XEROX

Xerox 4850/4890 HighLight Color Laser Printing Systems Print Description Language Reference

Version 5.0
November 1994
720P93590

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

Printed in the United States of America

Publication number: 720P93590

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 VP software. The typefaces used are Optima, Terminal, and monospace.

Table of contents

Introduction

The *Xerox 4850/4890 HighLight Color LPS 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:

{ }	Curly brackets indicate required selection of characters, values, or keywords.
...	Ellipses indicate that you can repeat a parameter or option, or list a series of parameters or options.
[ ]	Square brackets indicate optional command characters, values, or keywords.
|	Vertical bars are used to separate option choices. The vertical bar stands for “or.”
bold	Bold is used for characters you enter at the command line.
italics	Italics is used for variable names or numbers.
`TERMINAL FONT`	Terminal or monospace font is used to represent text that displays on the system screen.
underline	System default parameters are underlined.
UPPERCASE	Uppercase letters are used for command and parameter keywords.

1. Overview

This chapter contains basic information on using Xerox print description language (PDL) to create and control print jobs.

For a job to be printed on a Xerox Laser Printing System (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, highlight color, 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 JSL file is called a job descriptor library (JDL).

The primary element of a JSL is a job descriptor entry (JDE). The key words "JDE" and "JOB" are used interchangeably when coding JSL. This sometimes causes confusion from an ONLINE host perspective because a host job, delimited by the RSTACK and/or BANNER commands (discussed in the "Logical processing commands" chapter), is processed by the LPS as a report, and thus one LPS job may consist of multiple host jobs/reports sent in sequence. A particular JOB in a JDL may be generally used to print various reports (e.g., a 2-up format), or specific to a particular report format. A JDE name is specified along with the JDL name in order to define print job characteristics as the job is started. DJDEs may be used to select a different JDE, or otherwise alter formatting on the fly. Any characteristics not specified explicitly in the JSL will be derived from system generation settings or other system defaults.

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 LPS processes data tapes in standard host formats. Refer to the "Character code assignments" 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 binary coded decimal (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 correspondence between standard recording codes and printed characters are contained 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. One method uses the Honeywell 6000 (T4X3), while Honeywell 2000 uses a slightly different method of packing (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).

Record formats

All tape records input to the printing system are either blocked or unblocked with a fixed length, a variable length, or an undefined format. The 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 command” 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 “RECORD command” section of 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 “Copy modification entries“ section 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 want to print each volume on a separate system. This requires that each volume be processed independently. Refer to the VOLUME command (EOV parameter) 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 before performing a forward tape space to the recovery point:

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

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

Online mode

A channel-attached LPS can receive input from the host computer in online 3211 or 4245 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 may be coded for the 4245.

4245 processing mode features

The 4245 processing mode 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 can also be coded for the 4245.

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 two functions:

- Allows the LPS to accept data from the online channel
- Allows PDL to use a different set of defaults for the RECORD and LINE commands.

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

```
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 “VOLUME command” section in the “Data definition commands” chapter for additional information.

DJDE processing

Printing is controlled through parameters from a user-defined .JDL file. These parameters 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 description 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. 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) that 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 number of times the data needs to be modified. Thus, 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 printed successfully 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 that 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 FCBs. If FCB=IGNORE is invoked, this restriction is not significant.

- For print record lengths from 215 bytes to 2 140 bytes, you must not use the OPTIMIZE parameter of the VOLUME command, which provides for 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. 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 for the OPTIMIZE parameter. Refer to the "VOLUME command" section of the "Data description 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, appearing later in this chapter, shows a complete online job source library (JSL).

Online recovery

After a system failure, data acquired from the host and stored on 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 KB. 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 either 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 the operator guide for your

Xerox 4850 or 4890 LPS. Figure 1-1, appearing later in this chapter, is a sample printed online dump.

Dump format

Each host command directs the printer to perform an operation or provides control information used during printing. The control information could be printed 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 for each of these commands contains a HEADER and DATA segment. The HEADER segment consists of the fields shown in table 1-1.

Table 1-1. **Print format command usage**

Field	Definition
HOST COMMAND	The English translation of the host command in hexadecimal
END STATUS	One byte of status in hexadecimal transmitted to the host
SEQ#	Incremental count in decimal of the host commands received during a dump session. The maximum number generated is 999,999
LENGTH	Length in decimal of the data transmitted
DATE and TIME	Date and time when the host command was received by the dump processor. The time includes tics of seconds where one tic is approximately one millisecond.

The DATA segment consists of the input data in hexadecimal transmitted to or from the host as well as 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 these restrictions:

- DJDEs, BANNER, and other logical processing functions are not processed when the dump job is running, so you must enter an operator ENDJOB command to end the dump.
- 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, as 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 HIP START ... command is accepted, but the host interface processor (HIP) is unable to submit a print job while the system is online.

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

```
HOST COMMAND: 04  SENSE                          END STATUS: 0C     SEQ.#: 1          LENGTH: 6      13-APR-83 03:12:52.655

0000    00000000  0000                                                                *                                *

HOST COMMAND: 01  WRITE WITHOUT SPACING          END STATUS: 04     SEQ.#: 2          LENGTH: 1      13-APR-83 03:12:52.671

0000    00                                                                          *                                *

HOST COMMAND: 07  DIAGNOSTIC GATE                END STATUS: 04     SEQ.#: 3          LENGTH: 0      13-APR-83 03:12:52.873

HOST COMMAND: 06  CHECK READ                     END STATUS: 0C     SEQ.#: 4          LENGTH: 1      13-APR-83 03:12:52.886

0000    A0                                                                          *                                *

HOST COMMAND: 12  READ FCB                       END STATUS: 0C     SEQ.#: 5          LENGTH: 72     13-APR-83 03:12:52.906

0000    00000000  01000000  00000000  00000000   00000000  00000000  00000000  00000000  *                                *
0020    00000000  00000000  00000000  00000000   00000000  00000009  00000000  00000000  *                                *
0040    00000000  00000010                                                          *                                *

HOST COMMAND: 63  LOAD FCB                       END STATUS: 0C     SEQ.#: 6          LENGTH: 71     13-APR-83 03:12:52.936

0000    00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000  *                                *
0020    00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000  *                                *
0040    00000000  00000110                                                          *

HOST COMMAND: 8B  SKIP IMMEDIATE TO CHANNEL 1    END STATUS: 04     SEQ.#: 7          LENGTH: 0      13-APR-83 03:12:53.006

HOST COMMAND: 01  WRITE WITHOUT SPACING          END STATUS: 04     SEQ.#: 8          LENGTH: 100    13-APR-83 03:12:53.016

0000    616140D1  D6C240D7  D3C1C9D5  40404040   40404040  40404040  40404040  40404040  *// JOB PLAIN                     *
0020    40404040  40404040  40404040  40404040   40404040  40404040  40404040  40404040  *                                *
0004    40404040  40404040  C4C1E3C5  40F0F461   F1F361F8  F36BC3D3  D6C3D240  F1F561F3  *        DATE 04/13/83,CLOCK 15/3*
0060    F561F2F5                                                                    *5/25                            *

HOST COMMAND: 8B  SKIP IMMEDIATE TO CHANNEL 1    END STATUS: 04     SEQ.#: 9          LENGTH: 0      13-APR-83 03:12:53.052

HOST COMMAND: 09  WRITE AND SPACE 1 LINE         END STATUS: 04     SEQ.#: 10         LENGTH: 84     13-APR-83 03:12:53.262

0000    40D79989  95A38584  4082A840  A3888540   F7859996  A740F9F7  F0F040D6  9560D389  * Printed by the Xerox 4050 On-Li*
0020    958540C3  969497A4  A3859940  D7998995   A3899587  4032A8A2  A3859440  40404040  *ne Computer Printing System     *
0040    40D7C1C7  C540F0F0  F1404040  40D3C9D5   C540F0F1                                * PAGE 001    LINE 01            *
```

Downloading files from the host to the LPS

There are three methods of 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, Y, and TSK. In addition, card-image format files are restricted to destination file types CMD, FSL, JSL, MSC, PCH, TMP, and TPF.

HOSTCOPY

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

- User-created card-image files
- LPS-labeled files (such as font, logo, and patch tapes). Refer to the *Xerox 4850/4890 HighLight Color LPS System Programming and 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 4850/4890 HighLight Color LPS System Programming and 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 comes back into effect.
- 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 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, when a new UCSB is transmitted from the host, and when a new FOLD or UNFOLD command is received.

The selection of a folded or unfolded translate table is based on the FOLD and UNFOLD 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 by 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 comes back into effect 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, ; ( ), DJDE syntax errors result.

Code 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.

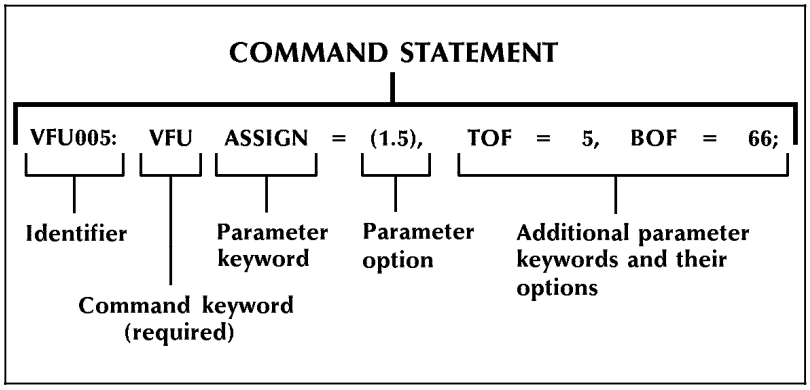
A PDL command consists of as many as nine elements:

- 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 with its parameter options
- Parameter option (right part)
- Parentheses, which group 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.

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



This PDL command 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 4850/4890 HighLight Color LPS System Programming and Administration Guide* for information on using the editor.

Identifiers

The purpose of an identifier is to provide a label for a command so 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 alphabetic (letter). *dd* indicates that the alphabetic 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, VFU 1: VFU; is invalid, whereas VFU1: VFU is valid.
- Must be separated from the command with a colon, such as VFU1: VFU in figure 1-5.
- 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;
- Must be coded with either the first three characters of the command or the whole the command. For example, RECORD can be coded as REC 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, FORMS, DUPLEX, and COPIES are parameter keywords of the OUTPUT command:

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

Parameters have these requirements:

- Must be coded with the first three characters of the command or the whole command. For example, DUPLEX can be coded as DUP or DUPLEX.
- Must be followed by an equal sign (=) and at least one parameter option, as shown in figure 1-2.

Parameter options

Most parameters have multiple parameter options available. Parameter options have the following requirements:

- Are placed on the right side of the equal sign (=)
- Multiple options for the same parameter must be enclosed in parentheses
- Can consist of 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 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, and also as decimal numbers for use with LMODIFY logical processing.

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 parameter option keywords.

Variable names

You may use string constants to name objects such 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 parameter option 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 immediately be preceded by the characters 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 immediately be 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 characters A apostrophe (A’) and followed by an apostrophe character. For example:

IDEN PREFIX=A’ABC’;
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 exclamation characters (!!) are necessary to represent one actual ! character when printing. The two-character sequence (!!) results in one byte.

EBCDIC

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

IDEN PREFIX=E’ABC!C4EFG’

is equivalent to the hexadecimal:

IDEN PREFIX=X’C1C2C3C4C5C6C7’

Each character represented in EBCDIC results in one byte. Each three-character sequence representing a character in hexadecimal results in one byte. EBCDIC is the default, therefore, E ’xxx’ is usually not required. It is required, however, when the EBCDIC string has a hexadecimal representation in it, as in the example above.

Octal

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

BLOCK CONSTANT=O'07070707';

H2 and H6

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

BLOCK CONSTANT=H2'373737'

BLOCK CONSTANT=H6'373737'

Since H2000 and H6000 BCD are defined as 6-bit codes (refer to 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.

Repeat count

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, later in this chapter, has a repeat count.

Decimal string constants

Decimal string constants should be coded using the same character code as the text data. For example:

T1: TABLE CONSTANT=E'100,000' /*EBCDIC Decimal 100000 */

T2: TABLE CONSTANT=A'-0.01' /*ASCII -0.01 */

T3: TABLE CONSTANT=H6'0' /* H6000 BCD zero */"

JSL structure

PDL commands are organized within a JSL in four groupings called command levels: ID level, system (or JDL) level, catalog level, and 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, appearing later in this chapter.

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, but it is not required. The format for naming a JDL is:

JDL-id: {JDL|SYSTEM};

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

JDLHLC: JDL;

For examples of JDL names refer to Figures 1-4 and 1-5. On Figure 1-4 the JDL name is listed on the top of the figure as: “IBMPDL: SYSTEM;“. On Figure 1-5 the JDL name is listed on the top of the figure as: “JDLONL: JDL;“

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 “JDL compilation” section of this chapter and in the operator guide for your 4850 or 4890.

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. The VFU command is usually coded at this level, as shown with VFU1: VFU in figure 1-4, appearing later in this chapter.

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. The command for creating a catalog name is as follows:

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. The command to create a job name is:

jde-name: {JOBIJDE} [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 want to specify for your print job. For example, in figure 1-4:

2: JOB; INCLUDE = CATPOW.

The commands following the JOBIJDE command specify specifications of the job which differ from the catelog specifications and the system level specifications.

If DFLT 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 “JDL 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 1-5.

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 and DJDE 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 illustrated in figure 1-3.

Figure 1-3. **Hierarchy of command replacement**

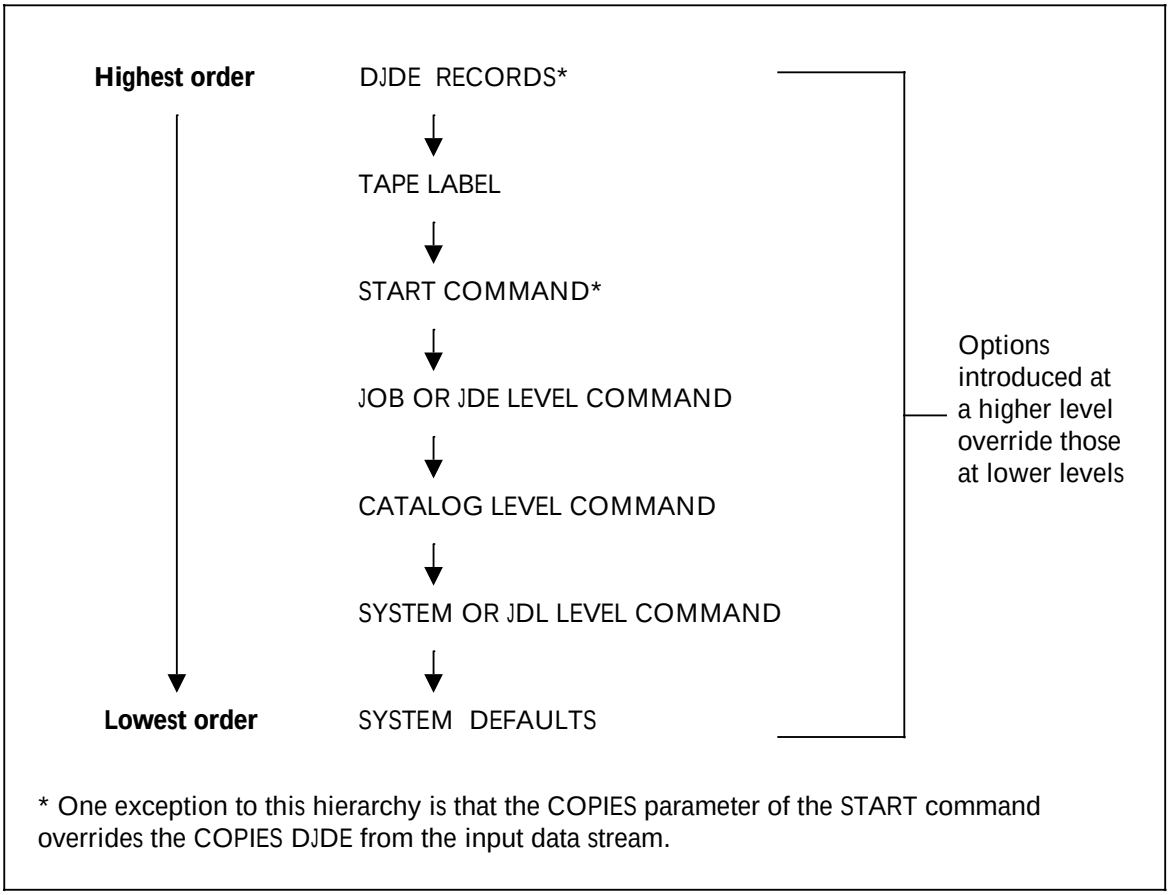


Figure 1-4 shows a coded JDL that contains four jobs. A command to specify the recording code (CODE parameter of the VOLUME command) of the input data appears in three places:

- 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.

PDL library

If you have multiple PDL commands of the same type, such as CMEs, IDRs, and PDEs, you may want to create separate files for them to group similar specifications together and to make your JSLs shorter and more efficient. PDL library files are easy to maintain and modify when compiled. PDL files create separate object files to call out for your reports.

PDL-related library files are .STK for paper stock (FEED= and STOCKS=) specifications, .PDE for format, .IDR for IDR command specifications, and .CME for copy modification entry (MODIFY=) specifications. You can only call out these specifications using DJDEs as separate object files.

Avoid these categories of restricted filenames 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, .TST and .FRM
- System utilities, such as INPUT, EDIT, DSR, IPD, and OCS.

Figure 1-4. **Sample offline JSL**

```
JDLHLC:     JDL;
/* JDL CODING ***COMMENTS ARE BOLDED***/
            /*******ID LEVEL COMMANDS*******/
VFU1:       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;
IDR1:       IDR         ILIST = ('RED', 'BLACK', 'PINK');  /***IDFAULT = 'RED'***/
IDR2:       IDR         ICATALOG = XEROX, PALETTE = 'PICTORIAL',
                        ILIST = ('BLACK', 'RED', 'PALE PINK');  /*IDFAULT = 'BLACK'*/
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=VFU1;
            IDEN        PREFIX = A'DJDE', SKIP = 6, OFFSET = 2,
                        OPERINFO = 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;
              OUTPUT    IDR=IDR1, NUMBER=(1, 1, 0, 1, 'BLACK')
                        FORMAT= PDE1, LOGO = (SIG1, 1.5 IN, 6.0 IN);
2:          JDE         INCLUDE=CATPOW;
              VOLUME    HOST=POWERVS; CODE=PEBCDIC;
              OUTPUT    IDR=IDR2, FORMAT=PDE2, IDFAULT=2, FORMS=STMT3;
              ABNORMAL  ERROR=CONTINUE, IMISMATCH=CONTINUE;
3:          JDE         INCLUDE=CATGRP;
              OUTPUT    COPIES = 5, IDFAULT = 'BLACK', XMP = REPORT;
DFLT:       JDE;
END;
```

Figure 1-5. **Sample online JSL**

```
JDLONL:         JDL;
/** JDL CODING*/
/***COMMENTS ARE BOLDED 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;
RED             IDR             ILIST = ('BLACK', 'RED');
GREEN:          IDR             ILIST = ('BLACK', 'GREEN');
BLUE:           IDR             ILIST = ('BLACK', 'BLUE');
MAGENTA:        IDR             ILIST = ('BLACK', 'MAGENTA');
CYAN:           IDR             ILIST = ('BLACK', 'CYAN');
P1:             PDE             FONTS=(P0612B, 3), BEGIN=(2,1,5), PMODE=PORTRAIT;
P2:             PDE             FONTS=L0112A, BEGIN=(.55, .55);
CME1:           CME             L=34, P=65, INK=2, P= 66, INK=1;
CME2:           CME             L=34, P=65, INK=3, P=66, INK=1;
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),
                                LINENUM=(1,20);
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, OPERINFO=NO;
                  OUTPUT        FORMAT=FMT3, DUPLEX=YES, SHIFT=YES;
HDRP:           JOB;            /*JOB WITH HEADER PAGES ONLY */
                  BANNER        TEST=C1, HCOUNT=2 ,TCOUNT=0;
                  OUTPUT        IDR=RED, NUMBER=(1, 1, 0, 2, 2), FORMAT=PDE1,
                                LOGO=(BNR, 480 DOTS, 300 DOTS);
TRLP:           JOB;            /*JOBS WITH TRAILER PAGES ONLY*/
                  BANNER        TEST=C2, HCOUNT=0, TCOUNT=3;
                  OUTPUT        FORMAT=FMT2, DUPLEX=NO, IRESULT=COLOR,
                                PAPERSIZE=(A3, FORM);
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,
                                INKINDEX=2;
                  OUTPUT        FORMAT=P1, DUPLEX=LONG, IRESULT=BLACK;
END;
```

JDL creation

JSLs are source files you create and compile to create a JDL. There are many methods and elements you can use in creating a JDL. The following sections provide the 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 three elements 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 them specify to run jobs contained within the JDL.
- JOB command: Within each JSL, there must be one JOB command (also referred to as a JDE). 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 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 4850/4890 HighLight Color LPS System Programming and Administration Guide* for instructions if you are not familiar with using the editor.
3. Code the JDL name.
4. Determine the command levels to 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 4850/4890 HighLight Color LPS Forms Creation Reference* for information on creating forms.
9. Identify the 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. Enter the END command to complete the JSL.
14. Compile the JSL to create a JDL file.

There are several things to keep 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 LPS 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 and then make necessary adjustments. You may have to run the job several times during the development process to check your JSL specifications.
- You 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, the point of origin for printing on the page, and the use of highlight color.
- 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 three types of commands:

- Data definition
- Print format
- Logical processing.

The syntax, parameters, and options for each of these types of commands can be found in the corresponding chapters.

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

For expanded information on highlight color, 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 1-2.

Table 1–2. **Data definition command usage**

To specify this	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

These 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 a block contains.

Refer to the “Data definition commands” chapter for a complete description 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, accounting sheets, and highlight color. They also define the placement of data on the page and system responses to error conditions, among many others.

To specify the following print format information, code the corresponding command, as shown in table 1-3.

Table 1-3. **Print format command usage**

To specify this	Use this command
Permitted operator activities, error condition activities, 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 or ink changes within variable data	CME
Identify ink catalogs, palettes, and ink	IDR
Margin control, overprinting, carriage control	LINE
Messages sent to operators during job processing	MESSAGE
Print mode, forms, paper stock, offsetting, stapling, highlight color, 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 identifies the part of the data in each record to print. For this reason, JSLs typically include a LINE command.

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

Printer control (DJDE) commands

Dynamic job descriptor entries (DJDEs) are commands sent through the input data stream that override the printing characteristics specified in a 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) commands” chapter.

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

- The printing system does not stop between reports and does not require operator intervention. The operator starts up a job on the printing system and 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, top-of-form and bottom-of-form assignments may be applied to reports as they are created using DJDEs instead of being stored in the printing system using 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 are listed in each DJDE section in the “Print control (DJDE) commands” chapter.

Page-oriented

Page-oriented DJDEs change specific pages within a report and can change these pages differently in different copies. Such parameters may be placed within the report 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 print 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 following page-oriented DJDEs as described in table 1-4.

Table 1–4. **Page-oriented DJDE usage**

To specify this	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 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
There is a new page descriptor entry (PDE) to control formatting	FORMAT
The form merged on the printed page	FORMS
The ink catalog containing the palettes and inks for subsequent pages	ICATALOG
The ink used when an ink is not specified	IDFAULT
Ink specifications for subsequent pages	IDR
List of inks referenced by the inkindex	ILIST
Field within a data record containing a number that represents the inkindex value for subsequent pages	INKINDEX
Result when objects with different inks overlap	IRESULT
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

Table 1–4. **Page-oriented DJDE usage** (continued)

To specify this	Use this command
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
Ink palette used for subsequent pages	PALETTE
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 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

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 1-5 lists these DJDEs and their uses.

Table 1–5. **Record-oriented DJDE usage usage**

To specify this	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	C *text*
Location and length of printable data within an input record	DATA
Enables files to load to system disk while print job in progress	FILE
Logo name and position on a page or on all pages of a report	LOGO
Instructs system when overprint lines occur	OVERPRINT
Top-of-form number	TOF
Control of xerographic mode switching for a report	XMP

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 1-6.

Table 1–6. **Graphics-related page-oriented DJDEs**

To specify this	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. Example DJDEs 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).
- 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 channel specified, a skip occurs. 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. This 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 JDE and JDL in the DJDE, which cause all processing parameters contained in the JDE to be updated except for those listed in the job parameter modification restrictions section. The modified parameter remains in effect until that parameter is encountered in another DJDE packet or until the end-of-report is reached. At the start of the next report, either of the following occurs:

- If in multi-report 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 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, 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 thus affects the next channel skip.

- You may insert a record after the DJDE packet to cause the page transition and then allow a second skip to channel 1 to position to the correct line. This causes a blank page to be printed if the new channel 1 is less than the old, but will be totally transparent if the new channel 1 assignment is greater (further down the page) than the old.
- In the online environment only, the carriage control on the DJDE record 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. 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 it specifies in figure 1-6:

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

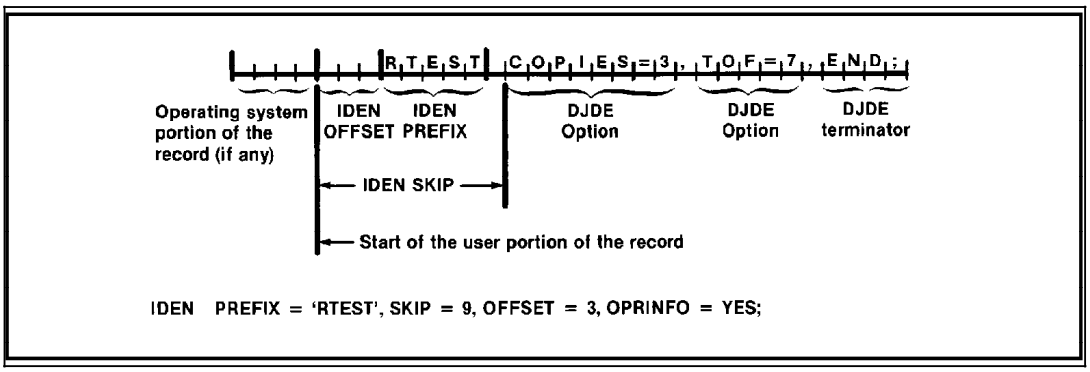
Figure 1-6. **Multiple record DJDE**

