050, 4090, and 4650 LPS have monochrome (black only) forms compilers. These systems on version 3.8 software make 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 one of the following: compiling the FSL using the 4850 LPS Forms Description Language (FDL) compiler, converting the monochrome form to color with the File Conversion Utility (FCU) resident on the 4850 LPS, or downloading color forms created from host or third party vendor software packages.

In addition, the 4050, 4090, and 4650 LPS on version 3.8 software makes a distinction between color and monochrome logos. Due to differences in file formats, these systems restrict 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 4050, 4090, and 4650 LPS V3.8 FDL compilers cannot compile an FSL containing color-unique commands. However, this does not imply that these systems will not print them. If an FSL were precompiled in a 4850 LPS FDL compiler and its file (.FRM) downloaded to a 4050, 4090, or 4650 LPS on version 3.8 software, the form will print. XDDI software does not have this limitation.

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

### **Creating 4850 applications on your 4050, 4090, or 4650 LPS**

---

You can also create color JSLs on your 4050, 4090, or 4650 LPS, but you must compile them on a 4850. When recompiled on your 4850, these JSLs can then be run on either your 4050, 4090, or 4650 LPS or on your 4850. Any ink specifications you have made in the JSL are printed in the default black ink on the 4050, 4090, or 4650, but print in the desired inks on your 4850.

The only exception to this is the RFEED command, which is supported on version 3.8 software, but not on your 4850. The RFEED command is discussed in the "4135 LPS" section, later in this appendix.

### **Points to note**

---

Light tints with isolated pixels of color that print on the 4850 may not print on the 4050, 4090, or 4650 LPS.

Color text printed over solid black background or black text printed over solid color backgrounds on 4050, 4090, or 4650 printers 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 4050, 4090, or 4650 printers 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 4650 are finer and more uniform than 300 spi tints and shades printed on the 4050, 4090, or 4850 systems.

## **4135 LPS**

---

Version 3.8 software allows limited compatibility between your 4135 and 4050, 4090, or 4650 printers. Instead of causing a LPS rollover, 4135 applications submitted to 4050, 4090, or 4650 printing systems are either processed and printed, or an error message allows you to either continue or abort the print job.

### **Running 4135 applications on your 4050, 4090, or 4650 LPS**

---

4135 data streams are processed and printed according to the capabilities of the LPS. 4135 features and commands that are not supported are either rejected or ignored so that as much of the job is printed as possible. In addition, the 4050, 4090, or 4650 LPS with V3.8 software informs you or the LPS operator when it encounters an unsupported 4135 PDL command or DJDE and gives you the opportunity to either continue or abort the print job. When an unsupported DJDE is encountered, the LPS automatically prints an OPRINFO page and delivers it to the sample tray on the printer, although the job may print acceptably anyway.



To minimize this situation, you should recompile your 4135 JSLs on a 4050, 4090, or 4650 LPS with version 3.8 software, rather than directly copying the 4135 JDL onto one of these printers. When recompiling, the V3.8 software removes and notes unsupported features before the application begins to print.

4135 fonts, graphics, and logos are compatible with V3.8 software and may be copied onto a 4050, 4090, or 4650 LPS. Forms are also compatible, except for A3 and B4 paper sizes, which cannot be physically supported on the 4050, 4090, or 4650 printer. The entire print job is rejected when A3 or B4 paper sizes are specified. You or the operator have the option to select an acceptable paper size so that the job can print successfully.

The following PDL commands, parameters, and DJDEs are 4135-specific:

#### **PDL commands and parameters**

- OUTPUT command:
  - OSTK parameter
  - PAPERSIZE (A3 and B4 options)
  - SEFFNT parameter
  - SEFMAP parameter
  - SYSPPR parameter
  - TRANS parameter.
- RFEED command
- RSUSPEND command
  - SEFFNT parameter
- SEFFNT command.

#### **DJDEs**

- SEFFNT
- SEFMAP
- TRANS.

---

### **Creating 4135 applications on your 4050, 4090, or 4650 LPS**

As with any application, you can create 4135 JSLs on your 4050, 4090, or 4650 LPS and compile them on any of these three printers or on your 4135 LPS for printing.

While version 3.8 software supports the RFEED command, it does not support the other 4135-specific PDL commands. Therefore, even though you may have created and compiled the JSL on a 4050, 4090, or 4650, the 4135-specific PDL commands and DJDEs, except for RFEED, are not processed.

---

### **Using the RFEED command**

The RFEED logical processing command allows you to change cluster names on a page-by-page basis without using DJDEs. The RFEED command is the only 4135 PDL command which is included as part of the version 3.8 software release. This means that 4050, 4090, and 4650 laser printing systems with version 3.8 software can process RFEED parameters in the data stream as part of a 4135 application, or as a 4050, 4090, or 4650 application alone.

RFEED command syntax and usage is discussed in detail in the "Using logical processing commands" chapter.