```
*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
```

This example of an IDEN command shows the single-record DJDE it specifies:

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

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



DJDE operator information pages

The changes to the JDE, 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 message:

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

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

- No END parameter is found in the DJDE.
- No complete page boundary was found before the end of the report, that is, some DJDEs 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 through a DJDE. The system does not allow these basic parameters to be changed using the JDE or JDL in the DJDE . The JDE specified by the DJDE JDE is referred to as a selected JDE.

Parameters that cannot be changed through a selected JDE or JDL are shown in table 1-7.

Table 1-7. **Parameters that cannot be changed through 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 1-8.

Table 1–8. **Parameters that can be changed through selected JDEs or JDLs**

Command	Parameter
ABNORMAL	SECURITY, ERROR
ACCT	All
IDR	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.

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 basis. The system determines whether 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 1-5, 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 and information, code the appropriate command, as shown in table 1-9.

Table 1–9. **Logical processing command usage**

To specify this	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 auxiliary tray for offline and online modes	RAUX
Record deletion for the offline and online modes	RDELETE
Specifies 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 and online modes	RRESUME
Record selection for the offline and online modes	RSELECT
End of report for the offline and online modes	RSTACK
Suspension of printing for the offline and online modes	RSUSPEND
One or more constants for logical processing	TABLE

Logical processing command format

To define a logical processing command fully, you must specify one or two fields in the record or block to be tested. In general, a logical processing command has the following format:

command TEST = *test-exp*;

The logical processing command tests the value of the specified *test-exp* and directs the flow of processing based on the result of the test.

The *test-exp* portion of the command defines a test to perform 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-*$_2$);

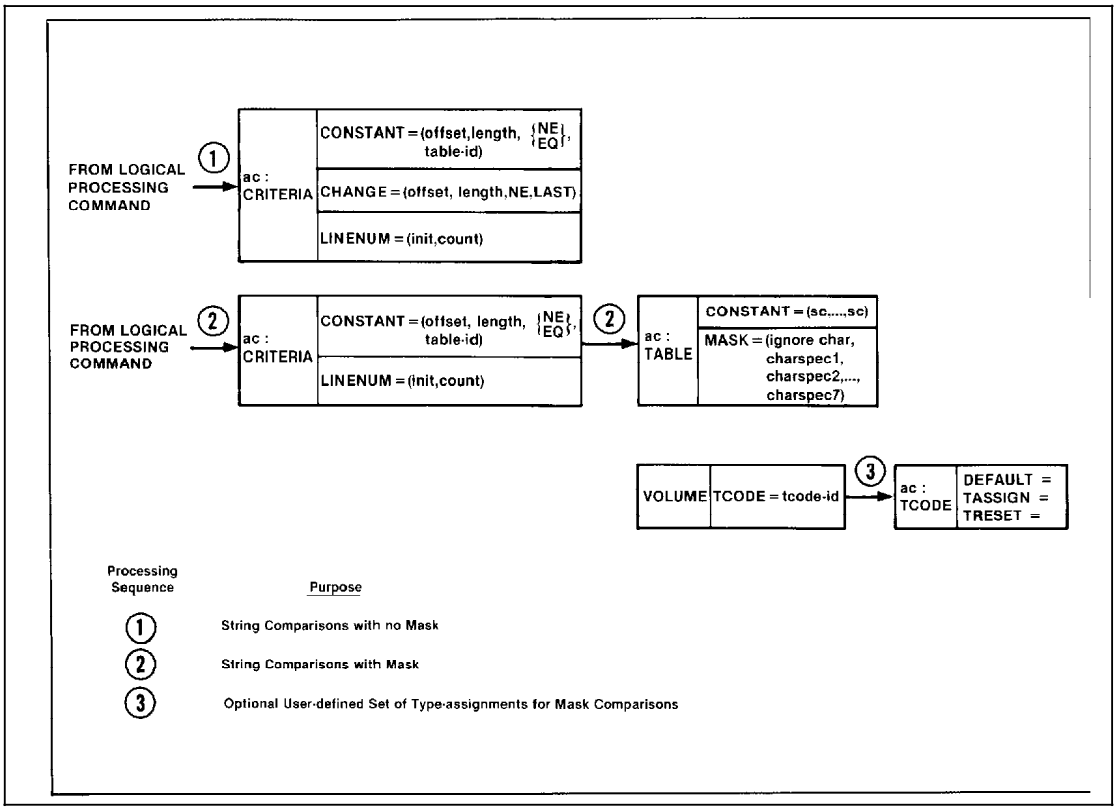
cri-id-$_1$ and *cri-id-*$_2$ are identifiers for either the change mode or the constant mode CRITERIA commands. If only cri-id-$_1$ is coded, the test is satisfied if the criterion in *cri-id-*$_1$ is satisfied. If *cri-id-*$_1$ and *cri-id-*$_2$ are both coded and the keyword AND is coded, the test is true only if 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.

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



These comparisons test for one of the following conditions:

- Absolute equality—Tested by either a change mode CRITERIA command or by a constant mode CRITERIA command, which references a TABLE command that does not have a MASK 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 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).

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 that 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 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 TRAY parameter is used to get the compiled listing of your code delivered to the sample tray on the printing system. Use of this parameter is optional.

Refer to your *Xerox 4850/4890 HighLight Color LPS Command Reference* for a complete list of available PDL command options.

JSLs created on a 4850 or 4890 LPS with version 4.0 (V4.0) software cannot be compiled on printing systems with earlier versions of software. JDLs compiled on earlier software versions will run on V4.0.

Printing a job

Once your JSl compiles without error, you can print your report by entering the START command at the operator console on your 4850 or 4890.

The START command calls in your compiled job descriptor library (JDL) and the job descriptor entry (JDE) within the library 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 1-4, 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 the operator guide for your 4850 or 4890 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.

BLOCK command

Defines block length characteristics and recording modes for offline data sources. The input processor decodes and formats input data from an offline magnetic tape, a host-attached channel interface, a remote communication, or an Ethernet interface.

ADJUST

Specifies a block length adjustment value to add to or subtract from the contents of the block length field to determine the true block length.

Syntax BLOCK ADJUST = *value*

Options *value*

Range for a *value* is -127 to 127, and must be less than the block length parameter (LENGTH). The plus (+) sign or minus (-) sign may be used to specify a positive or negative adjustment.

The default is 0.

Considerations Refer to the LENGTH parameter for more information.

With earlier versions of software, if a block containing fixed-length records was not a multiple of the record length (after taking into account any ADJUST parameter value), the following message would be displayed:

```
OS6550 Data not formatted as specified; space to next report
```

With V5.0 software, input processing may not detect this condition. V5.0 input processing is more lenient and prints the records, treating the last record of the block as a short record.

CONSTANT

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

Syntax BLOCK CONSTANT = *sc*

Options *sc*

Length of the constant may be 1 to 4 bytes.

There is no default.

Considerations If a block delimiter constant is positioned and is part of a record or block, use caution since the record or block will be truncated. As a result, the data is not formatted as you specified.

FORMAT

Specifies the recording mode of the block length field.

Syntax BLOCK FORMAT = *options*

Options

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.

Considerations The values for FORMAT may be overridden if RECORD STRUCTURE is changed through ANSI, IBM/OS Standard, or Honeywell 2000 COBOL label processing.

LENGTH

Specifies the longest physical block processed.

Syntax BLOCK LENGTH = *value*

Options

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

The default is 1330 bytes.

Considerations The maximum block size that can be processed depends 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). The LENGTH parameter can be overridden by ANSI, IBM/OS Standard, or Honeywell 2000 COBOL labels that specify block length. 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.

Earlier versions of software implemented a minimum BLOCK LENGTH of 368 bytes. This means that if a block length of less than 368 was coded in a JSL, blocks encountered in the data stream greater than the coded BLOCK LENGTH, but less than or equal to 368 bytes would be processed without error. With version 5.0 software, the maximum BLOCK LENGTH coded in the JSL is strictly enforced, and any blocks that are larger than the coded BLOCK LENGTH cause the following error message:

```
Input block length exceeds JDE max, do the following *Abort
and retry, specifying another JDE/JDL.
```

In this situation, the operator should then abort the job and restart it specifying an appropriate JDE or JDL.

LMULT

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

Syntax BLOCK LMULT = *value*

Options *value*

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.

The default is 1.

Considerations Refer to the LENGTH parameter for more information.

LTHFLD

Specifies the length of the field containing the block length.

Syntax BLOCK LTHFLD = *size*

Options *size*

An integer from 0 to 5 that specifies in bytes the length field you specify using the LENGTH parameter.

The default is 0.

Considerations If you set *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*

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.

The default is 0.

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 = *length*

Options *length*

Specifies an integer from 0 to the block length.

The default is 0.

Considerations

Do not code the length of the block delimiter constant as the BLOCK POSTAMBLE as 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 = *length*

Options

length

Specifies an integer from 0 to the block length.

The default is 0.

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 = *options*

Options

NO

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

YES

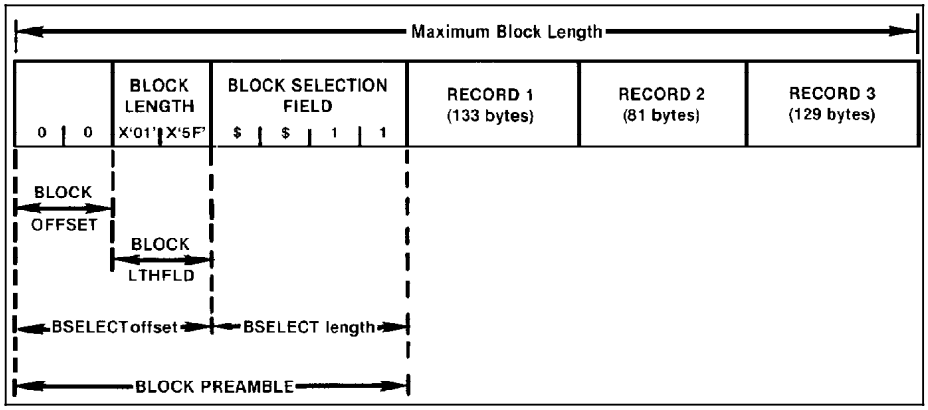
Indicates that the value for the end of a tape block is zero in the record length field before applying the record length adjustment. Data that follows the record is ignored through the end of the block.

Example

The BLOCK command in figure 2-1 is coded as follows:

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

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



ac:CODE command

Defines the input code of the translation table.

When you require a user-defined code translation table, reference the CODE command by the CODE or LCODE parameters of the VOLUME command. The USER option of the CODE and LCODE parameters of the VOLUME command 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). One of the characters must be a letter.

The default is EBCDIC.

ASSIGN

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

Syntax *ac*:CODE ASSIGN = *options*

Options *(input,output)*

This option has the following components:

input
Defines the input code.

output
Defines the output code, such as hexadecimal or decimal, that corresponds to the input code.

(*input*,(*output*$_1$,[*output*$_2$][,...]))

This option has the following components:

input
Defines the input code.

output
Defines an output code, such as hexadecimal or decimal, that corresponds to the input code.

DEFAULT

Specifies a base code from which you can make your code assignment exceptions. The base code is specified by a *code-type*. Your exceptions are specified with the ASSIGN parameter.

Syntax *ac*:CODE DEFAULT = *code-type*

Options *code-type*

Specifies one of the following options: ASCII, BCD, EBCDIC, PEBCDIC, H2BCD, H6BCD, IBMBCD, and *value.* A *value* is a one-byte hexadecimal, octal, or alpha constant. If you specify *value,* all inputs are coded to that specified value.

Considerations You must code the DEFAULT parameter prior to any ASSIGN parameters for the assignment exceptions to go into effect. A DEFAULT parameter following any defined corresponding ASSIGN parameter options causes this correspondence to be replaced by the DEFAULT parameter.

Examples

Example 1 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:

```
CODE1: CODE    DEFAULT = EBCDIC, ASSIGN = (X'5B', X'40'),
               ASSIGN = (X'5C', X'40'), ASSIGN = (X'5D',
               X'40');
```

Example 2 In Example 2, consecutive input codes, such as X'5C' and X'5D' need not be specified to accomplish code modification.

```
CODE2: CODE    DEFAULT = EBCDIC,
               ASSIGN = (X'5B', (X'40', X'40', X'40'));
```

IDEN command

You use this command to invoke 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 (that 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 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.

Syntax IDEN OFFSET = *value*

Options *value*

Specifies the number of bytes (beginning at 0) from the beginning of the user portion of the record to the beginning of the prefix string constant of the DJDE record. It may be a negative number.

OPRINFO

Specifies whether to print and deliver the DJDE record to the tray.

Syntax IDEN OPRINFO = *options*

Options NO

Specifies that the DJDE record does not print.

YES

Specifies that the DJDE record does print and is delivered to the output tray with the data sheets.

When OPRINFO=YES is selected, the length of the DJDE record that prints on the sheet is the amount that fits on the width of the paper. With portrait orientation, using OPRINFO=YES may cause page setup errors if the DJDE record is too long.

Considerations All non-Interpress system-generated pages, such as the 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 that 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.

Syntax IDEN PREFIX = *sc*

Options *sc*

Specifies the search criterion for recognizing DJDE records. It is a byte string of up to 255 characters represented as a hexadecimal, octal, BCD, ASCII, or EBCDIC character constant. Any records within the data stream containing an identification field equal to the specified prefix string *sc* are recognized and processed as DJDE records.

Considerations

The default character set is EBCDIC. Character strings must be preceded by a letter and a single quote as in the following examples:

```
EBCDIC:       IDEN    PREFIX = E'DJDE';
          or
              IDEN    PREFIX = 'DJDE';
          (EBCDIC is the default, so the E is optional)
ASCII:        IDEN    PREFIX = A'DJDE';
Octal:        IDEN    PREFIX = O'DJDE';
Hexadecimal:  IDEN    PREFIX = X'DJDE';
```

Refer to the "String constants" section in the "Overview" chapter for additional information.

SKIP

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

Syntax

IDEN SKIP = *value*

Options

value

Specifies the offset to the starting column of the DJDE parameters. A *value* specifies the number of bytes (beginning at 0) from the beginning of the user portion of the record to the beginning of the DJDE parameters. It may be a negative number.

ac:PCC command

Creates a table or modifies an existing standard table of one-byte printer carriage control codes and defines their actions. 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). One of the characters must be an alphabetic character.

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 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 advances 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.

Syntax

ac:PCC ADVTAPE = *options*

Options

YES
Specifies that multiple skips are allowed.

NO
Specifies that multiple skips result in only one skip action by the carriage control.

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 = *options*

Options

(byte,ccln)
This option has the following components:

byte
Defines a printer carriage control byte from 0 to 255 (X'00' to X'FF').

ccln
Specifies the action to take when the printer carriage control *byte* is encountered. A *ccln* may be one of the following options:

TOF
When *byte* causes the bottom-of-form (BOF) to occur, instructs the printing system to go to the top-of-form (TOF) on the next page and to stop spacing.

IGN
When *byte* causes the bottom-of-form (BOF) to occur, instructs the printing system to ignore it to continue spacing through the end of the physical page. The page then transitions to top-of-form and spacing continues.

OVR, *space-print parameters*
This option has the following components:

OVR
When *byte* causes the bottom-of-form (BOF) to occur, instructs the printing system to go to the top-of-form (TOF) on the next page and to continue spacing.

space-print parameters
This option has the following components:

[*field*$_1$][*field*$_2$][*field*$_3$]
Each of the three fields is optional; however, you must specify at least one field. The range for the variables *m* and *n* is 0 to 15.

field$_1$
Specifies the carriage action to take before printing. An example is the ANSI carriage control.

SP*m* Space *m* lines before printing.
SK*n* Skip-to-channel-*n* before printing.

field$_2$
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).

field$_3$
Specifies the carriage action to take after printing. An example is the IBM1403 carriage control).

SP*m* Space *m* lines after printing.
SK*n* Skip-to-channel-*n* after printing.

(*byte*,(*ccln*$_1$,[*ccln*$_2$][,...])
This option has the following components:

byte
Refer to the *byte* option description above.

ccln
Refer to the *ccln* option description above.

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

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

as a the single command:

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

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

DEFAULT

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.

Syntax *ac*:PCC DEFAULT= *options*

Options *ccln*
Specifies the required action when a code has not been

specifically assigned. The assignment codes for various actions are described in the *ccln* option description of the ASSIGN parameter. The system default PCC tables are defined in LPS-standard EBCDIC.

The default is a table of PSP1 codes (print-and-space-1).

pcctype
Specifies the following parameters and tables to modify: ANSI, B2500, B2700, B3500, B3700, B4700, B6700, H2000, H6000, IBM1401, IBM1403, US70, XEROX, and NONE.

Considerations 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.

Syntax *ac*:PCC INITIAL = *options*

Options TOF
Specifies the control program to perform the first spacing, skipping, or printing action from the top of the form.

BOF
Specifies the control program to perform the first spacing, skipping, or printing action form the bottom of the form.

MASK

Renders inaccessible unnecessary bits from the carriage control byte.

Syntax *ac*:PCC MASK = *value*

Options *value*
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.

The default is X'FF'.

Example

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:

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

RECORD command

Defines the characteristics of the offline input data record.

ADJUST

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

Syntax RECORD ADJUST= *value*

Options *value*

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. The *value* must be no greater than the record length field. The first character may be plus (+) or minus (-).

The default is 0.

CONSTANT

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

Syntax RECORD CONSTANT = *sc*

Options *sc*

Specifies a hexadecimal, octal, or alpha constant. The length of the constant may be from 1 to 4 bytes.

There is no default.

Considerations If the STRUCTURE parameter changes to U in label processing, CONSTANT 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.

Syntax RECORD FORMAT = *type*

Options *type*

Specifies one of the following options: BIN (binary), DEC (decimal), PACK (packed with no sign), or PKSG (packed with sign).

LENGTH

Specifies a length of the longest logical record.

Syntax RECORD LENGTH = *value*

Options *value*

Specifies in bytes the length of the longest logical record. The maximum *value* for online systems without print position indexing is 214 (optimize mode) or 2140 (non-optimize).

The *value* is an integer from 1 to 310 for offline systems, with 133 as the default.

Considerations The tape label contents may override a record length value. If you define a record length larger than the default block length (1330), enter a block length large enough for 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 greater than 150.

LMULT

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

Syntax RECORD LMULT = *value*

Options *value*

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 (default) to 15.

LTHFLD

Specifies the length of the field containing the record length.

Syntax RECORD LTHFLD = *value*

Options *value*

Specifies the record length field length in bytes from 0 (default) 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.

OFFSET

Specifies location of the record length field.

Syntax RECORD OFFSET = *value*

Options *value*

The location of the record length field offset, measured from the first byte of the record to the record length field (valid range from 0 (default) to LENGTH-LTHFLD-1).

POSTAMBLE

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

Syntax RECORD POSTAMBLE = *length*

Options *length*
Specifies the length in bytes. The *length* is an integer from 0 (default) to record length.

PREAMBLE

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

Syntax RECORD PREAMBLE = *length*

Options *length*
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 (the default) to record length.

STRUCTURE

Specifies the general structure of the input data.

Syntax RECORD STRUCTURE = *structure-type*

Options *structure-type*
Specifies one of the following options: FB (fixed length blocked), F (fixed length), V (variable length), VB (variable length blocked), U (undefined length), or UB (undefined length blocked). For offline processing, the tape label contents may override this parameter.

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.

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**

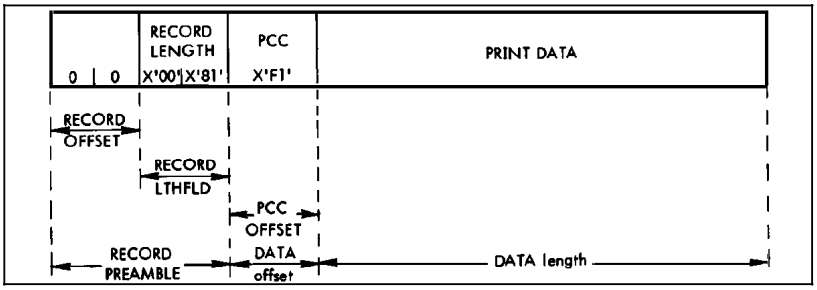
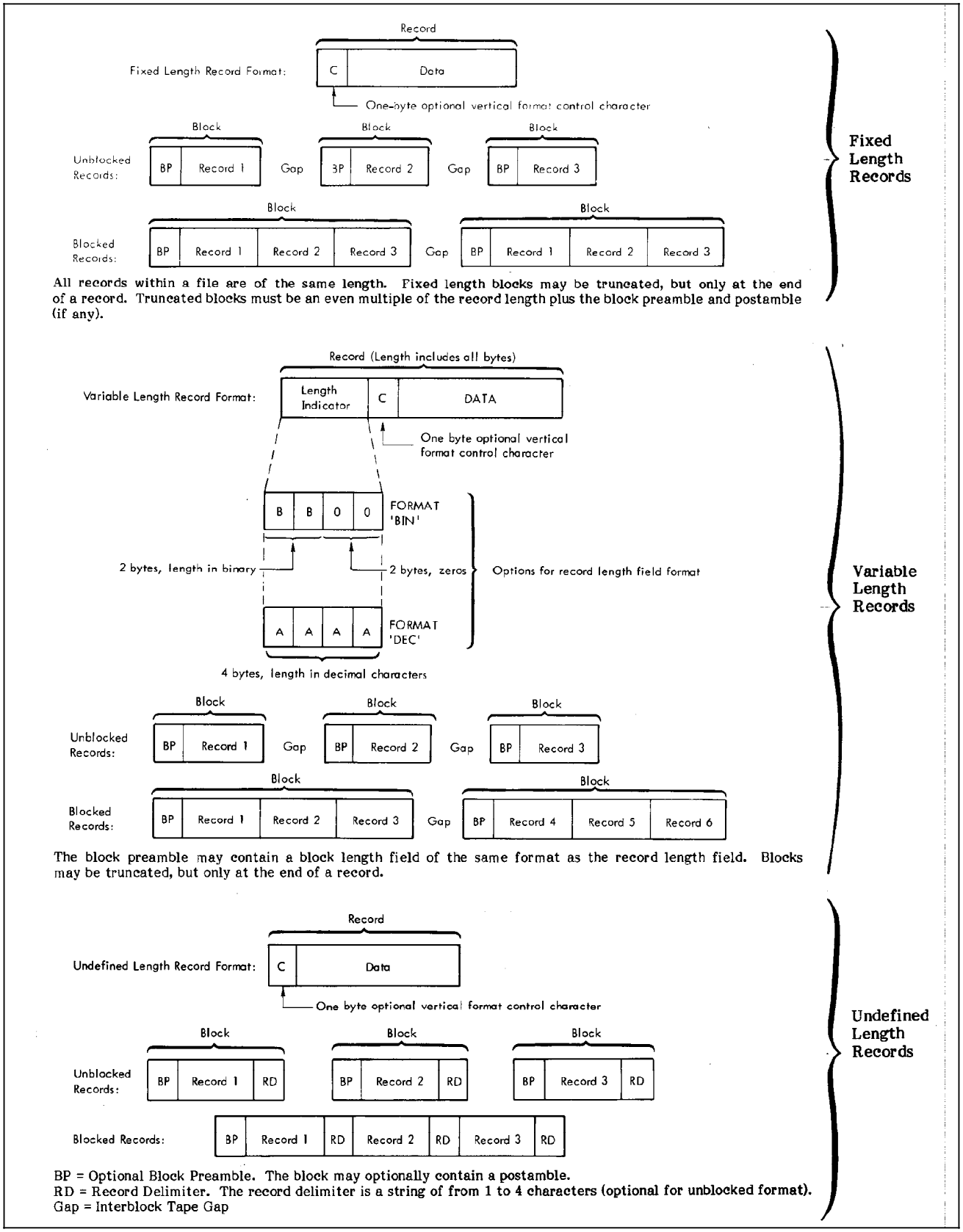


Figure 2-3. **Sample input record**



ac:TCODE command

Defines a set of input-character-to-type assignments for the masked string comparisons in a report. You must use the TCODE command 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, modify further 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 as hexadecimal 41, and the letter B as 42 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 console can process and display lowercase characters literally rather than symbolically.

Table 2-1 shows how to 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 that 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 *tcode-type* keyword.

Syntax *ac*:TCODE DEFAULT = *tcode-type*

Options *tcode-type*

Specifies one of the following options: ASCII, BCD, EBCDIC, PEBCDIC, H2BCD, H6BCD, IBMBCD, and *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.

Set initial type codes to allow for the following conditions:

- All numeric characters (0 through 9) are type 1 characters
- All alphabetic characters (A through Z, a through z) are type 2 characters
- All other characters are untyped.

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

Syntax *ac*:TCODE TASSIGN= *options*

Options (*typespec,'inputspec')*

This option has the following components:

typespec

Specifies a single integer *n* or series of integers from 1 through 7.

When a *typespec* is coded as a single integer *n,* all specified characters are associated with the type code.

When a *typespec* is coded as a series of integers, the specified characters are each associated with all of the specified types. A *typespec* may also be specified with the following keywords:

- NUMERIC instead of 1
- ALPHA instead of 2.

When a *typespec* is coded as 0, all characters specified by an *inputspec* are disassociated from all type code assignments. Thus, the TASSIGN parameter may untype a specified set of characters.

inputspec
Specifies a single character *char,* a series of characters *char...*, or a range of characters $char_1$,-,$char_2$. The syntax $char_1$,-,$char_2$ indicates the set of input codes corresponding to characters bounded by and including $char_1$ and $char_2$.

(typespec, (‘$inputspec_1$’,[‘$inputspec_2$’][,...])
This option has the following components:

typespec
Specifies a single integer *n* or series of integers from 1 through 7.

When a *typespec* is coded as a single integer *n,* all specified characters are associated with the type code.

When a *typespec* is coded as a series of integers, the specified characters are each associated with all of the specified types. A *typespec* may also be specified with the following keywords:

- NUMERIC instead of 1
- ALPHA instead of 2.

When a *typespec* is coded as 0, all characters specified by an *inputspec* are disassociated from all type code assignments. Thus, the TASSIGN parameter may untype a specified set of characters.

inputspec
Specifies a single character *char,* a series of characters *char...*, or a range of characters $char_1$,-,$char_2$. The syntax $char_1$,-,$char_2$ indicates the set of input codes corresponding to characters bounded by and including $char_1$ and $char_2$.

TRESET

Disassociates one or more characters in a set from one or more types.

Syntax *ac*:TCODE TRESET = *options*

Options (*typespec,* ALL*)*

This option has the following components:

typespec

Specifies a single integer *n* or series of integers from 1 through 7.

When a *typespec* is coded as a single integer *n,* all specified characters are associated with the type code.

When a *typespec* is coded as a series of integers, the specified characters are each associated with all of the specified types. A *typespec* may also be specified with the following keywords:

- NUMERIC instead of 1
- ALPHA instead of 2.

ALL

Specifies that all characters in the code table are disassociated from the specified types.

(*typespec,'inputspec')*

This option has the following components:

typespec

Specifies a single integer *n* or series of integers from 1 through 7.

When a *typespec* is coded as a single integer *n,* all specified characters are associated with the type code.

When a *typespec* is coded as a series of integers, the specified characters are each associated with all of the specified types. A *typespec* may also be specified with the following keywords:

- NUMERIC instead of 1
- ALPHA instead of 2.

inputspec

Specifies a single character *char,* a series of characters *char...*, or a range of characters $char_1$,-,$char_2$. The syntax $char_1$,-,$char_2$ indicates the set of input codes corresponding to characters bounded by and including $char_1$ and $char_2$.

(typespec, ('$inputspec_1$',['$inputspec_2$'][,...])

This option has the following components:

typespec

Specifies a single integer *n* or series of integers from 1 through 7.

When a *typespec* is coded as a single integer *n,* all specified characters are associated with the type code.

When a *typespec* is coded as a series of integers, the specified characters are each associated with all of the specified types. A *typespec* may also be specified with the following keywords:

- NUMERIC instead of 1
- ALPHA instead of 2.

inputspec
Specifies a single character *char,* a series of characters *char...*, or a range of characters $char_1$,-,$char_2$. The syntax $char_1$,-,$char_2$ indicates the set of input codes corresponding to characters bounded by and including $char_1$ and $char_2$.

Considerations You can specify the TRESET parameter more than once in the TCODE command.

VOLUME command

Specifies the characteristics of the input medium.

A JDL is supplied with the printing system 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, and 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.

Versions 3.7, 4.0, and 5.0 software can process Interpress jobs.

BMULT

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

Syntax VOLUME BMULT = *value*

Options *value*
Specifies a multiplication factor that you extract from the tape label and apply to the block length to determine the true block length. The *value* is an integer and can range from 1 (default) to 15.

CODE

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

Syntax VOLUME CODE = *options*

Options *keyword*
Specifies a specific, standard, system-defined code translation table. The available options are: ASCII, BCD, EBCDIC, H2BCD, H6BCD, IBMBCD, PEBCDIC, NONE, and USER. USER refers to the single unlabeled CODE command defined within the JDE or JDL. 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.

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.

NONE
Specifies that data is not to be translated by input processing.

Considerations

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

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

When you specify CODE = NONE for a highlight color job, an IDR command ILIST parameter containing a reference to an ink with a primary color causes the pages of the report to print in highlight mode. An ILIST parameter that references only black ink causes the pages of the report to print in black.

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

EOV

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

Syntax

VOLUME EOV = ([PAUSE I NOPAUSE], [EOF I NOEOF])

Options

PAUSE
Displays a message when EOV is encountered. A CONTINUE response by the operator causes the tape to rewind and the normal volume change sequence to proceed.

NOPAUSE
Issues a rewind instruction as soon as the EOV label is processed.

EOF
End-of-file. Specifies that the end-of-volume label is treated as an end-of-file label. When this occurs, the first part of the page spanning the volumes is output as the last page of the job. The second part of the spanned page is printed as the first page when the next volume is started, with possible page format irregularities. To use the EOF parameter effectively, special user-formatting of the multireel tapes is required to avoid these page format problems.

NOEOF
Specifies normal end-of-volume processing.

HOST

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

Syntax VOLUME HOST = *host-type*

Options *host-type*

Specifies one of the following options: IBMOS, ANSI, B2500, B2700, B3500, B3700, B4700, B6700, DEC, PDP11, DUMP, GRASP, H2000, H6000, IBMDOS, IBMONL, ICLS900, NCR, OCTDUMP, OLDUMP, OSWTR, POWER, POWERVS, RSX11, UNDEF, UNIVAC 1100, US70, XEROX, and ENET.

Considerations For online systems, you must specify IBMONL or OLDUMP.

For Ethernet, you must specify ENET.

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

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

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

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

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

INTERPRESS

Provides additional control over the Interpress environment.

Syntax VOLUME INTERPRESS = *options*

Options The following options are grouped in pairs and are mutually exclusive. For example, INTERPRESS =(ROBUST, SIMPLE) results in SIMPLE as the selection. The last of any mutually exclusive parameters is selected.

[YES],[CHECKSUM I NOCHECKSUM]
,[BREAKPAGE I NOBREAKPAGE]
,[INTEGRAL I FRACTIONAL]
,[PERFORMANCE I COMPLIANCE],[RIP I NORIP]
This option has the following components:

YES

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

CHECKSUM I NOCHECKSUM

This option has the following components:

CHECKSUM

Enables error-checking procedures on a job basis. If CHECKSUM is specified and verification fails, the job aborts.

NOCHECKSUM:

Disables that error-checking procedures.

BREAKPAGE | NOBREAKPAGE

This option has the following components:

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.

INTEGRAL | FRACTIONAL

This option has the following components:

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.

PERFORMANCE | COMPLIANCE

This option has the following components:

PERFORMANCE

Specifies, with the raster imager processor (RIP) that raster encoding standard (RES) files containing sampled color are imaged by the graphics hardware whenever possible. The RIP is used only when graphics hardware capabilities are exceeded.

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.

RIP | NORIP

This option has the following components:

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.

LABEL

Specifies the type of label on the input tape.

Syntax VOLUME LABEL = *label-type*

Options *label-type*

Specifies one of the following options: NONE (the input tape is unlabeled), ANSI, STANDARD, SPR (Honeywell 2000 System Print Tape), COBOL (Honeywell 2000 COBOL tape with 120-byte labels), and UNDEF (input tape label is not defined).

LCODE

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

Syntax VOLUME LCODE = *keyword*

Options *keyword*

Specifies a specific, standard, system-defined code translation table. The available options are: ASCII, BCD, EBCDIC, H2BCD, H6BCD, IBMBCD, PEBCDIC, NONE, and USER. USER refers to the single unlabeled CODE command. allowed within the JDE or JDL. You must specify the variable identifier *id* to correspond to the identifier label assigned to the defining CODE command.

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.

NONE

Specifies that data is not to be translated by input processing. For example, you select NONE for XPPI jobs that contain metacodes.

LPACK

Specifies whether undefined labels (LABEL=UNDEF) are packed or unpacked.

Syntax VOLUME LPACK = *options*

Options NO

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

YES

Indicates that the labels are packed.

MAXLAB

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

Syntax VOLUME MAXLAB = *value*

Options *value*

Specifies in bytes the maximum length of the longest physical block. The *value* can range from 2 to 4096 bytes. The specified *value* must be greater than the MINLAB *value* and must not exceed the BLOCK LENGTH.

The default is 81.

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

MINLAB

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

Syntax VOLUME MINLAB = *value*

Options *value*

Specifies in bytes the maximum length of the smallest physical block. The *value* can range from 1 to 4095 bytes. The specified *value* must be less than the MAXLAB value.

The default is 80.

OPTIMIZE

Enhances the performance of the data stream.

Syntax VOLUME OPTIMIZE = *options*

Options *keyword*

Specifies one of the following options: NCCHECK, NDCHECK, and NPRECOVER. The maximum performance occurs when all the parameters are selected, but there is still significant improvement when you specify only NCCHECK and NDCHECK.

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.

(*keyword*$_1$[,*keyword*$_2$][,*keyword*$_3$])
Allows you to specify more than one of the following options: NCCHECK, NDCHECK, and NPRECOVER. The maximum performance occurs when all the parameters are selected, but there is still significant improvement when you specify only NCCHECK and NDCHECK. Refer to the *keyword* option above for more information on NCCHECK, NDCHECK, and NPRECOVER.

When you specify both NCCHECK and NDCHECK, data stream performance is significantly enhanced because input processing of a record and the subsequent data transfer of the next record occurs immediately.

NONE
Does not invoke the optimization parameter.

OSCHN

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

Syntax VOLUME OSCHN = *value*

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. 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 which precede the report. The default is 0.

OSTLP

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

Syntax VOLUME OSTLP = *value*

Options *value*
Specifies the number of trailer pages which are to follow the report. The default is 0.

PLABEL

Specifies whether the tape labels are printed.

Syntax VOLUME PLABEL = *options*

Options 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.

RMULT

Specifies a multiplication factor the record length uses to determine the true record length.

Syntax VOLUME RMULT = *value*

Options *value*
An integer from 1 to 15.

RSAT

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

Syntax VOLUME RSAT = *options*

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

SPLIT
Indicates that only one pass is made over the input data and that the reports are in sections and therefore require manual merging. Input processing is faster for SPLIT since only one pass is required over the input data. SPLIT is recommended for graphics processing when using page interleaved or move mode, since moving a large amount of graphic data impacts data stream performance.

TCODE

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

Syntax VOLUME TCODE= *options*

Options *tcode-type*
Specifies one of the following standard character sets for which type assignments are defined by default: ASCII, BCD, EBCDIC, PEBCDIC, H2BCD, H6BCD, and IBMCD. By coding any one of the keywords, you are furnished with a set of character-to-type assignments.

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

UNPACK

Specifies the unpacking method of the input data when required.

Syntax VOLUME UNPACK = *options*

Options

NONE
Does not perform the unpacking operation.

T4X3 or T4X3H2
Specifies unpacking routines in which 6-bit characters are extracted and restored as 8-bit bytes. T4X3 is used primarily for Honeywell 6000 tapes while T4X3H2 is used for Honeywell 2000 tapes.

UNIVAC
Specifies an unpacking routine used for UNIVAC tapes, although you can specify any of these routines independent of the HOST type.

VCODE

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*)

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. The default value is VCODE0.

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

Examples

VOLUME CODE = ASCII, VCODE = (ASCII,VCODE0);
This 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 = EBCDIC, VCODE = (EBCDIC,VCODE1);
This command utilizes the character type code table applicable to France to process and compare numeric values through an EBCDIC code translation.

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

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

Examples

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

Example 1	VOLUME	HOST = UNIVAC, LABEL = STANDARD, UNPACK = T4X3, CODE = ASCII, LCODE = ASCII;
Example 2	VOLUME	HOST = IBMOS;
Example 3	VOLUME	HOST = IBMONL, LABEL = NONE, INTERPRESS = YES, CODE = EBCDIC;
Example 4	VOLUME	HOST = B6700, LABEL = ANSI, BMULT = 6, RMULT = 6, PLABEL = YES;
Example 5	VOLUME	HOST = H2000, LABEL = SPR, CODE = H2BCD, UNPACK = T4X3H2;
Example 6	VOLUME	HOST = OCTDUMP, CODE = H6BCD, UNPACK = T4X3;

3. Print format commands

Print format commands serve many different functions, the most common of which specify the physical characteristics of a print job, define the placement of data on the page, and identify system responses to error conditions.

ABNORMAL command

Allows you to restrict certain operator functions, define system responses to error or ink mismatch conditions.

ERROR

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

Syntax ABNORMAL ERROR = *options*

Options

STOP
Stops processing the report. Waits for you to respond with CONTINUE or ABORT. The displayed error message 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.

Considerations 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:

- Nonstop 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.

IMISMATCH

Indicates the action to take when a specified ink requires a primary that is not currently loaded.

An ink mismatch occurs when a report requests an ink that requires a primary color not currently loaded. It can occur within a report and may be detected after printing has started.

If the creator of a report did not include an ink list in the JDE for a report, printing may start before the system is aware of the primary colors needed. For example, page 100 of a report, which was designated black-only, calls for the green primary and the ink currently loaded in the printer is red.

Syntax ABNORMAL IMISMATCH = *options*

Options

STOP

Prompts the operator to load the requested primary or abort the report or job. If ISUBSTITUTE = ANY is specified for the report, the operator may also enter a SUBSTITUTE INK command followed by the CONTINUE command to resume printing.

The appropriate action for an operator to take for an ink mismatch condition when IMISMATCH = STOP is specified depends on a number of factors:

- If the report has not been sent for output processing yet, the operator can enter ABORT I to abort report input processing. However, if the operator then enters the CONTINUE I command, the report prints with black ink in place of the missing ink. If this is not the desired result, the operator should enter ABORT I and ABORT O, followed by the CONTINUE command.
- If the report is already being processed for printing, the system responds to the mismatch condition by displaying the following messages:

```
OS2513  x HOUSING IS NEEDED FOR THIS REPORT
OS2682  CHANGE HOUSING FOR PRINTING IN REQUESTED INK OR
OS2675  ABORT THE REPORT OR PRINT JOB
OS2000  ENTER ‘CONTINUE O’ TO RESUME PRINTING
```

  When this series of messages appears, the operator has three choices:

  – Change the color housing and enter CONTINUE O

  – Abort the report and enter CONTINUE O. If the operator elects to abort the report, there are two possible methods:

    - Enter ABORT O

    - Enter CANCEL ENTRY *#entry-string*, where *entry-string* is the number or numbers of the reports in the queue to be cancelled.

  – Abort the entire job that contains the report by entering ABORT *job-id* and CONTINUE O. Refer to your *Xerox 4850/4890 HighLight Color LPS Command Reference* for information on these commands.

ABORT

Printing of the report is aborted.

CONTINUE
Uses the currently loaded primary regardless of the ISUBSTITUTE parameter or the FDL INKS command NOSUBSTITUTION parameter.

Example:

```
ABNORMAL   IMISMATCH = CONTINUE;
IDR        ILIST = ('BLACK', 'BLUE');
```

In this example, if 'RED' is currently loaded and ready, the report is printed in red with no operator intervention.

Considerations

Ink mismatch errors are not handled by the ERROR parameter. Ink mismatch is a special type of error that requires specific operations and is processed independently of ERROR parameter specifications.

Refer to the "Coding for efficient queue management" section in the "Highlight color printing" chapter for information on the relationship between queue management and the IMISMATCH parameter.

ISUBSTITUTE

Specifies whether operator-initiated ink substitution is allowed.

Syntax

ABNORMAL ISUBSTITUTE = *options*

Options

ANY
Any operator-initiated substitution is allowed.

NONE
No operator-initiated substitution is allowed. The report is either printed without operator-initiated substitution or aborted by the operator.

Example: ABNORMAL ISUBSTITUTE=NONE;

Considerations

Refer to the "Efficient coding for queue management" section in the "Highlight color printing" chapter for information on the relationship between queue management and the ISUBSTITUTE parameter.

If the ISUBSTITUTE parameter specifies NONE, the printer cycles down and displays messages on the system controller screen. These messages indicate that a print quality adjustment may be performed before continuing to print the job or report with the requested primary, or that the job or report is being aborted.

Table 3-1 lists what happens if the housing is inoperable or not present.

Table 3-1. **Results of inoperable or absent housings**

IMISMATCH parameter option	ISUBSTITUTE parameter option	Does the system controller cycle down?
CONTINUE	--	NO
ABORT	--	NO
STOP	ANY	YES
STOP	NONE	YES

OTEXT

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

Syntax ABNORMAL OTEXT = *options*

Options

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.

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 is printed.

Figure 3-1. **Sample marker page**

```
*****************************
*RECOVERY MARKER PAGE RECOVERY MARKER*
*E                                  E*
*C                                  C*
*O   CRASHING TASK:   INPUT         O*
*V                                  V*
*E   CRASH CODE:   BPT              E*
*R                                  R*
*Y   DATE:  26 APR 85, 18:25        Y*
*Y                                  Y*
*M   JOB NUMBER:   2565             M*
*A                                  A*
*R   REPORT NUMBER:   1             R*
*K                                  K*
*E                                  E*
*RECOVERY MARKER PAGE RECOVERY MARKER*
*****************************
```

SECURITY

Specifies that certain operator functions are restricted.

Syntax ABNORMAL SECURITY = *options*

Options

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).

Considerations If SECURITY is invoked, 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 command

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 request that the LPS 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

ACCT DEPT = *sc*

Options

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 character ":" is also allowed for the department code. You must create the *sc* option name in the accounting file before you process a job.

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.

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.

The following example illustrates the use of the PDL command ACCT and the system level ACCOUNT parameter. When the system level parameter ACCOUNT ADD, OPERATIONS is entered at the keyboard display, the name OPERATIONS is added to the current list of names in the system accounting file.

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.

Restricting LPS usage to only those departments explicitly added to the accounting file is accomplished by deleting the NODEPT entry. If NODEPT is deleted, jobs are aborted unless a name has been entered in the accounting file that corresponds with the department name as specified on the ACCT command. The NODEPT entry must be deleted before any accounting data has been 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

ACCT USER = *options*

Options

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.

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 the user 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 by the user 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.

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 command

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 keyword, omit equal signs, and omit commas except where needed to avoid ambiquity.

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 OUPUT command MODIFY parameter.

CONSTANT

Specifies the character string to print.

Syntax *ac:*CME CONSTANT = *sc*

Options *sc*

Represents the string constant to print. 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.

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. 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 lowercase until another # is encountered. The sequence ## inserts the character #. This sequence does not toggle lowercase mode.

Considerations

A font switch is applied to a CME string constant based on 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, otherwise, the font switch occurs at the line position immediately after the string constant.

An ink switch is applied to a CME string constant based on 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, otherwise, 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*

Options

value

Range 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.

The default for *value* 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.

INK

Specifies the ink to be used to print variable data affected by CME or CME constant data.

Syntax *ac:*CME INK = *inkindex*

Options *inkindex*

The value that represents the *inkname* to use to print the CME constant or variable data affected by CME data. In the short form CME specifications, a comma is required following the *inkindex* option unless it is the last item in the command. It must relate to a predefined IDR command ILIST parameter. The *inkindex* is always a value from 1 to *n*, where *n* is the number of inks in the current ink list. An ink list is specified by the ILIST parameter of the IDR command or the ILIST parameter of the DJDE.

An *inkindex* option operates very much like a *fontindex* option, in that inkindex specifies an index into the current ink list. Once inks and fonts are changed they remain in effect until they are explicitly changed again or until the report copy ends.

The default is 0, which instructs the system to use the original ink, the ink in effect for the record prior to modification by the CME.

The following is an example of using the default 0:

```
CME1:CME LINE=(5,60),POS=10,POS=20, INK=0;
/*the original ink is printed starting with line 5 and ending
with line 64*/;
CME2:CME (L1I2F2, L5F1, L6I0F3);
/*report title on line 1 is printed in font 2 and ink 2
(both font and ink changed)*/
/*column headings on line 5 are printed in ink 2 with the
font1 (same as before CME)*/
/*column subheadings on line 6 are printed in font 3 with
ink unchanged (same as before CME)*/
```

By specifying CME2:, you can print variable data on a portion of a form in an ink other than the original ink.

Considerations JSLs which define CMEs that use the INKS parameter must be recompiled after loading the 4850 or 4890 enabler software. If INK=0 is specified, the ink for the variable data and CME text will revert back to the IDFAULT ink. If an ink index value is not specified, the IDFAULT ink is used to print variable data and CME text.

LINE

Specifies the line range of the CME.

Syntax *ac:*CME LINE = *options*

Options *n*

The initial line number of the copy modification rectangle.

(*n,m*)

This option has the following components:

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.

(*n, -*)

This option has the following components:

n

The initial line number of the copy modification rectangle.

-

A dash character indicates that information is to apply to all lines on a page beginning with the line indicated by the N options.

Considerations The 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,FON
T=2,
LINE=(31,50),POS=1,FONT=2,POS=40,FONT=3,POS=80,FON
T=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.

Syntax *ac:*CME POSITION = *n*

Options *n*

An integer value between 1 (the first position of the print line) and the value specified by the length right hand part of the DATA = parameter in the LINE command.

The default is 1.

Considerations The LINE parameter must always precede the POSITION parameter.

EXPORT command

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:

SNUMBER 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.

SPLIT=OFF 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 reenable segment

management processing after it has been disabled by the DJDE SPLIT=OFF command.

End of segment

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.

Resetting the segment number

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

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.

Segment management and disk saturation

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 = {FIRST|LAST|(LAST,D)|BOTH|(BOTH,D)|<u>NONE</u>}

Options

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.

<u>NONE</u>
: 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.

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 having the following format:

PART *nnn* of *nnn*

Syntax

EXPORT SNUMBER = (*lnum*, *cnum* [, *findex*])

Options

lnum
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 you want to use 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*)

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 EXPORT SRECOVER = (PAGE|SEGMENT|ASK|DEVICE)

Options

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.

STIMING

Provided for possible compatibility with other Xerox software supporting third party finishing devices and is ignored by the Xerox 4850/4890 LPS. Device specific timing parameters are actually set by using the Finishing Configuration (FCG) Utility.

The STIMING parameter, although ignored by the Xerox 4850/4890 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 STIMING = ([INTERVAL,*sec*][,DELAY,*sec*])

Options

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 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.

ac:IDR command

Allows you to define default ink catalogs, palettes, and inks.

When a JSL uses an IDR command and the IDR is not in the JSL, the PDL compiler provides the following warning message:

```
IDR FILE WILL BE USED DURING PRINTING.
```

An identifier is specified when defining the IDR and may be referenced with the IDR parameter of the OUTPUT command or DJDE.

IDR may be coded as part of the job descriptor library (JDL) or created as separate files such that they may be referenced by one or more JDLs or by DJDEs.

An IDR created as a separate disk file may be used as if it was part of the JDL that references it. A separate file is created when IDR commands outside the body of a JDL are processed by the PDL processor. For each such IDR command, PDL creates a file on disk, cataloged in the IDR directory. The IDR command identifier is used as the filename. When the IDR is referenced, the system searches the IDR directory for the named IDR and, if found, loads it into memory for use in processing the report.

If a DJDE is to call the IDR, it is required to define the IDR prior to the JDL statement so that the IDR is saved on the system disk as a separate file.

The parameters specify the default catalogs, palette and ink lists to be used for the job. The IDR parameter of the OUTPUT command references an IDR command (with appropriate parameters) for each job, and these are used for the job unless they are overridden by other commands or command parameters. For example, the IDFAULT parameter of the OUTPUT command can override an ink specified in an IDR that has been referenced in a job. The default ink is defined according to the following sequence of overrides:

- IDFAULT DJDE has highest priority.
- IDFAULT parameter of the OUTPUT command has the next priority at the job level.
- A catalog-level default overrides a system-level default if specified at the job level.
- A system-level default takes precedence if not overridden at the catalog or job levels.
- First element of an ILIST parameter.
- If no IDFAULT is specified, the first element of the ILIST is used as the default.
- If there is no ILIST specified, the operating system software (OSS) default is used.

ICATALOG

Establishes a default ink catalog to be used in ink references in the absence of a specific catalog reference at a lower JSL.

Syntax *ac*:IDR ICATALOG = [DFAULT|XEROX|XEROX1]

Options DFAULT|XEROX|XEROX1

Selection remains in effect until changed by another IDR ICATALOG parameter. This also applies to ink references in variable data through DJDEs.

The default is the system DFAULT ink catalog.

ILIST

Provides a list of all the inks which may be referenced by indexing in printing variable input data or CME data.

Syntax *ac*:IDR ILIST = *options*

Options *'ink-name'*

Provides an ink (solid primary colors only) which may be referenced by indexing it in printing input or variable data.

('*ink name*$_1$'['*ink name*$_2$'][,...])

Provides a list of all inks (solid primary colors only) which may be referenced by indexing inks in printing input or variable data. The first ink in the ILIST parameter is the default unless overridden by an IDFAULT parameter. The maximum number of inknames which may be specified in the ILIST parameter is 64.

Considerations If an IDR parameter is not specified in the OUTPUT command or if the IDR specified does not exist, PDL supplies DFIDR.IDR as the system default IDR. DFIDR.IDR is a standard file on the system software tape (SST), which contains the definition of the system default black ink specified by the following parameter line:

DFIDR: IDR ILIST='BLACK';

Notice that the ICATALOG and PALETTE parameters are not specified and *inkcatalog-name* and *palette* are not specified in the *inkname*. Only the ILIST parameter is required. Since an ink catalog and palette are not specified, the system default ink catalog and palette are used. The system default ink catalog and palette are DFAULT.ICT and 'DFAULT', respectively. Furthermore, the system default IDR file may be modified to specify a different ink catalog, palette, and ink list.

When the VOLUME command CODE=NONE parameter is specified, an ILIST parameter referencing a highlight color primary causes the entire report to print in highlight mode. If ILIST references only black ink, the report prints in black mode.

Refer to the "Efficient coding for queue management" section in the "Highlight color printing" chapter for information on the relationship between queue management and the ILIST parameter.

PALETTE

Establishes a default palette used in ink references in the absence of a specific palette reference.

Syntax *ac*:IDR PALETTE = '*palette*'

Options '*palette*'
Establishes a default palette to use in ink references in the absence of a specific palette reference. The parameter remains in effect until changed by another IDR PALETTE parameter. This also applies to ink references in variable data through DJDEs.

LINE command

Allows you to control margin, 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)*

Options *(pdo, length)*
This option has the following components:

pdo
Print data offset. This is 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. For offline systems, the default is (1,132). For online systems, the default is (0,150). For HOST=RSX11(FLX), the DATA parameter must be included in the user's JSL since the default *pdo* value causes the first data byte to be skipped.

Considerations When changing the print line data to a value greater than the default value (150), the LENGTH parameter of the RECORD command must also be changed accordingly.

FCB

Suppresses or accepts FCB processing.

Syntax LINE FCB = *options*

Options 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 forms control buffer (FCB).

FONTINDEX

Specifies that a field within a user data record defines the index to a specific font to be used for that line. If FONTINDEX is not specified, none is used.

Syntax LINE FONTINDEX = *options*

Options

NONE
Specifies that there is no font index.

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

(*offset*,[*init-val*][,*bit-opt*])
This option has the following components:

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

init-val
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. 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.

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

Characters in the previous record which are blank are replaced only when they are represented by an EBCDIC X’40’ or an ASCII X’20’. A binary character representations 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 in use in the previous line. The current line can also be adjusted downward by the difference in height between the first and the largest font in the current line.

The line spacing of the first line in a report is determined by the first font in the PDE, provided the FONTINDEX parameter has not been specified. When FONTINDEX is specified, the line spacing of the first line of a report is determined by the font indicated by the font index value in the first line of the report. If there is no font index value in the record and FONTINDEX is specified, the line spacing for the first line of the report is determined by the first font in the PDE.

When the FONTINDEX or CME parameters are specified, the line spacing of the first line of subsequent pages is determined by the line spacing of the font called out by the last font index of the previous page into the current PDL font list. If a PDE DJDE has occurred prior to the page transition, the font used for the line spacing comes from this list. The height of the last font used determines the initial base line of that first print line.

The FONTINDEX byte, if selected, is processed for every record, DJDEs, RPAGE, and all criteria records. A valid FONTINDEX byte should be present in all records since it controls line spacing and overrides line spacing at page transitions.

It is advisable to put the font index byte in the beginning of a variable record to conserve tape space. The byte must be present in every record.

INKINDEX

Specifies a field within a user data record that contains a number representing the index in the current ILIST.

The printer permits an entire line or a portion of a line to be printed in highlight color. You can specify highlight color for variable data on a form. This is analogous to the existing output capability of font switching.

Variable data is printed using the ink specified in the INKINDEX byte. If the INKINDEX byte is blank or does not exist in the variable data record, then the variable data is printed using the default ink.

Syntax LINE INKINDEX = *options*

Options NONE
Specifies that there is no ink index.

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

(*offset*,([*init-val*],[*bit-opt*]))
This option has the following components:

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

init-val
Specifies an index value associated with the first ink in the ink list. It can be ZERO or ONE.

ZERO
The first ink in the ink list is to be considered index zero.

ONE
The first ink in the ink list is to be considered index one, the system default.

bit-opt
Specifies how many bits to use in calculating the ink index. Allowable values for bit-opt are 1-7. The default is 4.

Considerations Example:

```
LINE    DATA=(1,132), FONTINDEX=133, INKINDEX = 143;
```

MARGIN

Specifies the left margin on a physical page.

Syntax LINE MARGIN = *options*

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. The default is 1.

(*value,value-type*)
This option has the following components:

value
The form *nnn.mm* (a positive decimal number with up to 2 digits to the right of the decimal point) which is the distance from the left margin. The default is 1.

value-type
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 a *value-type* is character positions. The default 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 = *options*

Options (*over-opt,disp*)
This option has the following components:

over-opt
The options are PRINT, IGNORE, MERGE, and PRINT2.

PRINT
Prints all overprint lines as they would be on an impact printer. The second line is 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
Prints up to two consecutive lines 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 preserved for compatibility with other Xerox laser printing systems. The number of overprint lines is always printed on the accounting page. The default is NODISP.

PCC

Specifies the position (and possible translation) of the printer carriage control field.

Syntax LINE PCC =*options*

Options *(offset, trans-type)*
This option has the following components:

offset
Specifies the byte offset of the user portion of the record within each logical record to the printer carriage control (PCC) field. The default is 0.

trans-type
The options NOTRAN and TRAN specify whether or not the printer carriage control byte is to undergo code translation.

TRAN indicates that the byte is to be translated into LPS standard EBCDIC before being applied, using the translation defined in the CODE parameter of the VOLUME command.

NOTRAN (default) prohibits translation.

PCCTYPE

Specifies a carriage control set used in printing a job.

Syntax LINE PCCTYPE = *options*

Options *pcc-name*
The available options are: 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 keyword USER is defined using the PCC command.

ANSI
For offline systems, the default is ANSI.

IBM3211
For online systems, the default (and the only possible parameter) is IBM3211.

ICL2900
For HOST=ICL2900, the PCCTYPE is forced to 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. It checks that the LPS online interface performs, such as for bad or NO-OP parameters, are not done in the offline mode. There is no check for bad or NO-OP parameters, and they are treated as undefined carriage control parameters in the offline mode and default to the carriage control parameter of print-and-space-1-line.

UCSB

Accepts or suppresses the host-transmitted UCSB.

Syntax LINE UCSB = *options*

Options

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 universal character set buffer (UCSB). If IGNORE is coded, the CODE parameter of the VOLUME command defines the character translation. For online systems only.

VFU

Specifies the vertical format table you are using.

Syntax 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.

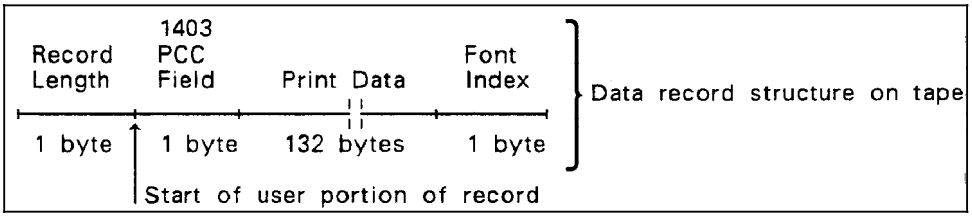
Examples

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**

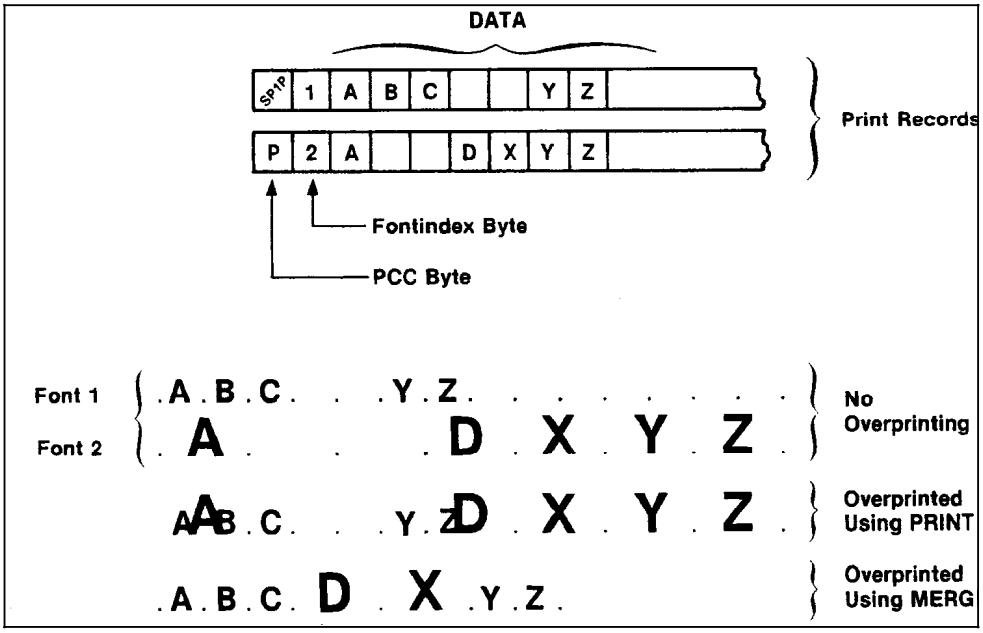


The print-data-offset option (the number of bytes between the start of the user portion of the record and the first character of the record to be printed) is one byte. The print-length parameter (the number of characters in the longest print line in the record) is defined as 132 bytes. The carriage control 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 font to select) 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, INKINDEX = 1;

Figure 3-3. **Sample print-data-offset option of the DATA parameter**



MESSAGE command

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 = *options*

Options

NONE
Specifies that no text message is output.

sc
Specifies a text message of up to 80 characters.

(*sc*,[*passnum*],[WAIT])
This option has the following components:

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.

WAIT
Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.

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

OTEXT

Sends a text message to the operator during job printing.

Syntax MESSAGE OTEXT = *options*

Options

NONE
Specifies that no text message is to be output to the operator during job printing.

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

(*sc*,[*passnum*],[WAIT])
This option has the following components:

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. 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.

(*sc*,[END],[WAIT])
This option has the following components:

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

END
Specifies that the text is displayed after the last copy of the report is printed. The maximum number of OTEXT characters which can be used in a report with GHO processing specified is 320 characters.

WAIT
Specifies that after the text is displayed, printing is suspended until the operator has responded with a CONTINUE parameter.

Considerations

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

OUTPUT command

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

The following parameters require specific user 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 3-2.

Table 3-2. **Hierarchy of OUTPUT command parameter restrictions**

If ...	Then ...			
	COLLATE =	DUPLEX =	FACEUP =	NTO1 =
FACEUP = YES	—	NO	YES	—
NTO1 = YES	YES	*specified option*	YES	YES
STAPLE = YES	YES	*specified option*	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 3-3.

Table 3-3. **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* was specified. If so, the input task tests on the operator's specified cluster-name.
- All applications that are printed on the 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 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.

IDR, IDFAULT
: If both parameters occur in a single OUTPUT command, the IDFAULT parameter overrides the implied default, the first ink specified in the ILIST parameter of the IDR command. This IDFAULT parameter remains in effect unless it is overridden by another IDFAULT parameter.

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 two 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.

Considerations

In association with the duplex printing mode DUPLEX=YES, a page containing only a form (no variable data can be printed with this page) may be printed on the back side of a user page. This feature can be used to print static data on the back of each page in a report without the use of DJDEs or the repetitive processing of that data. Different forms may be associated with different copies of a report by the use of multiple BFORM left parts on the same OUTPUT command. If BFORM specifications include some, but not all, copies of a report, those copies not included have blank back sides (no form).

Syntax

OUTPUT BFORM = *options*

Options

NONE
: No form is added to the associated report page of variable data.

form-id
: Specifies a 1- to 6-character filename (may be numeric,

alphabetic, or alphanumeric) which exists on disk. This file is created by compiling a forms description language (FSL) source file.

(*form-id*,[*init*],[*copies*], [(INKS,*inkref*$_1$,[*inkref*$_2$][,...])])
This option has the following components:

form-id
Specifies a 1- to 6-character filename (may be numeric, alphabetic, 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* are specified, the form applies to all copies of the report.

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

INKS
Specifies that the ink which follows is to be used for the form.

inkref
Identifies the inks that override the corresponding inks specified in the ink list of the form. If the form does not contain an ink list (an existing or new form that has not been colorized using FCU), the form is printed in BLACK. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

COLLATE

Specifies whether pages are collated or uncollated.

Syntax OUTPUT COLLATE = *options*

Options YES
Collates output pages.

NO
Does not collate output pages. If DUPLEX=YES, PDL ignores COLLATE=NO. If NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES and includes a warning in the JSL.

Considerations 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 = *number*

Options *number*

An integer value with range 1 (the default) to 32,767. 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

Selects cover pages from the AUX cluster. These cover pages may be placed at the front or back of each copy of a report.

Syntax OUTPUT COVER = *cover-opt*

Options *cover-opt*

The available options are: FRONT, (FRONT, SEP), BACK, BOTH, (BOTH, SEP), and NONE (default). FRONT selects a cover page (from the auxiliary tray) and places it at the front of each copy as the first page.

SEP specifies that each front cover does not have any data printed on it; also, no report data is printed on the back of a SEP cover.

BACK places a cover pag at the end of each copy. No report data is printed on back covers.

BOTH places both front and back cover pages on the report. The front of each copy is the first page of the copy.

NONE specifies that no cover pages are to be used.

Considerations The FEED MAIN operator command should be used in conjunction with this parameter to prevent the system from selecting the AUX tray 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.

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.)

CYCLEFORMS

Associates a set of forms with report pages in a cyclical fashion.

Syntax OUTPUT CYCLEFORMS = *options*

Options NONE

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

(*form-id*$_1$, [*form-id*$_2$][,...])

Specifies a 1- to 6-character filename (may be numeric, alphabetic, 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. You may enclose the *form-id* ink list pairs in parentheses.

(*form-id,* [(INKS,*inkref*$_1$,[*inkref*$_2$][,...])] ,[*form-id,* [(INKS,*inkref*$_1$,[*inkref*$_2$][,...])])

This option has the following components:

form-id
: Specifies a 1- to 6-character filename (may be numeric, alphabetic, 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. You may enclose the *form-id* ink list pairs in parentheses.

INKS
: Specifies that the subsequent *inkrefs* override any inks already defined for the job.

inkref
: The inks which override the corresponding inks specified in the ink list of the form. If the form does not contain an ink list (a form from another LPS that has not been colorized using FCU), the form is printed black, regardless of the inks specified in the CYCLEFORMS command.

  If you omit an *inkref* in the list, use commas to maintain the relative positioning of the remaining ink references.

Considerations

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.

The order of the list is the order in which the names are encountered. The number of *form-ids* is limited only by the number of forms retained on the disk.

Specifying CYCLEFORMS (or multiple forms in the FORMS parameter) 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 CYCLEFORMS or FORMS parameters is larger than the FORMS operator command value.

Examples

OUTPUT CYCLEFORMS=(*form*$_1$, [(INK,‘RED’)]);

OUTPUT CYCLEFORMS=(*form*$_1$, [(INK,‘RED’)]
[,*form*$_2$,(INK,‘BLACK’,‘RED’)]);

DENSITY

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

Syntax OUTPUT DENSITY= *options*

Options

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.

DESTINATION

Specifies the destination of the printed output.

Syntax OUTPUT DESTINATION= BIN|TRAY|EXPORT

Options

BIN
Sends the output to the output bin installed on your system.

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

EXPORT
Sends the output to the bypass transport. The EXPORT option takes effect only if the operator command options SELECT 1 or SELECT E are specified. For more information on the SELECT command, refer to the *Xerox 4850/4890 HighLight Color LPS Command Reference* or to the operator guide for your system.

DUPLEX

Specifies duplex or simplex printing.

Syntax OUTPUT DUPLEX = *options*

Options

NO
Specifies simplex printing (prints on one side of a sheet of paper).

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

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 will be printed using 8.5 by 14 inch paper size.

FACEUP

Specifies faceup delivery of pages.

Syntax OUTPUT FACEUP = *options*

Options

NO
Specifies sheets are delivered face down to the stacker tray.

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

FEED

Controls the stock on which the page is printed.

Syntax OUTPUT FEED= *options*

Options

OPR, MAIN, or 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.)

FORMAT

Specifies a page descriptor entry (PDE), listed in table 3-4, 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*

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.

Considerations Standard *pde-ids* are defined in table 3-4, such as FMT1 and FMT2. 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 through DJDEs.

To ensure reliability, enter the entire parameter keyword FORMAT. Do not abbreviate.

Table 3-4. **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 = *options*

Options

NONE
No form is added to the associated report page of variable data.

form-id
Specifies a 1- to 6-character filename (may be numeric, alphabetic, 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.

(*form-id*[,*init* [,*copies*]][,(INKS, $inkref_1$ [,$inkref_2$] [,...])])
This option has the following components:

form-id
Specifies a 1- to 6-character filename (may be numeric, alphabetic, 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.

init
Specifies the beginning copy 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.

INKS
Indicates that the *inkrefs* which follow are to override the inks defined in the form.

inkref
Identifies the inks that override the corresponding inks specified in the ink list of the form. If the form does not contain an ink list (an existing form that has not been colorized using FCU), the form is printed in BLACK. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list, the extra *inkrefs* are ignored.

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

Example: OUTPUT FORMS=(FORM3,3,2,(INK,'RED'));

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.

GRAPHICS

Specifies how to process graphics in a job.

Syntax OUTPUT GRAPHICS = *options*

Options

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
Means that the input stream contains no text, only graphics which are to be imaged, one per page.

(YES, NOSUB)
This option has the following components:

YES
Indicates there are graphics.

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.

(MOVE, NOSUB)
This option has the following components:

MOVE
Copies all referenced graphic disk files into the print file.

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). This appearance error is not be indicated by the system.

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.

Examples:

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

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

IDFAULT

Allows you to specify the default ink to use for objects whose ink is not given. The parameter remains in effect until you override it with another IDFAULT parameter.

Syntax OUTPUT IDFAULT = *inkref*

Options *inkref*

Represents the default ink to be used in all subsequent parameters that do not explicitly select an ink for print jobs not containing any color attributes, for example, the NUMBER parameter.

The default is the first referenced option in the ILIST parameter of the IDR in effect at the start of the page, or it is black if no IDR is stated ('DFAULT.DFAULT.BLACK').

Considerations Example: OUTPUT IDFAULT='RED';

IDR

Specifies the ink descriptor you want to use.

Syntax OUTPUT IDR = *idr-name*

Options *idr-name*

References an IDR command that is previously defined in the JDL, or references an IDR file that exists on the system. The first ink listed in the IDR commands ILIST parameter serves as the default ink for the job unless it is overridden by an IDFAULT parameter.

Considerations Example: OUTPUT IDR=IDR1;

IMAGE

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

Syntax OUTPUT IMAGE = *options*

Options (*vpos units* ,*hpos units* ,[*n*/[*d*]],[(INKS,*inkref*$_1$,[*inkref*$_2$][,...])])

This option has the following components:

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.

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*.

units
DOTS, centimeters (CM), inches (IN), or user-defined unit (UN). If no units are specified, inches are assumed. The default *units* value is IN.

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

The default is the top left corner scaled at 1/1.

n/d
Specifies the reference scale factor.

INKS
Indicates that the *inkrefs* that follow are to override the inks defined in the image.

inkref
Specifies the inks that override the corresponding inks specified in the ink list of the image file. Existing IMG files are printed using the system default black ink unless *inkrefs* are specified in the INK parameter.

Considerations Restricted IMG files are printed using the inks specified in the file unless *inkrefs* are specified in the INK parameter.

If you omit an *inkref* from the list, you must use commas to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

When printing images with 4850 color applications, the system uses only solid primary colors (e.g., red, blue, magenta).

Example: OUTPUT IMAGE=(1 IN,2 CM,INK,'RED'));

IRESULT

Specifies the result when objects imaged with different inks overlap.

The color resulting from overlapping objects may be only black or color for an entire page. For example, the same page could not have red text printed over a black background and also black text over a red background.

Two basic options are provided. These options specify the color of the resulting pixel on the page when both black and highlight pixels have been imaged for a single pixel location. The resulting pixel is black if BLACK is specified or highlight color if COLOR is specified. If neither is specified, the ink priority is the one chosen at sysgen.

These options can be specified on a page basis through DJDE commands. If specified more than once for a page, the last instance takes precedence. For monochrome (black) printers, this option is treated the same as BLACK.

Syntax

OUTPUT IRESULT = *options*

Options

BLACK
When two objects imaged with different inks overlap and black and highlight pixels coincide, the resulting pixel is black. This can be changed on a page basis.

COLOR
When two objects imaged with different inks overlap and black and highlight pixels coincide, the resulting pixel is highlight. This can be changed on a page basis.

The default is the ink priority chosen at system generation.

Considerations

For Interpress jobs, it is recommended that ink priority be set to black to ensure that areas where black is superimposed on a color background will print correctly. To do this, either set ink priority to black during system generation or modify your ENET.JSL to include IRESULT = BLACK and recompile the JSL. If you do not specify an IRESULT parameter in the JSL, the sysgened ink priority applies.

The color resulting from overlapping objects may only be black or a color for an entire page. For example, the same page could not have red text printed over a black background and also black text printed over a red background.

The IRESULT parameter can be specified on a page-by-page basis using the DJDE IRESULT. If it is specified more than once for a page, the last instance takes precedence. For monochrome printers, this parameter is treated the same as BLACK.

For Interpress jobs, it is recommended that the ink priority be set to BLACK. This ensures that areas where black is superimposed on a color background print correctly. To do this, set the ink priority to BLACK during system generation or modify the OUTPUT command of your ENET.JSL to include IRESULT=BLACK and recompile the JSL. If there is no IRESULT command in the JSL, the sysgen ink priority applies.

Example