**Points to note—RFEED**

---

RFEED is not supported by the Xerox 4850 HighLight Color LPS.

Its use in version 3.8 software applications may cause slower throughput processing.

RFEED is not available as a DJDE.

<b>A3</b>	International paper size measuring 297 by 420 mm or 11.69 by 16.54 inches.
<b>A4</b>	International paper size measuring 210 by 297 mm or 8.27 by 11.69 inches.
<b>B4</b>	International paper size measuring 250 by 353 mm or 9.84 by 13.9 inches.
<b>batch processing</b>	Process that allows for repetitive operations to be performed sequentially on batched data without much involvement from the computer operator.
<b>BCD</b>	Binary coded decimal.
<b>bitmap</b>	Visual representation of graphic images in which a bit defines a picture element (pixel) and a matrix of bits defines an image. For example, if a bit is 1, the corresponding pixel is printed.
<b>blocking</b>	Process of combining two or more records into a single block of data which can moved, operated upon, stored, and so on, as a single unit by the computer.
<b>block length</b>	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.
<b>BOF</b>	Bottom of form.
<b>BOT</b>	Beginning of tape.
<b>bpi</b>	Bits per inch.
<b>CCID</b>	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.
<b>character set</b>	Set of all characters defined in a font, including alphabet, numeric, and special characters such as symbols.
<b>cluster</b>	Group of related feeder trays, usually containing the same size and type of paper (stock). Each cluster has a name, consisting of one to six alphanumeric characters.

<b>CME</b>	Entry modifying the output printing characteristics of a report on a copy-to-copy basis.
<b>compiler</b>	Software that translates instructions written in high-level language into machine language for execution by a system.
<b>Copy Modification Entry</b>	See <i>CME</i> .
<b>copy-sensitive</b>	Job in which multiple copies of a report contain different data, such as paychecks and banking statements.
<b>cpi</b>	Characters per inch.
<b>default</b>	Value assigned to a field by the system if no input is received from the operator. You can change the default value of a field.
<b>DJDE</b>	Dynamic Job Descriptor Entry. Command within an input data stream used to modify the printing environment dynamically.
<b>dot</b>	Picture element (pixel) imaged by a printer. The number of dots imaged per inch measures printer resolution, for example, 300 dots per inch (dpi). See also <i>spot</i> .
<b>dpi</b>	Dots per inch. Indicates the number of dots per inch displayed on a terminal screen or printed to form a character or graphic.
<b>dry ink</b>	Minute particles of resin and carbon black that can accept an electrical charge and create images. Resin and carbon black or color pigment toner are combined with developer to form the dry ink.
<b>duplex</b>	1. Ability of a data communications system to send and receive information simultaneously. 2. In printing, duplex means printing on both sides of the paper.
<b>Dynamic Job Descriptor Entry</b>	See <i>DJDE</i> .
<b>edgemarking</b>	Use of graphic objects (usually lines or boxes) that bleed off the edge of the physical page. See also <i>physical page</i> .
<b>embedded blanks</b>	Blank spaces within a command line.
<b>ENET</b>	Ethernet network.
<b>EOT</b>	End of tape.
<b>Ethernet</b>	Xerox local area network (LAN) that allows transmission of data by cable from one device to another.

<b>FCB</b>	Forms Control Buffer. Controls the vertical format of printed output.
<b>FCP</b>	File Control Parameter.
<b>FCU</b>	File Conversion Utility.
<b>FDL</b>	Forms Description language. LPS-resident source language used to design electronic forms. See also <i>FSL</i> and <i>form</i> .
<b>FDR</b>	File directory.
<b>FIS</b>	Font Interchange Standard. Defines the digital representation of fonts and character metrics for the generation of an entire series of Interpress fonts.
<b>floating accent</b>	Nonspacing accent characters that can be combined with characters and printed as a composite.
<b>font</b>	Complete set of characters of a particular font family having the same point size, weight, stress, and orientation.
<b>Font Interchange Standard</b>	See <i>FIS</i> .
<b>form</b>	1. Compiled .FSL file. 2. Specific arrangement of lines, text, and graphics stored in an electronic version. Forms can be printed without variable data or merged with variable data during the printing process. See also <i>FDL</i> and <i>FSL</i> .
<b>Forms Control Buffer</b>	See <i>FCB</i> .
<b>Forms Description Language</b>	See <i>FDL</i> .
<b>Forms Source Library</b>	See <i>FSL</i> .
<b>FSL</b>	Forms Source Library. Uncompiled collection of user-created files containing FDL commands. See also <i>FDL</i> and <i>form</i> .
<b>hexadecimal</b>	Numbering system with a base of 16. The numbers 10 through 15 are represented by A through F.
<b>highlight color</b>	Printing with black plus another color. A range of colors, tints, and shades is printed by varying the percentage of black dots, colored dots, and the white space between the dots.
<b>HIP</b>	Host Interface Processor.
<b>image area</b>	Area on a physical page that may contain text or graphics.

<b>initialize</b>	1. To prepare a blank diskette so it can accept data. This is usually accomplished when a program is booted. 2. To set all information in a computer system to its starting values.
<b>Interpress</b>	Industry-standard page description language developed by Xerox. Interpress documents can be printed on any sufficiently powerful printer equipped with Interpress print software.
<b>JCB</b>	Job Control Block.
<b>JCL</b>	Job Control Language.
<b>JDE</b>	Job Descriptor Entry. Collection of job descriptions.
<b>JDL</b>	Job Description Library. Collection of compiled job descriptions. See also <i>JSL</i> .
<b>JID</b>	Job Identifier.
<b>job</b>	Synonymous with a START command, a job is a group of print data sets called reports. A job may contain one or multiple reports.
<b>job control</b>	Program called into storage to prepare each job or job step to be run.
<b>Job Descriptor Entry</b>	See <i>JDE</i> .
<b>Job Descriptor Library</b>	See <i>JDL</i> .
<b>job management</b>	Collective functions of job scheduling and command processing.
<b>Job Source Library</b>	See <i>JSL</i> .
<b>JSL</b>	Job Source Library. Collection of uncompiled job descriptions. See also <i>JDE</i> and <i>JDL</i> .
<b>keyword</b>	Required part of a command.
<b>label</b>	Reference to a file saved on tape or disk, a record indicating the file name or date created, or other control information.
<b>landscape</b>	Orientation in which text and images are positioned parallel to the long edge of the paper.
<b>legal size</b>	Paper size measuring 8.5 by 14 inches or 216 by 356 mm.
<b>letter size</b>	Paper size measuring 8.5 by 11 inches or 216 by 279 mm.

---

<b>line feed</b>	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.
<b>literal</b>	Alphanumeric character beginning with a letter, including an asterisk, period, colon, or slash, and not enclosed in single quotes.
<b>logical page</b>	In Xerox printing systems, a logical page is 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.
<b>lpi</b>	Lines per inch.
<b>LPS</b>	Laser printing system.
<b>mask</b>	Selection of bits from a storage unit by use of an instruction that eliminates the other bits in the unit. In accessing files, a file name mask is used to reference one or more files with similar file-id (identifier) syntax. In Interpress, a mask serves as a template, indicating the shape and position of an object on a page.
<b>metacode</b>	Method of controlling the image generator. The character dispatcher uses these codes 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. Also called native mode.
<b>monochrome</b>	Printing in one color only.
<b>nesting</b>	Subroutine or set of data, such as a comment, contained sequentially within another set of data.
<b>object file</b>	Source file converted into machine language (binary code).
<b>octal</b>	System of representing numbers based on 8.
<b>offset</b>	To place printed output sets in slightly different positions from each other in an output bin for easy separation of collated sets.
<b>operand</b>	That which is acted upon, for example, data, in an operation or process.
<b>operating system</b>	Software that controls the low-level tasks in a computer system, such as input or output and memory management. The operating system is always running when the computer is active.

<b>orientation</b>	In reference to image area, describes whether the printed lines are parallel to the long edge of the paper (landscape) or the short edge of the paper (portrait).
<b>origin</b>	In reference to image area, the upper left corner of a sheet.
<b>overprint ratio</b>	Maximum number of variable data and form characters that can be intersected by a single scan line.
<b>packet</b>	A group of DJDE records terminated by an END command.
<b>page end</b>	Command character (form feed) to terminate the current page.
<b>palette</b>	Predefined set of colors or inks. Different versions are provided with the printer and with host- or PC-based application software.
<b>parameter</b>	Part of a command, other than the keyword. See <i>keyword</i> .
<b>parse</b>	To read or interpret a command; to build up a parameter list from information within a command.
<b>PCC</b>	Printer Carriage Control.
<b>PDE</b>	Page Description Entry.
<b>PDL</b>	Print Description Language. Language used to describe printing jobs to a laser printing system. PDL describes the input (type, format, characteristics), performs the processing functions (logical processing), and describes the output (type, format, font selection, accounting options).
<b>PE</b>	Phase encoded.
<b>physical page</b>	Actual page size your printer uses to print a form.
<b>pitch</b>	Width of a fixed-pitch font expressed in characters per horizontal inch.
<b>pixel</b>	Acronym for picture element. Smallest addressable point of a bitmapped screen that can be independently assigned color and intensity.
<b>point</b>	In Xerox laser printing systems, a 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 an inch.
<b>portrait</b>	Orientation in which text and images are positioned parallel to the short edge of the paper.