```
OUTPUT  IRESULT = BLACK;
```

LOGO

Allows you to specify a logo for imaging on all pages of a report, unless you cancel it using CANCEL DJDE. You may specify numerous logos for the pages. The number is limited only by your system memory and buffer space, and by the complexity of data on the page. Logos can only be specified on a per-page basis with LOGO DJDE.

Syntax OUTPUT LOGO = *options*

Options (*name, vpos units ,hpos units,* [(INKS,*inkindex*$_1$,[*inkindex*$_2$][,...])])

This option has the following components:

name
The name of the logo.

vpos
Specifies the vertical position of the top edge of the logo, as an offset, relative to 0,0 on the current physical page (the position that would be specified by a PDE BEGIN of 0,0). This parameter is specified as a decimal number with up to three digits to the right of the decimal point, and may be in DOTS, XDOTS, centimeters (CM), inches (IN), or user-defined units (UN). The parameter and its unit must be separated by a space.

The default is (IN).

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

units
DOTS, centimeters (CM), inches (IN), or user-defined units (UN). An *xdot* is 1/600 unit of measurement. A logo specifying XDOTS can be created, edited, and compiled on an LPS with version 4 software. The default is IN.

INKS
Notifies the system that inkindex values follow.

inkindex
Specifies which inks in the ILIST parameter of the IDR command override the corresponding inks specified in the inklist of the logo file.

In order for this option to be valid, the referenced logo must be in color format. Logos may be converted from black-only format to color format using the File Conversion Utility (FCU). Refer to your *Xerox 4850/4890 LPS System Programming and Administration Guide* for more information.

Existing LGO files and restricted RES format LGO files are printed using the system default black ink unless *inkrefs* are specified in the INK parameter.

If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

If any of the inks to be substituted were specified in the logo definition with a NOSUBSTITUTION parameter, as in the FCU command, the system processes the ink requests according to what you have specified in the ISUBSTITUTE parameter of the ABNORMAL command.

Considerations

With OUTPUT LOGO, logos are printed on every page of the job until the system encounters a DJDE CANCEL or until the job ends.

Logos can only be specified on a page-by-page basis with the DJDE LOGO.

The number of logos allowed on a page varies depending on the structure of the logo and the amount of dynamic memory space available. The system allows an absolute maximum of 128 logos per page. If there is not enough dynamic memory available for the number of logos specified for a given page, the system notifies you with an error message.

MODIFY

Specifies the CMEs to be associated with report copies.

Syntax

OUTPUT MODIFY = *options*

Options

NONE

Processes data without modification.

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.

(*cme-id,* [*init*],[*copies*])

This option has the following components:

cme-id

Refer to the *cme-id* option above.

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*.

Considerations

If different CMEs are associated with different copies of a report, or if a CME is applied to only some copies of a report, then the CMEs are termed copy sensitive. You may not use copy-sensitive CMEs with COLLATE=NO, 871-CM, or online processing.

NTO1

Specifies, on a report basis, that 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 = *options*

Options

NO

Prints first page to last (1TON). NTO1=*n* allows you to specify the threshold after which input overrides your original request 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 NTO1 is 50 physical sheets, which is the nominal capacity of the stitcher for a single stitch.

Considerations

NTO1=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 = *options*

Options NO
Specifies that no page numbering is to be performed.

(*pnum,lnum,cnum,*[*findex*][*,inkref*])
This option has the following components:

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 goes 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.

inkref
Represents the ink to use for printing the page number string. If not specified, the current default ink is used.

Considerations You can use only one *inkref* in the NUMBER parameter.

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.

If both BFORM and NUMBER are specified, BFORM pages are numbered.

Example: NUMBER=(1,1,40,1,'RED');

OFFSET

Specifies offset control on a report basis.

Syntax OUTPUT OFFSET = *options*

Options ALL
Results in an offset of each copy of each report.

FIRST
An offset is to occur only on the first copy of a report.

NONE
Specifies that there is to be no offset at any time.

Considerations The OFFSET control of FIRST, ALL, or NONE may be modified by the ROFFSET command. If OFFSET=FIRST or ALL and ROFFSET criteria are 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.

PAPERSIZE

Identifies the paper size to use for printing the job.

Syntax OUTPUT PAPERSIZE = (*papersize* [,FORM])

Options *papersize*

A4: 8.27 by 11.69 inches/210 mm by 296 mm
USLEGAL: 8.5 by 14 inches/216mm by 356mm
USLETTER: 8.5 by 11 inches/216mm by 279mm
(*x,y*): a variable paper size selected at system generation. The *x* is the short edge of the sheet and the *y* is the long edge. The *x* and *y* variables may be specified in either inches (IN) or centimeters (CM).

The default is the paper size selected at system generation.

FORM

The paper size specified in the JSL is overridden by the paper size of the form called out for a given page. If FORM is not specified, or if the form does not contain paper size information, the paper size specified in the JDE or JDL, or the default paper size, is used. This paper size determination is done on a page-by-page basis. In other words, the paper size used for one page does not affect the paper size of subsequent pages.

Considerations If *papersize* is not specified, the LPS uses the system default paper size specified during system generation when printing the report.

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 through 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 keyword 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 graphics disk files at the end of a report.

Syntax OUTPUT PURGE = *options*

Options

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

NO
Does not delete graphics disk files.

Considerations 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 OUTPUT RESOLUTION = *options*

Options

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.

The default is the parameter specified at system generation.

SF1FUNCTION

Provides control over third-party finishing devices that conform to the DFA standard. Using this command, you can invoke sheet finisher function number 1 through the DFA channel C6.

Syntax OUTPUT SF1FUNCTION = {NO|YES}

Options

NO
Does not invoke the sheet finisher function number 1.

YES
Invokes the sheet finisher function number 1.

SF2FUNCTION

Provides control over third-party finishing devices that conform to the DFA standard. Using this command, you can invoke sheet finisher function number 2 through the DFA channel C7.

Syntax OUTPUT SF2FUNCTION = {NO|YES}

Options

NO
Does not invoke the sheet finisher function number 2.

YES
Invokes the sheet finisher function number 2.

SHIFT

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

Syntax OUTPUT SHIFT = *options*

Options

NO
Specifies that no shift occurs (v_1=0 and v_2=0).

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

(v_1, v_2)
This option has the following components:

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 *v2* is -75 to 75.

Considerations

On the 4850, if the origin of a print line falls off the page, the entire line of data fails to print.

On the 4890, if the origin of a print line falls off the page, the system prints as much of the line as possible and displays the following message once per report rather than for every page that has the error within the report:

```
OS6905 Data origin off page
```

Graphics and logos, however, are not allowed to start off the page.

When duplex shift is used, the left margin must be at least .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 = *options*

Options

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 because the LPS does not cycle down. If the exact paper size is not available, the system aborts the report.

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 OUTPUT STAPLE = *options*

Options

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 the defined override conditions are not encountered. To stitch copy sets properly, sheets must be delivered to the finisher's staging tray faceup.

Considerations

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 OUTPUT STOCKS = *stockset-name*

Options *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.

There is no default.

Considerations 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 on the OUTPUT command or an IMAGE, ALTER, or GRAPHIC DJDE parameter.

Syntax OUTPUT UNITS = *size in dots*

Options *size in dots*

The size defines the number of dots in one unit and is specified as a decimal number with up to two digits to the right of the decimal point. When used to compute the position of a graphic, the result is rounded to the nearest dot.

There is no default.

XMP

Xerographic mode persistence (XMP) controls xerographic mode switching (XMS) to use for the report. Depending on the option specified, the effect is to maximize throughput or minimize toner usage.

Syntax OUTPUT = *options*

Options DEFAULT

Specifies that the sysgened XMP is used for the report.

REPORT

Specifies that when the system determines that a report requires highlight color mode for any page of the report, all pages remaining unprinted of the report are printed in the highlight color mode.

Examples

Example 1

```
OUTPUT    PAPERSIZE = USLETTER,
          DUPLEX = NO,
          IDR = IDR0,
          NUMBER = (1,1,80),
          FORMAT = SMPPOE;
```

Example 2

```
OUTPUT    GRAPHICS = MOVE, UNITS = 150,
          CYCLEFORMS = NONE, FORMAT = PDE1,
          RESOLUTION = 300;
```

Example 3

```
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 command

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 3-4. 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-Z and 0-9). 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 by DJDEs. PDEs are called out on the FORMAT parameter of the OUTPUT command or a DJDE FORMAT parameter. PDEs must be precompiled to be accessed by a DJDE. Standard PDE specifications are provided on the operating system software (OSS) system tapes.

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 position to the next logical page. The following should be noted:

- Each online banner page is positioned as the first logical page of a new physical sheet (if BANNER 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 cause 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 cause the current page position to move 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 cause the current page position to move 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 *ac:*PDE BEGIN = *options*

Options *vpos*

The *vpos* parameter specifies the vertical position of the first character of the first print line on the logical page. It may be specified in inches (IN) or centimeters (CM). The default is 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 2.35 CM (all legal specifications). The default is .18 IN.

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. The default is .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, the default is FMT1.

FONTS

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

Syntax *ac:*PDE FONTS = *options*

Options (f_1[,f_2][,...])

This option has the following components:

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.

((f_1,s_1)[,(f_2,s_2)][,(...)])

This option has the following components:

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.

s

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

((f_1,s_1[*units*])[, (f_2,s_2[*units*])(,...)])

This option has the following components:

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.

s

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

units

Options are lines per inch (LPI), XDOTS, or the dots per print line (DOTS) to be associated with the font. 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.

PMODE

Specifies the printing mode for each physical sheet.

Syntax *ac:*PDE PMODE = *options*

Options 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.

Considerations

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 table where information about the fonts is cached.

If the patch to enable the use of Merganthaler 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 to 260 bytes of dynamic memory.

Examples

Example 1

```
PDE1:    PDE    PMODE = LANDSCAPE,
                BEGIN = (.861, .7)
                FONTS = (LO112B, LO1BOB);
PDE2:    PDE    PMODE = PORTRAIT,
                BEGIN = (1.3, .37),
                FONTS= ((PO8TYA, 6.8), (PO8OAA, 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 = (PO71TA, PO8OAA);
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 command

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.

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 filename) from either a JDL or a DJDE. RTEXT parameters must be precompiled to be accessed with a DJDE.

To catalog RTEXT parameters, an identifier (of the type *ac*) must be included on the ROUTE command. This identifier is used to reference the RTEXT parameters with the RTEXT=*rtext-id* parameter of the ROUTE command. An example of cataloged RTEXT usage is illustrated below. If the cataloged RTEXT is defined within a JSL, it is also automatically applied to that JSL.