	<b>ppm</b>	Pages per minute.
<b>Print</b>	<b>Description</b>	<b>Language</b>
		See <i>PDL</i> .
	<b>print file</b>	Portion of the system disk memory (up to 4 MB) reserved for temporary storage of formatted pages for printing. Pages are retained until they are delivered to the output tray.
	<b>PSC</b>	Printer Subsystem Controller.
	<b>query</b>	Request for data or other information, entered by an operator while the system is processing.
	<b>record</b>	A line of data as defined in the RECORD command.
	<b>report</b>	A single output data set, delimited by an RSTACK command or as a file. In setting a separation boundary, reports are subsets of a job.
	<b>resolution</b>	Number of dots per inch (dpi) or spots per inch (spi). The greater the number of dots, the higher the resolution and the clearer the image. The terms dots, spots, and pixels are synonymous.
	<b>scale</b>	To adjust font or image size according to given proportions.
	<b>sequential</b>	1. In numeric sequence, usually in ascending order. 2. A file structure in which records are written one after another and cannot be randomly accessed.
	<b>set</b>	Multiple copies of the same report.
<b>simplex</b>	<b>printing</b>	Printing on one side of the page.
	<b>spi</b>	Spots per inch. See <i>resolution</i> .
	<b>spot</b>	A picture element imaged by the printer. Synonymous with <i>dot</i> and <i>pixel</i> .
	<b>statement</b>	Detailed instructions in a program step, written according to specific rules called syntax.
	<b>stock</b>	User-defined name in the JSL that specifies a certain type of paper for printing a job.
	<b>stockset</b>	Collection of stocks to be used on a print job. See also <i>stock</i> .
	<b>string</b>	Connected sequence of alphanumeric characters treated as one unit of data by a program.

<b>syntax</b>	Rules governing the structure of expressions in a programming language.
<b>system page</b>	Maximum area in which text and graphics can be imaged on a printing system.
<b>tape density</b>	Expression of the format of a magnetic tape measured in number of bytes that can be stored per inch of tape.
<b>TOF</b>	Top of form.
<b>two-up</b>	Printing two logical pages on one side of a physical page.
<b>UCSB</b>	Universal Character Set Buffer.
<b>UCS</b>	Universal Character Set.
<b>variable data</b>	Changeable information which is merged with a standard document to create specialized or personalized versions of that document. Variable data is not a part of a form design, but varies from page to page.
<b>virtual page</b>	Page area selected by a forms designer for printing.
<b>vpos</b>	Vertical positioning.
<b>wildcard</b>	Character (usually an asterisk *) which can be inserted into a command string to indicate that it may represent one or more characters in that position.
<b>XDDI</b>	Xerox Dynamic Document Interface.
<b>xdot</b>	Unit of measurement representing a fraction of an inch. May also be referred to as a picture element (pixel) or spot; for example, 1/600 spots per inch (spi).
<b>xerographic mode</b>	Either of two possible printer configurations: 1. Black mode which allows printing with black dry ink only. 2. Highlight mode which enables both highlight color and black printing.
<b>XNS</b>	Xerox Network Systems.
<b>XPAF, XPF</b>	Xerox Printer Access Facility.

## Numerals

- 3211 emulation
  - commands 4,1-4, D-5
  - UCSB, 1-11 to 1-12, D-2 to D-4
- 4135 LPS compatibility, E-1 to E-6
- 4245 emulation
  - commands, D-5
  - ignoring UCSB, 1-12
  - protocol emulation, 1-4
- 4850 Highlight Color compatibility, E-1 to E-6

## A

- ABNORMAL command
  - description, 3-1
  - parameters
    - ERROR, 3-1 to 3-2
    - OTEXT, 3-2
    - SECURITY, 3-3 to 3-4
- ACCT command
  - description, 3-4
  - example, 3-6
  - parameters
    - DEPT, 3-4 to 3-5
    - USER, 3-5 to 3-6
- ALTER command, 5-1
- ANSI carriage control commands, 1-4
- ASCII
  - string constants, 1-16
  - tape codes, 1-2
- ASSIGN command, 5-2
- automatic recovery procedures, 3-2

## B

- BANNER command
  - description, 4-1
  - parameters
    - HCOUNT, 4-1
    - HJOBNO, 4-2
    - HRPTNA, 4-2
    - TCOUNT, 4-2 to 4-3
    - TEST, 4-3
    - TYPE, 4-4
- banner page
  - detection, 1-5, 4-1
  - interleaved graphics, 6-6
- BATCH command, 5-2
- batch mode, graphics, 6-1, 6-5 to 6-7
- BCD tape codes, 1-2
- BDELETE command
  - description, 4-4
  - parameter, TEST, 4-4 to 4-5
- BEGIN command, 5-3
- binary codes, D-1

- BFORM command, D-1
- BLOCK command
  - description, 2-1
  - example, 2-4
  - overriding, 1-3
  - parameters
    - ADJUST, 2-1
    - CONSTANT, 2-1 to 2-2
    - FORMAT, 2-2
    - LENGTH, 2-2
    - LMULT, 2-2 to 2-3
    - LTHFLD, 2-3
    - OFFSET, 2-3
    - POSTAMBLE, 2-3
    - PREAMBLE, 2-4
    - ZERO, 2-4

## block

- constants, 6-3 to 6-4
- mode, graphics, 6-1, 6-5
- BOF command, 5-4
- BSELECT command
  - description, 4-5
  - example, 4-5 to 4-6
  - parameter, TEST, 4-5
- bypass transport option, 3-9

## C

- C text command, 5-4
- CANCEL command, 5-4 to 5-5
- card-image format, 5-8 to 5-9
- catalog level, JSL, 1-19
- channel-attached, 1-4
- channel positions, 1-10
- characters
  - code assignments, B-1 to B-6
  - string constants, 1-16
  - types, 1-40
- CLUSTER.LIB file, 7-1 to 7-2
- clusters
  - applications
    - mixing, 7-8
    - OTEXT, 7-6
    - STOCKSET, 7-6 to 7-7
  - creating, 7-4 to 7-6
  - names, 7-3
  - printing, 7-1, 7-6 to 7-8
  - STOCKSET application, 7-6 to 7-7
- CME command
  - description, 3-6
  - parameters
    - CONSTANT, 3-6 to 3-7
    - FONT, 3-7 to 3-8
    - LINE, 3-8

- POSITION, 3-9
- CMEs
  - multivolume processing, 1-3
  - copy sensitive, 1-5
- CODE command
  - DATA segment, 1-7
  - description, 2-5
  - parameters
    - ASSIGN, 2-5
    - DEFAULT, 2-5 to 2-6
- coding, JDL, 1-26
- COLLATE command, 5-5
- commands
  - constants
    - string, 1-15 to 1-17
    - value, 1-15
  - data definition, 1-27, 2-1 to 2-30
  - elements, 1-12
  - identifiers, 1-13
  - input processing, 2-1 to 2-29
  - keywords, 1-14
  - logical processing, 1-37 to 1-41, 4-1 to 4-31
  - length, 1-13
  - parameter options, 1-14
  - print
    - control, 1-28 to 1-37, 5-1 to 5-27
    - format, 1-27 to 1-28, 3-1 to 3-49
    - replacement, hierarchy, 1-20 to 1-21
    - summary, PDL, A-1 to A-6
    - see also *JSL, structure*
  - comments, JSL, 1-19 to 1-20
  - comparisons
    - masked, 1-40 to 1-41
    - string, 1-39 to 1-40
  - compatibility, 4135/4850, E-1 to E-6
  - compressed format, see *format, packed data*
  - constants
    - graphics, 6-3 to 6-4
    - string, 1-15 to 1-17
    - value, 1-15
  - COPIES command, 5-5, 5-6
  - Copy Modification Entries, see *CME*
  - creating JDL, 1-25 to 1-26
  - CRITERIA command
    - description, 1-37, 1-39, 4-6 to 4-7
    - parameters
      - CHANGE, 4-7
      - CONSTANT, 4-8
      - LINENUM, 4-8
      - VALUE, 4-9 to 4-11
- D**
- data
  - definition commands, 1-27, 2-1 to 2-30
  - formats, packed, 1-2
  - IBMONL default, 1-4
  - inputting streams, 1-1
  - representation, 1-2
- DATA command, 5-6
- DATA segment, 1-7
- decimal constants, see *value constants*
- defaults
  - IBMONL, 1-4
  - print jobs, 1-18
  - type assignments, 1-40
- delimiters, batch stream, 6-6
- DEPT command, 5-6
- DESTINATION command, 5-6
- DJDEs
  - application, 1-33 to 1-34
  - job parameter, restrictions, 1-35 to 1-37
  - graphic
    - definitions, 6-8
    - requirements, 6-2 to 6-5
  - IMAGE records, 6-5
  - invoking processing, 2-6
  - online processing, 1-5
  - operator information pages, 1-35
  - orientation, 1-29 to 1-31
  - paper stock, 7-2 to 7-3
  - PDL equivalents, A-1 to A-6
  - print control commands, 1-28 to 1-37, 5-1 to 5-27
  - printing, 1-1
  - record specification, 1-32 to 1-33
- document-interleaved
  - file transfers, 6-7 to 6-8
  - rules, 6-6
  - tape format, 6-2 to 6-3
- downloading files, 1-9
- dump
  - online, 1-6 to 1-8
  - tape formats, 2-20
- DUPLEX, 5-7
- Dynamic Job Descriptor Entries, see *DJDE*
- E**
- EBCDIC
  - string constants, 1-16 to 1-17
  - tape codes, 1-2
  - translation, 1-7
- END command, 5-7
- ENDJOB command, 1-7
- examples
  - ACCT, 3-6
  - BLOCK, 2-4
  - BSELECT, 4-5 to 4-6
  - LINE, 3-19
  - OUTPUT, 3-38
  - PDE, 3-41 to 3-42
  - RAUX, 4-13
  - RDELETE, 4-13 to 4-14
  - RECORD, 2-15 to 2-16
  - ROFFSET, 4-16
  - RRESUME, 4-22
  - RSTACK, 4-26
  - STOCKSET, 3-46
  - TABLE, 4-30 to 4-31
  - VFU, 3-48 to 3-49
  - VOLUME, 2-30
- EXPORT command
  - description, 3-9 to 3-10
  - parameters
    - SEPARATORS, 3-10