```
ROUTE1: ROUTE  RTEXT=('USER 1',1,33,64),
               RTEXT=('BLDG 1',1,34,64),
               RTEXT=('USER 2',2,33,64),
               RTEXT=('BLDG 2',2,34,64);
```

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 ROUTE RFORM = *options*

Options NONE
Specifies that no form is to be printed.

form-id
If the name of a file is 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 which is being defined as a cataloged file, but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file.

RTEXT

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

Syntax ROUTE RTEXT = *options*

Options NONE
Specifies no text is to be printed on a separate page.

sc
Specifies the message to be printed (1 to 132 characters). It is printed with the first font specified in the FONTS parameter of the PDE command (refer to *fontindex*).

(sc[,*passnum*/ALL][,*line*][*col*][,*fontindex*])
This option has the following components:

sc
Refer to the *sc* option above.

passnum or ALL
An integer or the keyword ALL. An integer number specifies the pass (copy ply) to which the text applies.

line
Specifies the line number on which the first line of a block of RTEXT message is printed. The default is line 1 for the first text string of the pass. Otherwise, the default is the next line of the page. Only one RTEXT parameter may be specified per line for a *passnum*.

col
The column number on which to print the first character of a block of RTEXT messages (default is column 1).

fontindex
Identifies the index (starting with 1) of the PDE font with which the text is printed.

rtext-id
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.

Considerations

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).

If more than one font is used to print any number of RTEXT strings on a page, the line and the character spacing values of the different fonts are used to place RTEXT on the page.

If a proportional pitch font is used, the starting column number for printing the RTEXT is determined using the character spacing value for the space character in the character font.

If RTEXT is specified along with COVER=FRONT,SEP or BOTH,SEP, the RTEXT pages are printed on the covers. Front covers are picked only on copies for which there is RTEXT. If RTEXT is specified in non-collate mode, only the routing page for copy one is printed.

ac:STOCKSET command

Defines the paper stocks used in a report as follows:

First, a STOCKSET defines a set of clusters used by a particular application.

Second, clusters (as created via CLUSTER or CLP system commands) define a collection of paper trays which may contain various paper stocks. Paper trays can be part of only one cluster in a stockset.

Stocksets are thus 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*$_1$,[*stock-descriptor*$_2$][,...])

Options (*stock-descriptor*$_1$,[*stock-descriptor*$_2$][,...])

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

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 = *options*

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.

SYSPAGE

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

Syntax *ac:*STOCKSET SYSPAGE = *options*

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. OUTPUT FEED = *stock-reference* allows you to change paper stocks associated with a job, without altering the stock references in the data application.

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

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 size of the operator entered. If the paper size is not available, the system attempts to print the page on a different paper size chosen from the same set of trays as would normally be used. This occurs without any warning, caution, or cycle down. If a larger paper size is available, it is used. If a larger paper size is not available, a smaller paper size is used. This is the only time a page may be printed on a paper size smaller than what was specified to format the page.

When a system-generated page is not printed on the same paper size used to format it, the positioning of the data on the physical page is not guaranteed. For example, when printing on a larger paper size, the data does not fall off the physical page, but it may not appear in the correct location. When printing on a smaller paper size, the data may fall off the physical page and appear incomplete.

Example

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), ('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, the application 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 command

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 prejob page alignment and for page bottom-of-form overflow processing. For all PCCTYPEs except ANSI and user-defined PCCs, the prejob page alignment is to top-of-form so that the first carriage control parameter of the job will print and space one line. 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 referencing 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. One of the characters must be a letter.

ASSIGN

Specifies the output line-to-channel assignments.

Syntax *ac:*VFU ASSIGN = *options*

Options (*channo, lineno*)

This option has the following components:

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.

(*channo,* (*lineno*$_1$[,*lineno*$_2$](,...)))

Refer to the *channo* and *lineno* options above.

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 *ac:*VFU BOF = *value*

Options *value*

The number of lines from the top of the output page to the last print line on the page (bottom of form). The BOF specification is independent of channel assignments. BOF should be greater than or equal to the largest line number assigned to a channel. The maximum value for BOF is 255. 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 *ac:*VFU TOF = *value*

Options *value*

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.

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.

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 system is printing a report and the current line number is 11. If a skip-to-channel-1-and-print parameter were issued, a page transition occurs. Printing begins on page 5 (assigned to channel 1), which is the TOF on the new page.

If you issue a skip-to-channel-2-and-print parameter when the current line is 11, the next line printed is 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 command

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 and operating system labels). In processing stacked reports, the system checks each record for the logical end-of-report specification, as defined by the TEST parameter of the RSTACK and BANNER commands. When BANNER is coded, the user specifies the number of consecutive banner pages which must satisfy the test criteria before end-of-report occurs.

HCOUNT

Specifies the total number of consecutive banner pages.

Syntax BANNER HCOUNT = *value*

Options *value*

Specifies the total number of consecutive header banner pages that must be detected to satisfy the test expression. The default is 0.

Considerations Refer to the “Considerations” section of the “TCOUNT” parameter section in 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 = *options*

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

(*offset,length*)
This option has the following components.

offset
Specifies the offset in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record.

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

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 = *options*

Options NONE
Specifies that no report name is to be selected.

(*offset,length*)
This option has the following components.

offset
Specifies the offset in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record.

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

TCOUNT

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

Syntax BANNER TCOUNT = *value*

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

Options *test-exp*

If a *test-exp* is satisfied, the page containing the record tested is considered a banner page.

TYPE

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

Syntax BANNER TYPE = *options*

Options 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 the user 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.

BDELETE command

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

BDELETE TEST = *test-exp*

Options

test-exp

Defines test expression for selecting blocks for printing. The block is deleted (BDELETE) for printing if a *test-exp* is satisfied.

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 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.

You can only use the TEST parameter with the CRITERIA command in the CONSTANT or CHANGE mode.

BSELECT command

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 BSELECT TEST = *test-exp*

Options *test-exp*
Defines test expression for selecting blocks for printing. The block is selected (BSELECT) for printing if a *test-exp* is satisfied.

Considerations 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.

You can only use the TEST parameter with the CRITERIA command in the CONSTANT or CHANGE mode.

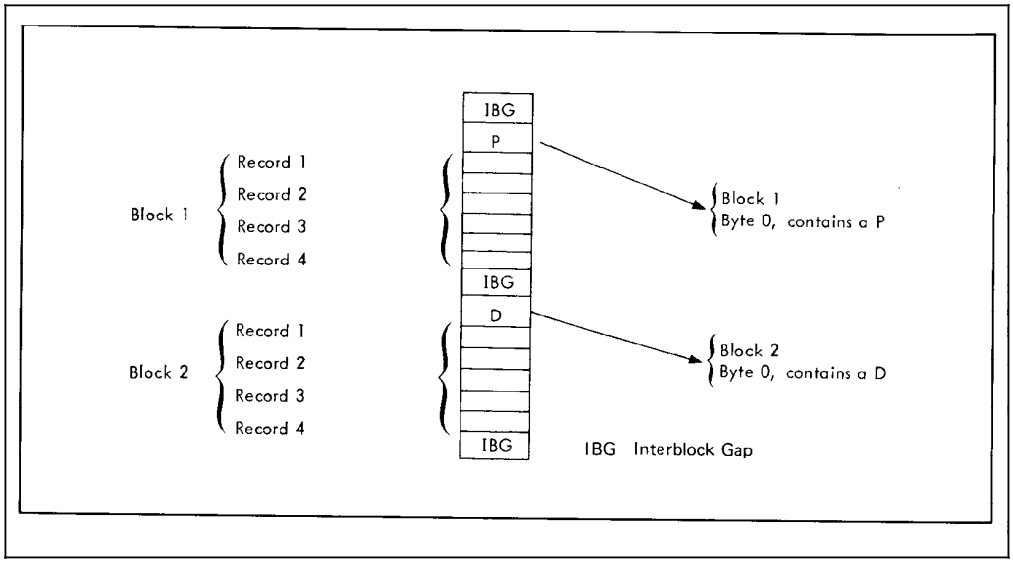
Examples

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 command

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.

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)

Options (*offset,length,* NE, LAST)

This option has the following components:

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.

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.

The system ignores the CRITERIA CHANGE command if you use it in conjunction with a BSELECT or BDELETE statement.

CONSTANT

Defines test specifications for a logical processing function with constant mode criteria in block processing.

Syntax

*ac:*CRITERIA CONSTANT = (*offset,length,* NE|EQ, *tab-id*)

Options

(*offset,length,* NE|EQ, *tab-id*)
This option has the following components:

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.

NE or EQ
Indicates not equal to (NE) or equal to (EQ).

tab-id
The identifier of a TABLE command.

This option has 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*)

Options

(*init, count*)

This option has the following components:

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.

The default is all lines.

Considerations

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 are 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 and 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 that 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.

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*)

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 1-255). 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 multi-national 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 ‘thousands’ 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 'thousands 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 'thousands 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 aren't 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. Don't 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 seperators. For example, '50000' is faster to evalute than '050,000.00', since the former expresses the same value using only half as many characters as the latter.

LMODIFY command

This command allows you to add highlight color to existing black-only applications.

Syntax

LMODIFY TEST=*test-exp* [,INK=*ink-index*] [,SELECT=*select-op*] [,..., TEST=*test-exp* [,INK=*ink-index*] [,SELECT=*select-op*]

Options

test-exp

Specifies the test to be performed using either one or two criteria. If two test criteria are used, they are linked by an AND or OR operator.

Example:

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.

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:

select-op:=LINE
select-op:=(*offset,length*)

LINE

Print 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.

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 FONTINDEXing (and optionally INKINDEXing) in conjunction with OVERPRINT=MERGE will override LMODIFY.

RAUX command

Allows you to define a select page from an auxiliary tray test.

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

RAUX TEST= *test-exp*

Options

test-exp
If a *test-exp* is satisfied, a page is selected from the auxiliary tray.

Considerations

If a record intended to satisfy the criteria for RAUX is suspended by RSUSPEND, that record is not checked for the RAUX criteria. 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).

Example

```
T1:   TABLE      CONSTANT = 'CUSTOMER COPY';
C1:   CRITERIA   CONSTANT = (20, 13, EQ, T1);
      RAUX       TEST = C1;
```

Figure 4-2. **Sample RAUX command usage**

```
                                  13 characters (length)

XXXXXXXXXXXXXXXXXXXXCUSTOMER COPY

20 character positions (offset)
```

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 auxilliary tray. The Xs are depicted in the example to illustrate the specified offset of 20 character positions.

RDELETE command

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 test expression for deletion of records from printing.

Syntax RDELETE TEST = *test-exp*

Options *test-exp*

If a *test-exp* is not satisfied, the record is selected for printing. If the RDELETE *test-exp* is satisfied, the record is deleted from the printed output.

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's 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, neither the RAUX nor suspend nor resume functions.

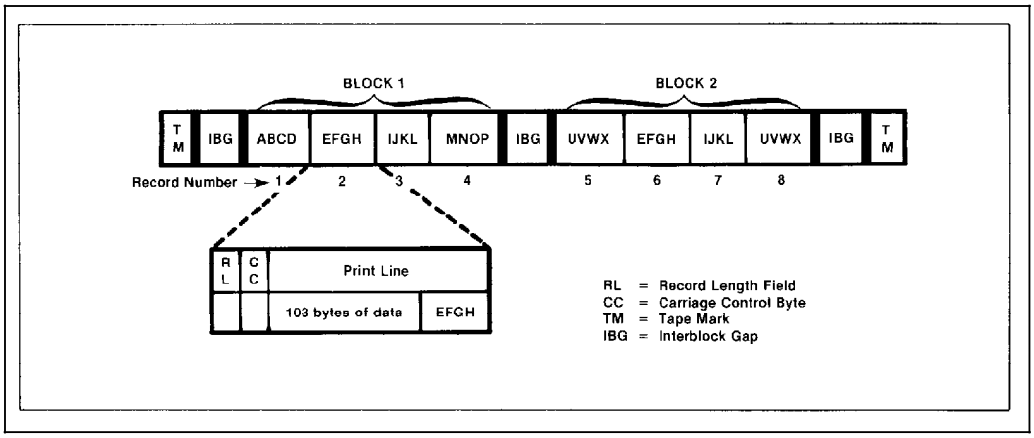
Example

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 4-3 and records 2 and 6 would not be printed.

Figure 4-3. **Sample RDELETE command usage**



RFEED command

Allows you to change cluster names on a page-by-page basis without using DJDEs. The RFEED command allows you to specify paper fed from different clusters if certain criteria are met on a record basis. RFEED is essentially an extension of the RAUX command in that it allows you to specify virtually any *cluster-name* or *cluster-reference* without being limited to the AUX cluster.

TEST

Specifies paper fed from different clusters if certain criteria are met on a record basis.

Syntax RFEED TEST = *((test-exp,clu-def)*[,*(test-exp,clu-def)*][,...])

Options *((test-exp,clu-def)*[,*(test-exp,clu-def)*][,...])

This option has the following components:

text-exp
References a criteria table to be used.

clu-def
Defines the cluster to use for the page, either a *cluster-name* or a *cluster-reference*. There is no default.

Considerations 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 criterion 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 command

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 instance, 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 ROFFSET PASSES = *options*

Options

ALL
: Specifies that the satisfied criteria cause an offset on all passes of a collated print run.

FIRST
: Specifies that the satisfied criteria cause an offset only on the first pass of a collated print run.

Considerations

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 are satisfied on the first page of a report, the normal offset from the preceding report is nullified.

TEST

Defines test expression for offsetting pages to the stacker tray.

Syntax ROFFSET TEST = *test-exp*

Options *test-exp*

If a *test-exp* is satisfied, the record causes an offset in the stacker tray. If a *test-exp* is satisfied in duplex mode, the logical page on which the record occurs is the first logical page of a new sheet.

Considerations The ROFFSET feature prints the record satisfying the test expression according to the normal job parameters.

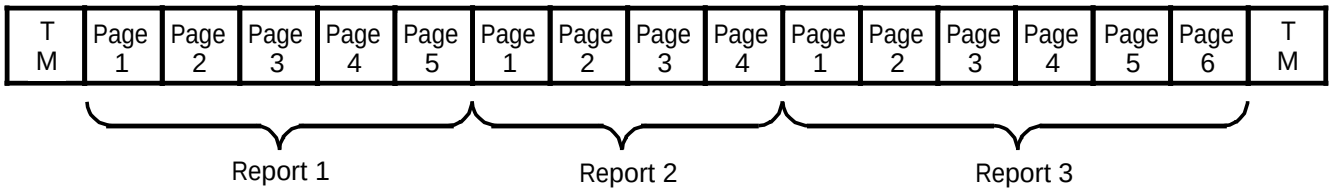
If a record satisfying the ROFFSET test criteria is not selected for or deleted from printing, the offset indication is maintained and used to cause the offset on the next record printed.

Example

A file has multiple reports without any delimiters separating the reports, as illustrated in figure 4-4. 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 4-4. **Sample ROFFSET command parameters**



RPAGE command

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.

Syntax RPAGE SIDE = *side-opt*

Options *side-opt*

Available *side-opt* options are: NUFRONT (the default), BACK, NUBACK, NEXT.

In duplex (with no BFORM), *side-opt* produces the following results:

- If NUFRONT is coded, the specified logical page is positioned as the first logical page on a new sheet. If BACK is coded, the specified logical page is positioned as the first logical page of the next available back.
- If NUBACK is coded, the specified logical page is positioned as the first logical page on the back of a new sheet.
- If NEXT is coded, the specified logical page is positioned as the first logical page on the next available side, the back of the current sheet or the front of the next sheet.

The logical page is moved unless it is already properly positioned as the first logical page on the specified side.

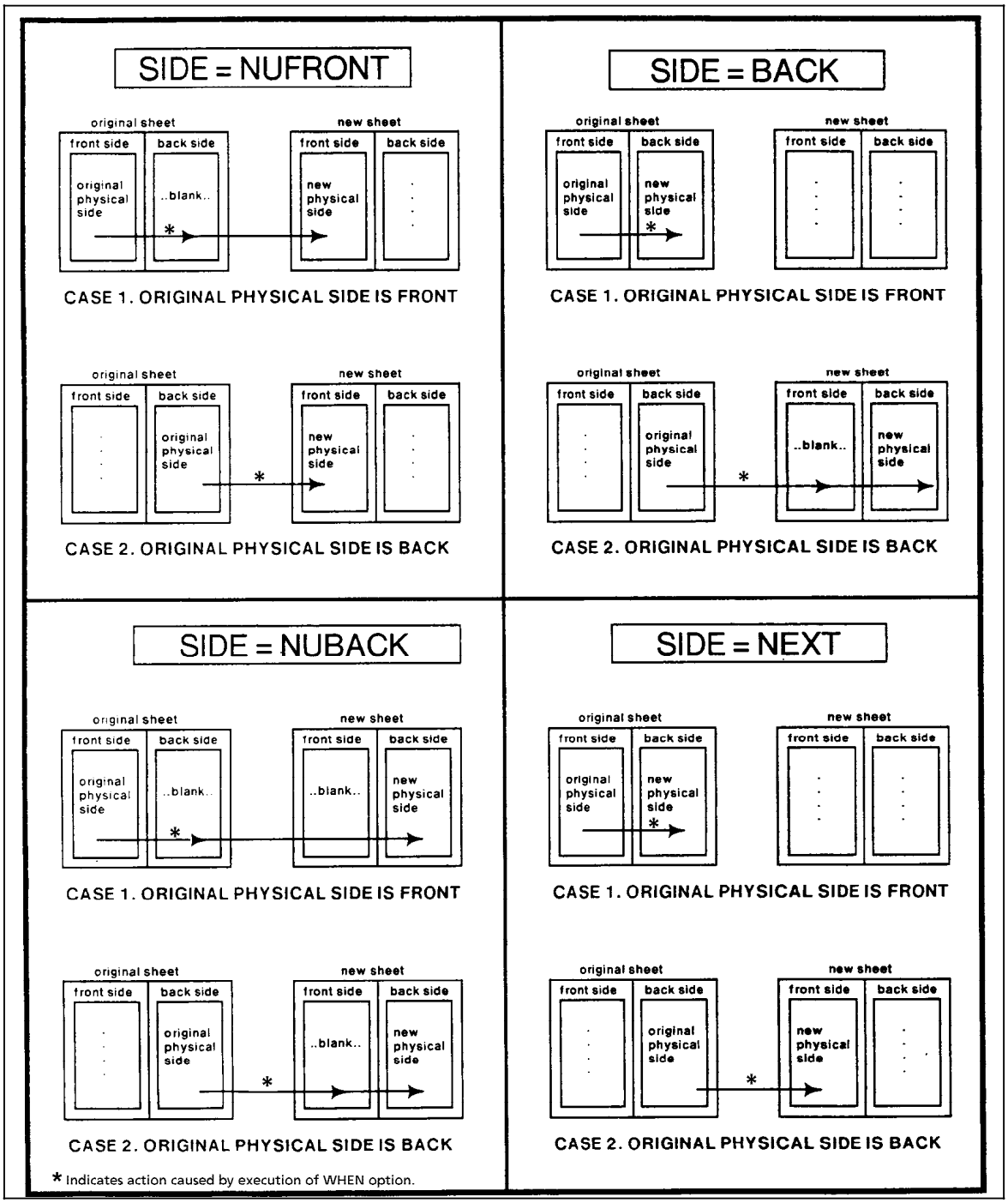
In simplex, or in duplex with BFORM, the SIDE parameter defaults to NUFRONT and the specified page is positioned as the first logical page. If the logical page is already properly positioned, a blank sheet is not created.

offset-opt

Available *offset-opt* parameters are: NOFFSET (the default), OFFSET. If OFFSET is coded, the sheet on which the repositioned logical page occurs is offset in the output stacker. If NOFFSET is coded, no offset occurs.

Example Figure 4-5 illustrates SIDE parameter options.

Figure 4-5. **SIDE parameter options**



TEST

Defines the test expression for detection of an RPAGE record.

Syntax RPAGE TEST = *test-exp*

Options *test-exp*
The logical page on which this record normally prints is called the current logical page. 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 take 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 = *options*

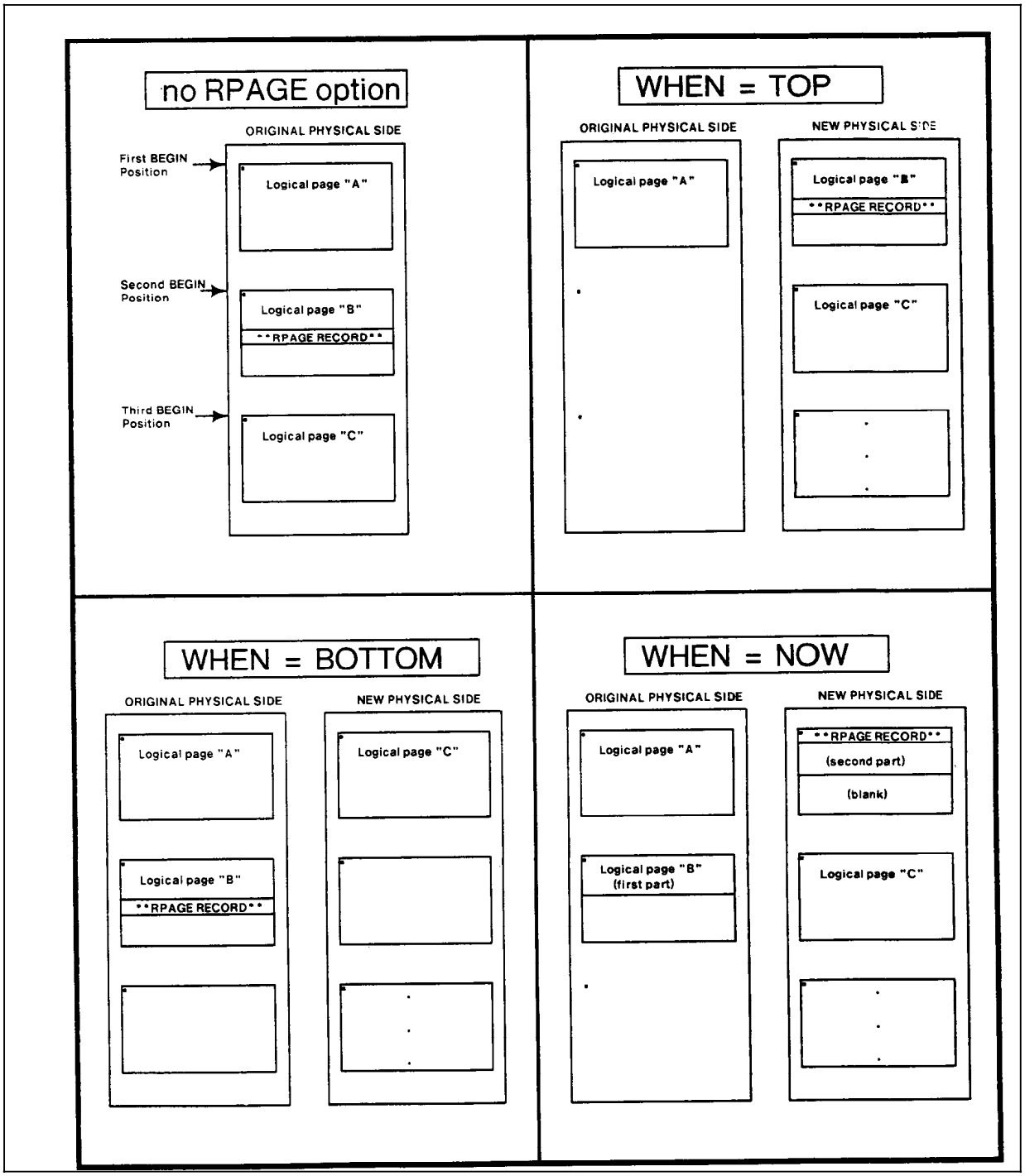
Options TOP
The current logical page is repositioned.

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.

Considerations Figure 4-6 illustrates WHEN parameter options.

Figure 4-6. **WHEN parameter options**



RRESUME command

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.

Syntax RRESUME BEGIN = *options*

Options

NEXT
If NEXT is coded, the record satisfying *test-exp* is not printed, and printing begins with the next record.

CURRENT
Specifies whether printing resumes on the current or next record for offline jobs. If CURRENT is coded, the record is printed.

TEST

Defines 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.

Syntax RRESUME TEST = *test-exp*

Options

test-exp
If the *test-exp* is satisfied, the record is used to resume printing following print suppression.

Considerations

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 resume processing. If a record satisfying the resume 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 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.

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.

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 4-7 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 4-7. **Sample RRESUME and RSUSPEND data**

```
//JOB        .FOROO1AO
OPTION       CATAL
//ASSIGN     SYSRLB, 3340, TEMP        Records not printed
  INCLUDE  IMPCBBM
//EXEC       FCOBOL
             .
             .
             .                         Records printed
             .
             .
```

RSELECT command

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 RSELECT TEST = *test-exp*

Options *test-exp*
If a *test-exp* is satisfied, the record is selected for printing.

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 selection is performed prior to RAUX and suspend or resume, offline. If a record satisfying either the RAUX suspend or resume test criteria was previously not selected for printing, neither the RAUX nor suspend nor resume functions.

RSTACK command

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 usually 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.

Syntax RSTACK ACCTINFO = *options*

Options (*offset, length*)

This option has the following components:

offset
Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record.

length
The number of bytes in the subfield (1 to 64).

DELIMITER

Specifies whether all or single records are part of a subsequent report.

Syntax RSTACK DELIMITER = *options*

Options 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.

Considerations 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 = *options*

Options (*offset, length*)

This option has the following components:

offset

Specified 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 no report name is selected.

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 = *options*

Options 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. The parameter BOTH in duplex is treated as BIN.

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.

TRAY

Specifies that report delimiters are printed and the output is delivered to the sample print tray.

TEST

Defines test expression for end-of-report conditions.

Syntax RSTACK TEST = *test-exp*

Options *test-exp*

Defines 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. 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.

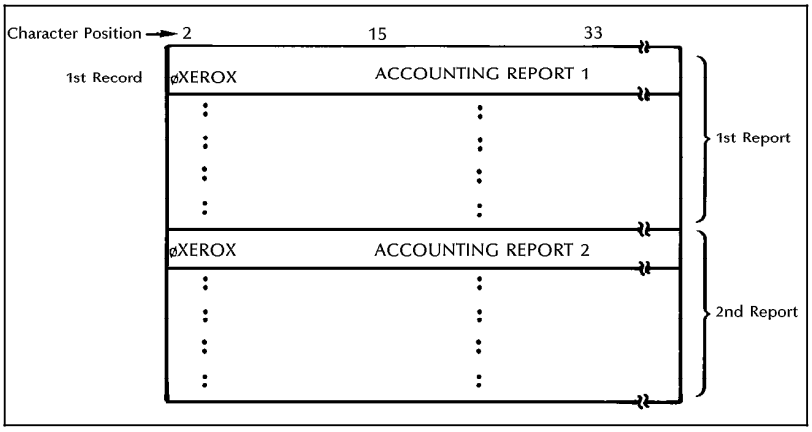
A record that is an RSTACK delimiter (that is, it satisfies the TEST expression) cannot be deleted from (or not selected for) printing by the RSELECT or RDELETE logical processing. If the record satisfies the RSTACK test criteria but is not a delimiter, it can be deleted from (or not selected for) printing but still causes report separation.

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 4-8. The delimiter record is not printed with the report.

Figure 4-8. **Sample RSTACK command**



RSUSPEND command

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 which satisfies the test criteria specified on the RSUSPEND command, printing is suspended. If BEGIN=CURRENT is coded on the RSUSPEND command, this record is not printed. If BEGIN=NEXT is coded, the record satisfying the test criteria is printed, and records are discarded beginning with the following record.

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.

Syntax RSUSPEND BEGIN = *options*

Options

NEXT
If NEXT is coded, the record satisfying *test-exp* is not printed, and printing 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 *test-exp* does not print.

TEST

Defines 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*

Options

test-exp
If the *test-exp* is satisfied, the record is used to resume printing following print suppression.

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 command

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.

Syntax *ac*:TABLE CONSTANT= *options*

Options (sc_1, [sc_2][,...])

Specifies string constants composed of literal characters and masking characters (as defined by the ASK 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.

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 CONSTANT parameter), indicates the character positions that have special type testing, and defines the test.

Syntax

ac:TABLE MASK= *options*

Options

(*ignore-char*[,*charspeci*$_1$], [*charspeci*$_2$][,...])

This option has the following components:

ignore-char

Specifies a character to appear in those positions within a string constant that are not to be compared.

charspeci

(i = index number) specifies a character that appears in those positions within a string constant where a test for type *i* 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.

Examples

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: *xx*9*x*, *x*999*x*9, and 99*x*999. 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 parameters embedded within the input data stream. You use them to modify a printing environment which 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 = *options*

Options (*name*,*vpos units*, *hpos units*,[*n*[/*d*]]),[(INKS,*inkref*$_1$,[*inkref*$_2$][,..])])

This option has the following components:

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.

hpos
Specifies the horizontal position of the left edge of the graphic relative to 0,0 on the current physical page.

units
Available 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.

n/d
Specifies the reference scale factor.

INKS
Specifies that the *inkref* is to override the one specified in the ink list.

inkref
Defines the inks that override the corresponding inks specified in the ink list of the image file. Existing IMG files are printed.

Examples