SNUMBER, 3-11  
 SPLIT, 3-11  
 SRECOVER, 3-12  
 STIMING, 3-12 to 3-13

**F**

FCB, 1-6, 1-10, D-1  
 FEED command, 5-7  
 FILE command, 1-9, 5-8 to 5-10  
 files  
   downloading, 1-9  
   transferring, graphics, 6-7  
 Finishing Configuration (FCG) Utility, 3-12  
 FOLD command, 1-11  
 FONTINDEX command, 5-11  
 FONTS command, 5-11 to 5-12  
 format  
   downloading files, 1-9  
   dump, 1-7 to 1-8  
   host tape, 1-2  
   logical processing commands, 1-38  
   packed data, 1-2  
   record, 1-3  
 FORMAT command, 5-12  
 FORMS command, 5-12 to 5-13  
 Forms Control Buffer, see *FCB*  
 forms length, 1-10

**G**

graphics  
   DJDE descriptions, 6-8  
   exception page, 3-1  
   file transfers, 6-7  
   formats  
     online, 6-5 to 6-7  
     tape, 6-2 to 6-5  
   PDL command options, 6-7 to 6-8  
   printing, 6-8  
   processing modes, 6-1 to 6-2  
   restrictions, 6-8 to 6-9  
 GRAPHICS command, 5-13

**H**

H2 string constants, 1-17 to 1-18  
 H6 string constants, 1-17 to 1-18  
 HEADER segment, 1-7  
 hexadecimal  
   codes, D-1  
   string constants, 1-16  
 hierarchy, command replacement, 1-20 to 1-21  
 Honeywell 2000/6000, 1-2  
 host  
   downloading files, 1-9  
   tape formats, 1-2  
 HOSTCOPY  
   command, 1-9  
   utility, 1-9, 6-5

**I**

ID level, JSL, 1-18  
 IDEN command  
   description, 2-6

DJDEs, identifying, 1-5  
 parameters  
   OFFSET, 2-6  
   OPRINFO, 1-35, 2-7  
   PREFIX, 2-7  
   SKIP, 2-8  
   START command, 1-32  
 identifiers, command, 1-13  
 IMAGE command, 5-14 to 5-15  
 images file, 6-7  
 input data, source, 1-1  
 interleaved graphics, transferring, 6-8  
 ITEXT command, 5-15

**J**

JDE  
   command, 5-15  
   level, 1-19  
   primary element, JSL, 1-1  
 JDL  
   coding, 1-26  
   command, 5-16  
   creating, 1-25 to 1-26  
   JSL file, 1-1  
   level, 1-18 to 1-19  
   name, 1-18  
   source files, host, C-3  
 Job Descriptor Entry, see *JDE*  
 Job Descriptor Library, see *JDL*  
 Job Source Library, see *JSL*  
 job level, string constants, 1-19  
 JSL  
   compilation, 1-41  
   components, required, 1-25  
   ID level, 1-18  
   PDL commands, 1-1  
   structure  
   catalog level, 1-19  
   comments, 1-19 to 1-20  
   END command, 1-20  
   JDL name, 1-18  
   replacement hierarchy, 1-20 to 1-21  
   system level, 1-18 to 1-19  
   see also *commands*

**K**

keywords  
   command, 1-14  
   parameter, 1-14  
   string constants, 1-15

**L**

laser printing system, see *LPS*  
 length  
   block, 2-20  
   command, 1-13  
   record, online, 1-5 to 1-6  
 library, PDL, 1-24  
 LINE command  
   description, 3-13  
   example, 3-19  
   parameters

DATA, 1-4, 1-6, 3-13  
 FCB, 3-14  
 FONTINDEX, 3-14 to 3-15  
 MARGIN, 3-16  
 OVERPRINT, 3-16  
 PCC, 3-17  
 PCCTYPE, 3-17  
 UCSB, 3-18  
 VFU, 3-18  
 LMODIFY command  
   description, 4-11  
   parameters, TEST, 4-11 to 4-12  
 logical processing commands, 1-37 to  
   1-41, 4-1 to 4-31  
 LPS  
   file transfers, 1-9  
   host formats, 1-2  
 LPS-labeled files, 5-9 to 5-10  
  
**M**  
 magnetic tape, 1-1  
 MARGINS, 5-16  
 masking  
   characters, 4-30 to 4-31  
   comparisons, 1-40 to 1-41  
 marker page, 3-2 to 3-3  
 MESSAGE command  
   description, 3-20  
   parameters  
     ITEXT, 3-20  
     OTEXT, 3-20 to 3-21  
 mode  
   batch, graphics, 6-1  
   block, graphics, 6-1  
   move, graphics, 6-2  
   offline, 1-1 to 1-3  
   online, 1-4 to 1-12  
   random, graphics, 6-2  
 MODIFY command, 5-16 to 5-17  
 move mode, graphics, 6-2  
 multiple passes, see *processing, multivolume*

**N**  
 NO-OP command, 1-7  
 noninterleaved  
   online formats, 6-5  
   tape format, 6-2  
 NUMBER command, 5-17

**O**  
 octal, string constants, 1-17  
 offline  
   processing, 1-1 to 1-3  
   specifications, C-1 to C-4  
 online  
   default  
     PCCTYPE, 1-4  
     record length, 1-5 to 1-6  
   dump, 1-6 to 1-8  
   graphics formats, 6-5 to 6-7  
   recovery, 1-6  
   specifications, D-1 to D-5

online-specific commands, 1-4  
 operator  
   information pages, 1-35  
   tasks, 7-1  
 orientation, DJDE, 1-29 to 1-31  
 OTEXT command, 5-18, 7-6  
 OUTPUT command  
   description, 3-21 to 3-23  
   graphics, parameters, 6-7  
   example, 3-38  
   parameters  
     BFORM, 3-23 to 3-24  
     COLLATE, 3-24  
     COPIES, 3-24  
     COVER, 3-24 to 3-25  
     CYCLEFORMS, 3-25 to 3-26  
     DENSITY, 3-26  
     DESTINATION, 3-26  
     DUPLEX, 3-26 to 3-27  
     FACEUP, 3-27  
     FEED, 3-27  
     FORMAT, 3-28 to 3-29  
     FORMS, 1-5, 3-30  
     GRAPHICS, 3-30 to 3-31, 6-2  
     IMAGE, 3-31  
     MODIFY, 1-5, 3-32  
     NTO1, 3-32 to 3-33  
     NUMBER, 3-33 to 3-34  
     OFFSET, 3-34  
     PAPERSIZE, 3-34 to 3-35  
     PURGE, 3-35  
     RESOLUTION, 3-35  
     SHIFT, 3-36  
     SIZING, 3-36 to 3-37  
     STAPLE, 3-37  
     STOCKS, 3-37  
     UNITS, 3-38  
   restrictions, hierarchy, 3-21  
 OVERPRINT command, 5-18 to 5-19  
 overprint ratio, 3-23  
 overprinting, 3-15

**P**  
 Page Description Language, see *PDL*  
 page-interleaved, graphics, 6-3 to 6-4, 6-6  
 page-oriented, DJDEs, 1-29 to 1-30  
 paper stock, 7-2 to 7-3  
 parameter  
   job, restrictions, 1-35 to 1-37  
   keywords, 1-14  
   options, 1-14  
 PCC command  
   description, 2-8  
   ignored, 1-33  
   parameters  
     ADVTAPE, 2-8  
     ASSIGN, 2-9 to 2-10  
     DEFAULT, 2-10 to 2-11  
     INITIAL, 2-11  
     MASK, 2-11  
   record, dividing, 1-3  
 PDE command

- description, 3-38 to 3-40
  - example, 3-41 to 3-42
  - parameters
    - BEGIN, 3-40
    - FONTS, 3-40 to 3-41
    - Pmode, 3-41
  - PDL commands
    - library, 1-24
    - graphics, 6-7 to 6-8
    - summary, A-1 to A-6
  - Pmode command, 5-19
  - PPI, 1-10
  - print
    - control commands, 5-1 to 5-27
    - file, saturation, 1-3
    - format
      - commands, 3-1 to 3-49
      - using, 1-27 to 1-28
  - Print Position Indexing, see *PPI*
  - printing
    - clusters, 7-1 to 7-2
    - job, 1-41
    - paper stock, 7-2 to 7-3
  - procedures
    - clusters, creating, 7-4
    - JDL, creating, 1-25 to 1-26
  - processing, multivolume, 1-3, 2-27
  - programmer tasks, 7-1
  - protocol emulation, 1-4
- R**
- random mode, graphics, 6-2, 6-8
  - RAUX command
    - description, 4-12
    - example, 4-13
    - options, 7-3
    - parameter, TEST, 4-12
  - RDELETE command
    - description, 4-13
    - example, 4-13 to 4-14
    - parameter, TEST, 4-13
  - record
    - constants, 6-3 to 6-4
    - defaults, IBMONL, 1-4, 2-22
    - formats, 1-3
    - length, online, 1-5 to 1-7
    - orientation, 1-29, 1-31
    - specification, DJDE, 1-32 to 1-33
    - setup, graphics, 6-3 to 6-7
    - structure, 1-3
  - RECORD command
    - constants, 6-3 to 6-4
    - description, 2-12
    - example, 2-15 to 2-16
    - input data format, 1-3
    - parameters
      - ADJUST, 2-12
      - CONSTANT, 2-12
      - FORMAT, 2-12 to 2-13
      - LENGTH, 1-6 to 1-7, 2-13
      - LMULT, 2-13
      - LTHFLD, 2-14
      - OFFSET, 2-14
      - POSTAMBLE, 2-14
      - PREAMBLE, 2-14
      - STRUCTURE, 2-15
    - overriding, 1-3
    - recovery
      - online, 1-6
      - point, system rollover, 1-3
      - procedures, automatic, 3-2
    - replacement hierarchy, commands, 1-20 to 1-21
    - reports, separating, 1-5
    - restrictions, job parameter, 1-35 to 1-37
    - RFEED command
      - description, 4-14
      - parameter, TEST, 4-14
    - RFORM command, 5-19
    - ROFFSET command
      - description, 4-15
      - example, 4-16
      - parameter
        - PASSES, 4-15
        - TEST, 4-15 to 4-16
    - ROUTE command
      - description, 3-42
      - parameters
        - RFORM, 3-42
        - RTEXT, 3-42 to 3-44
    - RPAGE command
      - description, 4-16
      - parameters
        - SIDE, 4-16 to 4-18
        - TEST, 4-19
        - WHEN, 4-19 to 4-20
    - RRESUME command
      - description, 4-21
      - example, 4-22
      - parameters
        - BEGIN, 3-39, 4-21
        - TEST, 4-21 to 4-22
    - RSELECT command
      - description, 4-23
      - parameter, TEST, 4-23
    - RSTACK command
      - description, 4-23
      - example, 4-26
      - parameters
        - ACCTINFO, 4-24
        - DELIMITER, 4-24
        - HRPTNA, 4-25
        - PRINT, 4-25
        - TEST, 4-26
    - RSUSPEND command
      - description, 4-27
      - parameters
        - BEGIN, 4-27
        - TEST, 4-28
    - RTEXT command, 5-20
- S**
- SAVE command, 5-21
  - separating reports, 1-5

SEPARATORS command, 5-21  
SHIFT command, 5-22  
SIDE command, 5-23  
SNUMBER command, 5-24  
specifications  
    offline, C-1 to C-4  
    online, D-1 to D-5  
SPLIT command, 5-25  
SRECOVER command, 5-26  
START command, 1-6, 1-10, 1-32, 7-6  
STOCKS command, 5-27  
STOCKSET command  
    applications, 7-6 to 7-8  
    defining, 7-3  
    description, 3-44, 7-6  
    example, 3-46  
    parameters  
        ASSIGN, 3-44  
        INIFEED, 3-44 to 3-45  
        SYSPAGE, 3-45  
string  
    comparisons, 1-39 to 1-40, 2-17  
    constants, 1-15 to 1-17  
system  
    level, 1-18 to 1-19  
    recovery, online, 1-6  
    rollover, 1-3

## T

TABLE command  
    description, 1-37, 1-39, 4-28  
    example, 4-30 to 4-31  
    parameters  
        CONSTANT, 1-41, 4-29  
        MASK, 1-39, 1-41, 4-29  
tape formats  
    graphics, 6-2 to 6-5  
    host, 1-2  
TCODE command  
    coding, 2-17 to 2-18  
    description, 2-17  
    parameters  
        DEFAULT, 2-18  
        TASSIGN, 2-18 to 2-19  
        TRESET, 2-19 to 2-20  
TEST expression definition, 1-39  
TOF command, 5-27  
type assignments, default, 1-40

## U

UCSB, 1-11 to 1-12, D-2 to D-4  
UNFOLD command, 1-11  
Universal Character Set Buffer, see *UCSB*

## V

value constants, 1-15  
variable names, string constants, 1-15  
vertical format controls, see *VFU*  
VFU, 1-10, 1-18  
VFU command  
    description, 3-46 to 3-47  
    example, 3-48 to 3-49

ID level, 1-18  
parameters  
    ASSIGN, 3-47  
    BOF, 3-48  
    TOF, 3-48  
VOLUME command  
    data representation, 1-2  
    description, 2-20  
    example, 2-30  
    multivolume processing, 1-3  
    online processing, 1-4  
    parameters  
        BMULT, 2-20  
        CODE, 1-2, 2-20 to 2-21  
        EOV, 1-3, 2-21 to 2-22  
        HOST, 1-4, 2-22 to 2-23  
        INTERPRESS, 2-23 to 2-24  
        LABEL, 2-24  
        LCODE, 2-24  
        LPACK, 2-25  
        MAXLAB, 2-25  
        MINLAB, 2-25  
        OPTIMIZE, 1-4, 1-6, 2-26  
        OSCHN, 2-26  
        OSHDP, 2-27  
        PLABEL, 2-27  
        RMULT, 2-27  
        RSAT, 1-3, 2-27 to 2-28  
        TCODE, 1-40, 2-28  
        UNPACK, 2-28  
        VCODE (XDDI), 2-29

## X

XDDI commands  
    EXPORT, 3-9 to 3-13  
    VOLUME, VCODE, 2-29  
XDOT, 5-14  
Xerographic Mode Switching (XMS), 2-7