```
ALTER=(BUS,5,5,1,(INK,'RED'));
ALTER=(IMA,1,3,5,(INK, 'RED')), END;
```

ASSIGN

Assigns a VFU channel to a page line number or set of line numbers. ASSIGN is a record-oriented DJDE.

Syntax

ASSIGN = *options*

Options

(*channo*,*lineno*)
This option has the following components:

channo
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 cannel or a different channel.

lineno
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.

(*channo*,(*lineno*$_1$[*lineno*$_2$][,...]))
This option has the following components:

channo
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 cannel or a different channel.

lineno
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.

Considerations

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. This command is a record-oriented DJDE.

Syntax BATCH = *options*

Options 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.

Considerations 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.

Syntax BEGIN = *options*

Options (*vpos units* ,*hpos units*)

This option has the following components:

vpos

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.

hpos

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.

units

Available options are: IN (inches), and CM (centimeters). The default is 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.

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.

Syntax BFORM = *options*

Options NONE

Does not add a form to the associated report page of variable data.

form-id

Specifies a filename that exists on disk. This file is created by compiling an FSL source file with the FDL system task.

(*form-id*,[*init*,[*copies*]],[(INKS,$inkref_1$[,$inkref_2$][,...])])

This option has the following components:

form-id

Specifies a filename which exists on disk. This file is created by compiling an FSL source file with the FDL system task.

init

Specifies the beginning ply number to which a specified form applies. This defaults to the first or next copy.

copies

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

INKS

Specifies that the *inkref* that follows overrides the ink defined in the job.

inkref

Specifies the inks that override the corresponding inks specified in the ink list of the form. If the form does not contain an ink list (an existing form that has not been colorized using FCU), the form is printed in BLACK. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

Example BFORM=(FORM, 1, 2, (INK, 'RED')), END;

BOF

Specifies the bottom-of-form line number. BOF is a record-oriented DJDE.

Syntax BOF = *value*

Options *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 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. The default is 66.

Considerations 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.

If bottom-of-form (BOF) is altered by DJDE to a line number smaller than the current line from which the DJDE was processed, input effects an immediate page transition.

If 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), input spaces from TOF a number of lines equal to the difference between its location at DJDE and the new BOF.

C *text*

Specifies comment text in the DJDE record. C *text* is a record-oriented DJDE.

Syntax C *text*

Options *text*

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.

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.

Syntax CANCEL = *options*

Options (*name*$_1$[,*name*$_2$][,...])

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.

COLLATE

Specifies whether the pages of a job are collated (placed into sets) or uncollated (placed together). COLLATE is a page-oriented DJDE.

Syntax COLLATE = *options*

Options 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.

COPIES

Specifies the number of copies or sets of the pages to produce at the next page boundary. COPIES is a page-oriented DJDE.

Syntax COPIES = *options*

Options *number*

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 *number* is set to zero, this copy count remains in effect until a new copy count or end of report is encountered. DJDEs encountered while *number* is zero is in effect are processed normally.

Considerations The number of copies specified in the DJDE overrides the number of copies in the OUTPUT command. You may encounter problems when the copy count at the start of the report is 1 (either by virtue of the JDE or the last DJDE encountered at the start of report), and the copy count is later increased. There is no problem if the COPIES=DJDE is in the delimiter at the start of report and the value of the COPIES

=DJDE is at least 2. The problem is caused when output processing releases space as soon as it believes that it no longer needs it.

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 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.

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

Options (*pdo*, *length*)

This option has the following components:

pdo

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.

length

Specifies the maximum length of printable data within each logical record. For offline systems, the default is (1,132). For online systems, the default is (0, 150). For HOST = RSX11(FLX), the DATA parameter must be included in the user JSL since the default *pdo* 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.

Syntax DEPT = *sc*

Options *sc*

Specifies the accumulation of accounting statistics for reports on a name basis. It takes effect at the next page boundary. 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.

Syntax DESTINATION = {BIN|TRAY|EXPORT}

Options

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.

Syntax DUPLEX = *options*

Options

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.

END

Specifies the end of DJDE information.

Syntax END;

Specifies the end of DJDE information. When an END parameter is encountered, the system applies all DJDE information specified to the current printing environment at the next page or record boundary. Note that after an END parameter, a DJDE is compiled and record-oriented DJDEs take effect immediately.

FEED

Controls the stock (type of paper) on which the page image prints. FEED = *stock-reference* references the stock assigned to a *stock-name* by the STOCKSET command in effect at the time the page is printed.

FEED is a page-oriented DJDE.

Syntax FEED = *options*

Options OPR, MAIN or AUX

Provides compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN. FEED MAIN should be entered before control paper feeding using DJDEs.

stock-reference

Allows users 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=*cluster-reference* should always be used.)

FILE

Enables files to load to the system disks while a print job is in progress. The print job may be invoked solely for the purpose of downloading files. Although intended primarily for use with online systems, FILE is not restricted to online use.

FILE is a record-oriented DJDE relating to graphics.

Syntax FILE = ([*filename*],[[*file-type*],[[*f*],[[*s*],[*n*]]]])

Options This option has the following components:

filename

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 (default).

s

Specifies one of the following file storage parameters:
D = delete after report printed
P = permanent (default).

n

Specifies the maximum number of card-images.
The default is 120.

Considerations For LPS-labeled files, the DJDE FILE parameter may be specified as FILE=0. If the *filename* 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 filename and type are obtained from the label record that precedes the file data.

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. 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 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.

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.

Syntax

FONTINDEX = *options*

Options

NONE
Specifies that there is no font index.

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

(*offset*,[*init-val*][,*bit-opt*])
This option has the following components:

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

init-val
Can be one of the following: ONE or ZERO. ONE specifies that an index value of 1 is associated with the first font in the font list. When the *bit-opt* parameter is used, the *init-val* ONE parameter must also be used. ZERO specifies that an index value of 0 is associated with the first font in the font list, an index value of 1 is associated with the second font in the font list, and so forth. The default is ONE.

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. These low-order bits specify an index value into the font list of the current PDE. The default value is 4.

Considerations

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 within a DJDE takes effect on the next page boundary. FONTS is a page-oriented DJDE.

Syntax FONTS = *options*

Options (f_1,[f_2][,...])

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 parameters must be used.

((f_1,s_1),[(f_2,s_2)][,...])

This option has the following components:

f

Specifies a 1- to 6-alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability and up to 94 fonts on a single page). If the number of fonts to be invoked exceeds the size of one DJDE record, multiple FONTS parameters must be used.

s

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; minimum value is 10 XDOTS or DOTS.

((f_1,s_1[*units*])[,(f_2,s_2[*units*])][,...])

f

Specifies a 1- to 6-alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability and up to 94 fonts on a single page). If the number of fonts to be invoked exceeds the size of one DJDE record, multiple FONTS parameters must be used.

s

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; minimum value is 10 XDOTS or DOTS.

units

Available *units* are: lpi (lines per inch), XDOTS, and DOTS. If *units* is not specified, lpi is assumed by 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.

An XDOT is 1/600 unit of measurement. A form specifying XDOTS, however, can be created, edited, and compiled on any

LPS with version 4 software, but the form, which is 600 spots per inch (spi), will not print on your 4850 or 4890, 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 (DJD) encounters more than 127 fonts in a packet, the remaining fonts in the packet are ignored or truncated. Because the DJDE packet in this case is truncated, the outcome of the DJDE is unpredictable. Thus, the 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*

Options Refers to a separately cataloged file in the PDE library on disk. Parameters of the selected PDE, such as BEGIN, FONTS, and PMODE, may also be modified on an individual basis.

FORMS

Specifies the form to be merged on the printed pages. FORMS is a page-oriented DJDE.

Syntax FORMS = *options*

Options NONE

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

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 an FDL source file (FSL file).

(*form-id*[,*init* [,*copies*]][,(INKS, *inkref*$_1$ [,*inkref*$_2$][,...])])

This option has the following components:

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 an FDL source file (FSL file).

init

Specifies the beginning copy 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.

INKS
Indicates that the *inkrefs* that follow are to override the inks defined in the form.

inkref
Specifies the inks that override the corresponding inks specified in the ink list of the form. If the form does not contain an ink list (an existing form that has not been colorized using FCU), the form is printed in BLACK. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than the inks in the ink list of the file, the extra *inkrefs* are ignored.

Considerations

Example: FORMS=(FORM3,3,2,(INK,'RED'));

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.

GRAPHIC

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 last previous text record should have caused a page transition. If not, the occurrence of this DJDE parameter forces an immediate page termination, and the next text record encountered is processed relative to TOF on the next physical page.

For online processing, an interleaved graphic which immediately follows a header banner page is treated as document-interleaved. The header banner page must be defined by a BANNER command. Skip carriage control on a graphics sentinel DJDE is ignored for online processing.

GRAPHIC is a record-oriented DJDE.

Syntax

GRAPHIC = *options*

Options

(*name*,*vpos units*,*hpos units*[,H][,*n*][/*d*][,(INKS,*inkref*$_1$ [,*inkref*$_2$][,...])])

This option has the following components:

name
Identifies the graphic. If document interleaved, it becomes the filename with which the IMG file is written. If page interleaved, it is used to associate the graphic with other DJDE references. (If no other parameters are specified, the parentheses may be omitted.)

None of the remaining parameters may be used if the graphic is document interleaved nor should it normally be used if the graphic is referenced by an IMAGE DJDE. If page interleaved, the remaining parameters may be used exactly as on an IMAGE DJDE (except for the T parameter). If these parameters are specified in the GRAPHIC DJDE and also on one or more IMAGE DJDEs which reference the same graphic, the graphic is imaged on the page once for each reference.

vpos
Specifies the vertical position of the top edge of the graphic, relative to 0,0 on the current physical page.

hpos
Specifies the horizontal position of the left edge of the graphic relative to 0,0 on the current physical page.

units
Available options are: units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS. An XDOT is 1/600 unit of measurement, which is used for 600 spots per inch (spi) processing. Jobs with XDOTS can be created, edited, and compiled on any LPS with version 3 software, but will not print on your 4850 or 4890, which is 300 spi.

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.

INKS
Indicates that *inkrefs* follow.

inkref
Specifies the inks that override the corresponding inks specified in the ink list of the image file. Existing .IMG files are printed using the system default ink unless otherwise specified. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references.

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).

Example: GRAPHIC=(IMG1, 2, 1, (INKS, 'RED')), END;

ICATALOG

Specifies the ink catalog containing the palettes and inks for subsequent pages. ICATALOG is a page-oriented DJDE.

Syntax ICATALOG = *'inkcatalog-name'*

Options *'inkcatalog-name'*
Specifies the ink catalog to be used on the page.

Example ICATALOG='DFAULT'

IDFAULT

Specifies the default ink to be used for objects whose ink is not given. IDFAULT is a page-oriented DJDE.

Syntax IDFAULT = *inkref*

Options *inkref*
Represents the default ink to use in all objects that do not specify an ink. An example would be page numbers whose color is not specified in an OUTPUT command NUMBER parameter or in a DJDE NUMBER.

Considerations Example: IDFAULT = 'RED'

You can only use fully saturated (primary) colors in PDL.

IDR

Specifies the ink descriptor entry (IDR) for subsequent pages. The DJDE IDR parameter is not applied retroactively, but it becomes effective when it is actually invoked. IDR is a page-oriented DJDE.

Syntax IDR = *idr-name*

Options *idr-name*
Specifies the ink descriptor entry (IDR) to be used on the page.

Considerations Example: IDR=IDR1

When using the DJDE IDR, the IDR file must be defined globally. In other words, the *filename* .IDR must exist on the system disk.

ILIST

Specifies a list of all the inks which may be referenced by the ink index. ILIST is a page-oriented DJDE.

Syntax ILIST = *options*

Options *ink-name*
Specifies the ink name to be used on the page.

(*ink-name*$_1$,[*ink-name*$_2$][,...])
Specifies the ink names to be used on the page.

Example ILIST =('BLACK','XEROX..RED')

IMAGE

Defines the new imaging parameters for a graphic in batch mode. IMAGE is a record-oriented DJDE relating to graphics.

Syntax IMAGE = *options*

Options (*img-name*,*vpos units*,*hpos units* [,H][,T][,D][,*n*[/*d*]][,(INKS, *inkref*$_1$,[*inkref*$_2$][,...])])
This option has the following components:

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 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 specified by a PDE BEGIN of 0,0). Specify this parameter as a decimal number with up to three digits to the right of the decimal point.

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 specification is the same as for *vpos*.

units
Available options are: user-defined units (UN), centimeters (CM), inches (IN), or DOTS. 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.

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 on all subsequent pages until changed by a DJDE CANCEL or ALTER.

Do not use the H parameter and the D parameter together in this command.

T
: Specifies that the referenced graphic is page interleaved and follows the current page data in the input stream. This mode is limited to one graphic image per page.

D
: Specifies the disk-interleaved mode for handling graphics. Unlike the T mode (refer to the paragraph above), this mode is not limited to one graphic image per page.

Do not use the MOVE mode (specified by the GRAPHICS=MOVE statement) in conjunction with the D mode.

Do not use the GRAPHICS command to position or scale a graphic image when using the D mode; use the image command instead.

Do not use the D parameter and the H parameter together in this command.

n/d
: Specifies the reference scale factor.

INKS
: Indicates that *inkrefs* follow.

inkref
: Specifies the inks which override the corresponding inks defined in the ink list of the image file. Existing IMG files are printed using the system default black ink unless *inkrefs* are specified in the INK parameter. Restricted RES format IMG files are printed using the inks specified in the file unless *inkrefs* are specified in the INK parameter. If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

Considerations

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

Example: IMAGE=(IMG1, 3, 2, (INK, ‘RED’))

INKINDEX

Specifies a field within a data record which contains a number representing the ink index to be used for subsequent pages. INKINDEX is a page-oriented DJDE.

Syntax INKINDEX = *options*

Options *offset*
Indicates the byte offset in the data record where the ink index number is to be found.

(*offset* [ZERO|ONE] [*bit-opt*])
This option has the following components:

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

ZERO or ONE
ZERO: The first ink in the ink list is to be considered index zero.

ONE: The first ink in the ink list is to be considered index one.

bit-opt
Specifies how many bits are to be used in calculating the ink index. Allowable values for *bit-opt* are 1-7. The default is 4.

NONE
Specifies that there is no ink index.

Example INKINDEX=134

IRESULT

Specifies the result when objects imaged with different inks overlap. All jobs should be formatted assuming that the inks are opaque and all objects are imaged in the order in which they are specified. IRESULT is a page-oriented DJDE.

Syntax IRESULT = *options*

Options DEFAULT
Use the parameter specified in the system default IRESULT, which may be BLACK or COLOR.

BLACK
When two objects imaged with different inks overlap and black and highlight pixels coincide, the resulting pixel is black. This can be changed on a page basis.

COLOR
When two objects imaged with different inks overlap and black and highlight pixels coincide, the resulting pixel is highlight. This can be changed on a page basis.

Example IRESULT=BLACK

ITEXT

Specifies a text message to be displayed to operators during the input process. It takes effect on the next page boundary. ITEXT is a page-oriented DJDE.

Syntax ITEXT = *options*

Options NONE

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

(*sc*,*passnum*)
This option has the following components:

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

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.

JDE

Specifies the JDE to be used within the selected JDL at the next page boundary. JDE is a page-oriented DJDE.

Syntax JDE = *jde-id*

Options *jde-id*
Specifies the JDE to be used within the selected JDL at the next page boundary. In a DJDE packet containing a JDE, other record- or page-oriented DJDE parameters included in the packet override parameters specified by this selected JDE.

JDL

Specifies the name of the JDL to be invoked at the next page boundary. JDL is a page-oriented DJDE.

Syntax JDL = *jdl-id*

Options *jdl-id*
Specifies the name of the JDL to be invoked at the next page boundary. The *jdl-id* must exist on disk in the JDL directory. If only JDL is specified (not JDE) in the DJDE, the JDE to be used is the one named in the START command. In a DJDE packet containing a JDL parameter, other record- or page-oriented DJDE parameters override parameters specified by the JDL DJDE.

LOGO

Specifies the logos printed on a page or on all pages of a report. LOGO is a record-oriented DJDE. It has two sets of options.

Syntax LOGO = *options*

Options (*name*,*vpos units*,*hpos units*, [H],[(INKS, *inkindex*$_1$,[*inkindex*$_2$][,..])])

or

(*name*, HERE,[(VADJ,[+ I -] *tvalue*)],[(HADJ,[+ I -] *tvalue*)],[H],[(INKS, *inkindex*$_1$,[*inkindex*$_2$][,..])])

The first option has the following components:

name
Specifies the name of a logo to be imaged. A logo specified through PDL will appear on all pages of a report. A logo specified through DJDE will appear only on the current page unless the HOLD option is specified.

The named logo is assumed to be in the LGO file directory. If the JDE-specified logo is not found, the job aborts and an error message displays on the system controller.

The action taken will be that specified by the error option of the ABNORMAL PDL command.

vpos
Specifies the vertical position of the top edge of the logo, as an offset, relative to (0,0) on the current physical page (that is, the position that would be specified by a PDE BEGIN of (0,0)). The parameter and its unit must be separated by a space, otherwise a syntax error occurs.

hpos
Specifies the horizontal position of the left edge of the logo, as an offset, relative to (0,0) on the current physical page (that is, the upper left corner of the physical page as specified by the current page PMODE). The form of the specification is the same as for *vpos*.

units
Available options are: user-defined units (UN), centimeters (CM), inches (IN), 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.

H
Specifies that the DJDE-invoked logo should be Held and imaged without further callouts, at the same position on all subsequent pages until canceled by a DJDE CANCEL command. A subsequent change in PMODE will cause the positioning of any held logos to be recalculated.

INKS
Notifies the system that inkindex values follow.

inkindex

Specifies which inks in the ILIST parameter of the IDR command or DJDE ILIST override the corresponding inks specified in the inklist in the logo file.

In order for this option to be valid, the referenced logo must be in color format. Logos may be converted from black-only format to color format using the File Conversion Utility (FCU). Refer to your *Xerox 4850/4890 LPS System Programming and Administration Guide* for more information.

Existing LGO files and restricted RES format LGO files are printed using the system default black ink unless *inkrefs* are specified in the INK parameter.

If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

If any of the inks to be substituted were specified in the logo definition with a NOSUBSTITUTION parameter, as in the FCU command, the system processes the ink requests according to what you have specified in the ISUBSTITUTE parameter of the ABNORMAL command.

The second option has the following components:

name

Specifies the name of a logo to be imaged. A logo specified through PDL will appear on all pages of a report. A logo specified through DJDE will appear only on the current page unless the HOLD option is specified.

The named logo is assumed to be in the LGO file directory. If the JDE-specified logo is not found, the job will abort and an error message is displayed on the system controller.

The action taken will be that specified by the error option of the ABNORMAL PDL command.

HERE

Specifies the position of the upper left corner of the logo as the top of the next printed line, at the currently defined left margin (see the MARGIN parameter of the LINE command).

VADJ

Specifies an adjustment to the vertical position of the top edge of the logo relative to the current position defined by the HERE option.

+ or -

Specifies a positive or negative positioning of the *tvalue.* When following VADJ, the plus sign shifts the logo down and the minus sign shifts the logo up.

tvalue

Typed value. Specifies the positioning of a logo as a decimal number with up to three optional digits to the right of the decimal point. *tvalue* may be specified as IN, CM, UN, DOTS, or XDOTS.

HADJ
: Specifies an adjustment to the horizontal position of the left edge of the logo relative to the current position defined by the HERE option.

+ or -
: Specifies a positive or negative positioning of the *tvalue.* When following HADJ, the plus sign shifts the logo to the right and the minus sign shifts it to the left.

H
: Held specifies that the DJDE-invoked logo should be imaged without further callouts, at the same position on all subsequent pages until canceled by a DJDE CANCEL command. A subsequent change in PMODE will cause the positioning of any held logos to be recalculated.

INKS
: Notifies the system that ink index values follow.

inkindex
: Specifies which inks in the ILIST parameter of the IDR command or DJDE ILIST override the corresponding inks specified in the inklist in the logo file.

For this option to be valid, the referenced logo must be in color format. Logos may be converted from black-only format to color format using the File Conversion Utility (FCU). Refer to your *Xerox 4850/4890 LPS System Programming and Administration Guide* for more information.

Existing LGO files and restricted RES format LGO files are printed using the system default black ink unless *inkrefs* are specified in the INK parameter.

If one *inkref* in the list is omitted, commas must be used to maintain the relative positioning of the remaining ink references. If more *inkrefs* are specified than contained in the ink list of the file, the extra *inkrefs* are ignored.

If any of the inks to be substituted were specified in the logo definition with a NOSUBSTITUTION parameter, as in the FCU command, the system processes the ink requests according to what you have specified in the ISUBSTITUTE parameter of the ABNORMAL command.

Considerations

Any logos specified in a selected JDE replace those from the original JDE. If none are specified in the selected JDE, logos from the original JDE are canceled.

Do not code the HERE option with *vpos* and *hpos* values.

Include the HERE keyword whenever you code the VADJ and HADJ options.

MARGIN

Specifies the left printing margin within each logical page. It takes effect at the logical page following a DJDE END parameter. MARGIN is a page-oriented DJDE.

Syntax MARGIN = *options*

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. The default is 1.

(*value*, *value-type*)

This option has the following components:

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.

value-type

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 a *value-type* is character positions.

MODIFY

Specifies the copy modification entry (CME) to be used on the page. MODIFY is a page-oriented DJDE.

Syntax MODIFY = *options*

Options NONE

Specifies that data are processed without modification.

cme-id

Identifies a CME command. Any CME invoked by a DJDE must be coded within a JSL and cataloged separately. If the CME command is within the JSL, it must precede a reference to its identifier by the OUTPUT MODIFY statement.

(*cme-id,init*[,*copies*])

This option has the following components:

cme-id

The 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 OUTPUT command MODIFY parameter.

init

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

copies

Specifies the number of copies to apply the CME. If *copies* is not specified, then the CME applies to all copies beginning with the copy number specified by *init*.

NUMBER

Specifies page numbering control. It takes effect at the next logical physical page boundary. NUMBER is a page-oriented DJDE.

Syntax NUMBER = *options*

Options NO

Specifies that no page numbering is to be performed.

(*pnum*,*lnum*,*cnum*,[[*findex*],[*inkref*]])

This option has the following components:

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 goes 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.

inkref

Represents the ink to use for printing the page number string. If not specified, the current default ink is used.

Example NUMBER=(1, 1, 40, 1, 'RED')

OTEXT

Provides a text message that displays to the operator during job printing. OTEXT is a page-oriented DJDE.

Syntax OTEXT = *options*

Options NONE

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

sc

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

(*sc*[,*passnum*][,WAIT])

This option has the following components:

sc

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

passnum
Specifies the pass (copy ply) 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. 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.

(*sc*[,END][,WAIT])
This option has the following components:

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

END
Specifies that the text is displayed after the last copy of the report is printed. The maximum number of OTEXT characters which can be used in a report with GHO processing specified is 320 characters.

WAIT
Specifies that after the text is displayed, printing is suspended until the operator responds with a CONTINUE command.

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.

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

Options This option has the following components:

over-opt
The options are PRINT, IGNORE, MERGE, and PRINT2.

PRINT
Specifies that all overprint lines are printed as they would be on an impact printer, with the second line printed over the top of the first line with no regard to previous data, including character spacing, which may vary between the two lines of data.

IGNORE
Specifies that all overprint lines are ignored.

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

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. The number of overprint lines is always printed on the accounting page.

The default is NODISP.

PALETTE

Specifies the palette to be used for subsequent pages. PALETTE is a page-oriented DJDE.

Syntax PALETTE = *'palette-name'*

Options *'palette-name'*
Specifies the palette to be used on the page. Note that the palette name is specified within single quotes.

Example PALETTE = 'SIMPLE'

PMODE

Specifies the printing mode for each page. PMODE is a page-oriented DJDE.

Syntax PMODE = *options*

Options LANDSCAPE
Printing is parallel to the long edge of paper.

PORTRAIT
Printing is parallel to the narrow edge of paper.

Considerations If PMODE is changed via a DJDE command, the BEGIN command should be specified again (even if the same horizontal and vertical position was 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 = *options*

Options NONE
Specifies that no form is to be printed.

form-id
If the name of a file is 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 which is being defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file.

Example RFORM = (GBAR,(INK,'RED'))

RTEXT

Specifies text to be printed on a separate page preceding a report. RTEXT is a page-oriented DJDE. The cataloged RTEXT file is stored on the system disk as ROUTE1.TST.

Syntax RTEXT = *options*

Options NONE

Specifies no text is to be printed on a separate page.

sc

Specifies the message to be printed (1 to 132 characters). It is printed with the first font specified in the FONTS parameter of the PDE command (refer to fontindex).

(*sc*[,[*passnum* I ALL][,*line*[.*column*[,*fontindex*]]]])

This option has the following components:

sc

Refer to the *sc* option above.

passnum or ALL

An integer number or the keyword ALL. An integer number specifies the pass (copy number) to which the text applies.

line

Specifies the line number on which the first line of a block of RTEXT message is printed. The default is line 1 for the first text string of the pass. Otherwise, the default is the next line of the page. Only one RTEXT parameter may be specified per line for a *passnum*.

column

Specifies the column number at which the first character of a block of RTEXT messages is to be printed (default is column 1).

fontindex

Identifies the index (starting with 1) of the PDE font with which the text is printed.

rtext-id

Used to reference 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.

Example RTEXT = ('TEST',ALL,15,5,2,'RED');

SAVE

Specifies that IMG files are updated during the current report and are not purged when the report has completed printing. This command must occur prior to or at the end of the report. SAVE is a record-oriented DJDE relating to graphics.

Syntax SAVE = *options*

Options (*filename*[,...])

Specifies the name of an IMG file or a filename mask whose form and interpretation are as described for the operator SAMPLE command.

ALL

Specifies that all document interleaved files created by this report processing are to be saved, overriding an explicit (or default) specification in the job's OUTPUT command.

SF1FUNCTION

Provides for control over third-party finishing devices which conform to the DFA standard. Using this command, you can invoke sheet finisher function number 1 through the DFA channel C6.

Syntax SF1FUNCTION = {NO|YES}

Options NO

Does not invoke the sheet finisher function number 1.

YES

Does invoke the sheet finisher function number 1.

SF2FUNCTION

Provides for control over third-party finishing devices which conform to the DFA standard. Using this command, you can invoke sheet finisher function number 2 through the DFA channel C7.

Syntax SF2FUNCTION = {NO|YES}

Options NO

Does not invoke the sheet finisher function number 2.

YES

Does invoke the sheet finisher function number 2.

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.

Syntax

SEPARATORS = {FIRST|LAST|(LAST,D)|BOTH|(BOTH,D)|NONE}

Options

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
: 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.

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.

Considerations

If segment management is disabled using the SPLIT=OFF DJDE, SEPARATORS defaults to NONE.

SHIFT

Specifies a shift of image on the page for binding purposes. SHIFT is a page-oriented DJDE.

Syntax

SHIFT = *options*

Options

NO

Specifies that no shift occurs (v_1=0 and v_2=0).

YES

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

(v_1, v_2)

This option has the following components:

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.

v2

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.

Considerations

An LPS configured with 600 spi automatically doubles dot values to xdots for positioning.

SIDE

Specifies the repositioning of 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*)

Options

This option has the following components:

side-opt

Available *side-opt* options are: NUFRONT, BACK, NUBACK, NEXT.

In duplex (with no BFORM), *side-opt* produces the following results:

- If NUFRONT is coded, the specified logical page is positioned as the first logical page on a new sheet. If BACK is coded, the specified logical page is positioned as the first logical page of the next available back.
- If NUBACK is coded, the specified logical page is positioned as the first logical page on the back of a new sheet.
- If NEXT is coded, the specified logical page is positioned as the first logical page on the next available side, the back of the current sheet or the front of the next sheet.

The logical page is moved unless it is already properly positioned as the first logical page on the specified side.

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.

offset-opt

Available *offset-opt* parameters are: NOFFSET, OFFSET. If OFFSET is coded, the sheet on which the repositioned logical page occurs is offset in the output stacker. If NOFFSET is coded, no offset occurs.

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 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 = (*lnum*, *cnum* [, *findex*])

Options

lnum
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 EXPORT SPLIT = {NOW|(*min*, *max*)|OFF}

Options

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 reenable 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.

Syntax EXPORT SRECOVER = (PAGE|SEGMENT|ASK|DEVICE)

Options

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 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.

Considerations 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.

STIMING

Provided for possible compatibility with other Xerox software supporting third party finishing devices and is ignored by the Xerox 4850/4890 LPS. Device specific timing parameters are actually set by using the Finishing Configuration (FCG) Utility.

The STIMING parameter, although ignored by the Xerox 4850/4890 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 STIMING = ([INTERVAL,*sec*][,DELAY,*sec*])

Options 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 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.

STOCKS

Redefines the parameters of a STOCKSET command (the types of paper to use) for the page. STOCKS is a page-oriented DJDE.

Syntax STOCKS =*stockset-name*

Options *stockset-name*
This option defines the STOCKSET and the associated stock to be used in a report. If the *stockset-name* does not refer to a STOCKSET command coded earlier in the same JDL, it is assumed that the STOCKSET exists globally as an STK file, which is read at print time. In this case, PDL inserts a message in the JSL file listing, indicating that an STK file will be used at print time.

Whenever a new stockset is chosen, that is, at 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.

Syntax TOF = *value*

Options *value*

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

The default is 1.

Considerations 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.

XMP

Controls the xerographic mode switching used for a report. XMP is a record-oriented DJDE.

Syntax XMP = *options*

Options DEFAULT

Specifies that the sysgened XMP is used for the report.

REPORT

Specifies that when the system determines that a report requires highlight color mode for any page of the report, all remaining unprinted pages of the report are printed in the highlight color mode.

Example XMP=REPORT

6. Highlight color printing

Optimizing performance with xerographic mode persistence (XMP)

You can use xerographic mode persistence (XMP) to optimize performance in several ways:

- Reducing dry ink consumption. Avoid printing monochrome jobs in the highlight color mode, since some nonimaged toner is consumed.
- Maintaining high throughput (rate of pages printed per minute). The time required to switch modes depends on the condition of the color developer housing. It ranges from 15 seconds to five minutes. Print speed is reduced by frequent mode switching.

Mode switching results in longer printing interruptions when print quality adjustment (PQA) is needed.

Xerographic mode

Xerographic mode is the current print mode. There are two xerographic modes for the 4850 and 4890 LPS:

- Highlight. The laser printing system (LPS) uses both color and black dry inks. This mode can be used for printing a single color on a page or for printing tones.
- Monochrome black. The LPS uses only black dry ink.

The 4850/4890 LPS operating system software (OSS) sets the xerographic mode by evaluating the demands of the printing job. The system looks at the ILIST (ink list) for the current job and determines which primary colors are needed and determines the current xerographic mode persistence (XMP) setting.

Xerographic mode persistence (XMP)

You can use xerographic mode persistence (XMP) to control some of the performance factors for the print jobs you code. Although the 4850/4890 LPS chooses the xerographic mode required for the pages in a report, you can set XMP to control how long the specified xerographic mode remains in effect. The minimum page number for XMP is 50. In other words, if you set the XMP to specify at least 60 pages, the controller remains in highlight color mode for 60 pages even if pages requiring only black are printed.

You can set XMP in one of the three methods listed below in increasing priority:

- Sysgen parameters. The sysgen parameter you enter becomes the default for the conditions when xerographic mode switching is valid. If a job has no PDL or DJDE xerographic mode switching parameters, the mode set during sysgen prevails.
- OUTPUT command. You may change the XMP setting for a particular report with the OUTPUT command, specifying either the default value or the entire report.
- DJDE. You may change the XMP setting for certain copies of a report using the XMP in DJDE, also specifying either the default value or the entire report.

Pages containing color are printed in highlight color mode, however, black pages may be printed in either monochrome (black) mode or highlight color mode. XMP has a minimum of 50 pages. That is, up to 50 monochrome pages requiring only black may be printed in highlight color mode.

XMP examples

Two XMP scenarios are described below:

- The first 50 pages of a 150-page report call for highlight color, while the second 50 pages specify monochrome black. If you specify XMP=DEFAULT in your PDL command or DJDE, the highlight color mode operates for the first 100 pages, and then the sysgen specified XMP= pages. If no new highlight color page is encountered, the last 50 pages print in monochrome black. If you specify XMP REPORT, the entire 150 page report prints in highlight color.
- A job prints a cover sheet in highlight color followed by several hundred pages in monochrome black. You can increase throughput by switching the job to monochrome mode after the cover is printed. However, if the report has chapter headings that require highlight color on every tenth page, for example, you achieve better throughput with consistent print quality by leaving the printer in highlight color mode.

Ink substitution

Ink substitution is the use of a primary color other than the one required by inks referenced in your job. Ink substitution can occur during input processing as the result of actions specified in the ISUBSTITUTE parameter of the ABNORMAL command or during output processing as the result of an operator-initiated SUBSTITUTE INK command.

Refer to the ”ABNORMAL command” section in the “Print format commands” chapter for information on the ERROR parameter. Refer to your *Xerox 4850 HighLight Color LPS Operator Guide* or *Xerox 4890 HighLight Color LPS Operator Guide* for information on the SUBSTITUTE INK operator command.

4850/4890 inks

It is helpful to examine the palette samples and experiment by printing inks from the different palettes. Three sets of printed palettes are included with your 4850 or 4890 reference set and you can print more sets yourself by sampling the forms that contain the palettes.

To print the available palettes and inks contained in the XEROX1, DFAULT and XEROX ink catalogs, specify the appropriate forms in the SAMPLE command. Refer to the *4850/4890 Highlight Color LPS Palettes* for a list of the palettes currently available.

To experiment you can print the same form with filled boxes several times, using different defined inks each time. Seeing the printed colors should help you choose among the inks.

Coding for efficient queue management

Queue management allows you to schedule printing on a report-by-report basis. You can select receive mode, in which reports are printed on a first-come-first-print basis or resource mode, which sequences printing according to specified color and stitching attributes:

- Primary color
- Multiple primaries
- Stitching.

Refer to the *4850/4890 Highlight Color LPS Palettes* appendix at the end of this document for a list of primary colors currently available.

Resource mode

Resource mode helps you minimize the number of times you must change the color toner housing in the printer in the course of running a series of print jobs. In resource mode, reports whose color specifications do not match the currently loaded color housing are held in queue while reports matching the current color housing are printed. There are two resource modes: single and multiple.

Single mode

In single mode, reports that specify the highlight color currently loaded in the printer are processed. All other single-color reports are held, as are reports containing multiple primaries.

Multiple mode

In multiple mode, all single-color reports are held. When all multiple mode reports are printed, the single reports that match the loaded primary are printed, while all others remain held.

Guidelines for efficient coding in resource mode

To facilitate the most efficient use of the printer when in resource mode, there are several steps you can take, involving the IDR command ILIST parameter, the DJDE ILIST, and the ABNORMAL command IMISMATCH and ISUBSTITUTE parameters.

Remember that all inks must be in place by page one of the customer data, especially in DJDEs, so that the system receives and schedules color information for the entire report according to all of the color specifications. Other steps are described in the following sections.

IDR command ILIST parameter

- Make sure the ILIST is accurate and complete. It should contain all the primary inks called out in the reports within the job.

  Missing or incorrect ILIST definitions can produce different results from one print run to the next, depending on when in the report processing cycle formatting and printing occur. This depends on the size of the report, the number of reports in queue, and other factors. When the entire report is formatted before printing begins, the report can be scheduled using accurate resource requirements. When only a portion of the report has been formatted, scheduling based on color requirements may be based on incorrect or missing ink information, and the result may be different than expected. The differences are those permitted by the ABNORMAL command IMISMATCH and ISUBSTITUTE parameters you have specified.

- Avoid using generic ink lists, such as the following, because the system can only determine the actual primary color requirements by processing the report:

  ```
  GENIDR: IDR   ILIST = ('BLACK', 'RED', 'GREEN', 'BLUE'
                        'MAGENTA', 'CYAN');
  ```

  Using a generic ink list can delay the printing of large reports. When in single mode and no ink substitutions are in effect, a report using the currently loaded color primary is not scheduled to print until input processing is completed for the entire report. This can cause considerable delays for large reports. There is no delay in scheduling reports to print when ink substitutions are in effect, such that all the inks listed can be printed with the currently loaded primaries.

- Keep in mind that if you do not code an IDR command in your JSL or if you do not include an IDR parameter in an OUTPUT command, the system supplies DFIDR.IDR as the default IDR, which specifies only black ink.

- If you code the VOLUME command CODE parameter as CODE=NONE, an IDR command ILIST parameter that specifies a highlight color primary causes the entire report to print in highlight color. If ILIST specifies only black, the entire report prints in black ink.

DJDE ILIST

Take great care in using the DJDE ILIST for your print jobs. Assume that you start a large monochrome black print job and

insert an ILIST DJDE into the data stream to print the last page of a report in red. If input is completed before the job is passed to output for printing, the system recognizes the DJDE ILIST specification for red and the report is held in the queue if red is not loaded. If, however, the report is sent to output for printing before it processes the last page, the system will not process and queue the specification for red and the entire report will print in black, depending on the IMISMATCH and ISUBSTITUTE parameter options of the ABNORMAL command you have specified for the report. Refer to the "ABNORMAL command" section of the "Print format commands" chapter for additional information on ABNORMAL command parameters.

ABNORMAL command IMISMATCH parameter

- Within resource mode, there are two types of scheduling: single and multiple. With single scheduling, only reports containing one highlight primary color are scheduled for printing. Reports specifying a different primary color or multiple primaries are held in queue. You or an operator can remove these reports from hold in one of two ways: changing the color housing in the printer, or entering a SUBSTITUTE INK command if you have selected the STOP option for the IMISMATCH parameter of the ABNORMAL command. They cannot be aborted. For this reason, you want to code only the STOP option for single report scheduling.
- In multiple report mode, only reports requiring more than one highlight primary are scheduled. When all multiple primary reports have been scheduled and no more multiple reports are being input, the system schedules the single primary reports that match the currently loaded color housing in the printer.

ABNORMAL command ISUBSTITUTE parameter

If you are printing reports in single mode, code ABNORMAL ISUBSTITUTE = ANY so that operators can remove reports from hold by entering SUBSTITUTE INK commands.

Excessive primaries specified for one page

Input processing does not detect that the highlight color capabilities of the printer (one highlight color plus black per page) have been exceeded until the time that printing two different primaries is actually attempted. This may occur at the beginning, in the middle, or at the end of the page, depending on the feature that causes this condition. The page causing this error may be skipped, partially printed, or printed completely with black substituted for the extraneous color. Following this page, the system generates an error page indicating that the error has occurred. This error page may also contain DJDE information which may or may not be related to the excessive primary condition.

Stitching

If your 4850/4890 does not have stitching capability, the Queue Manager ignores the stitching requirement when determining report scheduling.

If your 4850/4890 has stitching capability but it is unavailable, reports requiring stitching are held regardless of their highlight

color requirements unless you enter an operator STITCH OFF command.

Refer to your operator guide for information on the STITCH command and other stitch-related topics.

Referencing inks

In all PDL commands and DJDEs that contain ink parameters and options, you need to specify the inknames you wish to use, along with the ink catalog and palette that contains the ink. Ink referencing is done by either naming a specific ink or an inkindex that specifies the sequence number of the ink within an ILIST parameter. Refer to the “IDR command” section in the “Print format commands” chapter for information on the ILIST parameter.

Ink catalogs

There are four ink catalogs resident on your 4850/4890 version 5.0 software:

- XEROX1
- XEROX
- DFAULT
- SDFLT$.

The XEROX1 ink catalog contains both long names and alphanumeric names. XEROX1 is different than XEROX and DFAULT in that it uses a generic alphanumeric color value naming convention that can be applied across palettes. The XEROX1 catalog has no DFAULT palette, so you must specify one for your jobs.

The DFAULT and XEROX ink catalogs contain both long names, such as ‘VIVID RED’, and alphanumeric names, such as R41K4. Alphanumeric names represent the percent of highlight color (red in this instance) and the percentage of black in the ink. DFAULT is the default catalog for PDL and FDL. The XEROX ink catalog is the default catalog for Interpress jobs.

The SDFLT$ catalog is the system default and cannot be changed or referenced. It prints black only.

Inkname elements

An inkname contains three elements:

catalog-name
: Name of the ink catalog from which the palettes are being taken. If you have not specified a different ink catalog, the system uses the default catalog you specify in the ICATALOG parameter, the IDR command, or the DJDE ICATALOG. If you do not specify an ink catalog, the system uses the DFAULT ink catalog for PDL and FDL. Single quotes are optional for ink catalog names. For example, both 'XEROX1' and XEROX1 are correct.

palette-name
: Name of a palette in the ink catalog you have specified. It contains the inknames for the report. If you have not specified a palette, the system uses the default palette in the PALETTE parameter of the IDR command or the DJDE PALETTE. If you have not set up a default palette, the system uses the DFAULT palette. You must enclose palette names in single quotes, for example, 'SIMPLE' and 'RED' are correct.

inkname
: Name of an ink defined in the ink catalog and palette. It represents a specific tone, tint, or shade of the primary color you have selected. Inkname elements may contain an alphanumeric name, such as 'H38K21', or a long name consisting of one or more words separated by spaces, such as 'LIGHT GRAYISH RED'. You must enclose *inknames* in single quotes, as shown in the inknames above.

The three elements of an inkname are separated by periods, so that the following formats are possible:

- '*inkname'*
- *'palette-name.inkname'*
- *'ink-catalog-name.palette-name.inkname'*
- *'ink-catalog-name..inkname'.*

Printing tints and shades

The system prints all text in solid primary color. The system prints a tint (a color mixed with white) as the solid of the primary color. A gray, for example, prints as black. The system prints a shade (a color mixed with black) as the solid of the primary color with the greatest concentration in the specified ink.

Specifying inks

Ink referencing in PDL commands and DJDEs is specifying an *inkref* parameter option. The *inkref* option calls out the specific ink to be used on the page. It is defined by either an inkindex (value) or an inkname.

Inkindex

An inkindex is a numeric value that represents an index into the current ink list (IDR command). It is a value that selects an ink listed in the current ink list specified in the ILIST parameter of the IDR or DJDE command. Its value may range from 1 to *n*. (*n* is the number of inks that are defined in the current ILIST command parameter.)

For example, assume that you have this IDR command in your JSL:

```
IDR1:  IDR   ILIST = ('BLACK', 'RED', 'BLUE', MAGENTA',
'CYAN');
```

In your job, you then want to override the inks specified in the ink list of the image and page number that are called out for the job. Given that red is called out for the image and black for the page number, you could override each of these in the IMAGE and NUMBER parameters of the OUTPUT command as follows:

```
JOB1:  JOB;
         OUTPUT    IMAGE = (1 IN, 2 CM, 1),
                   NUMBER = (1, 1, 40, 1, 2);
```

1 ('BLACK') and 2 ('RED') are referenced from the IDR command ILIST parameter specified at the ID command level in the JSL. Refer to the "JSL structure" section in the "Overview" chapter for information on command levels. Refer to the "IDR command" section in the "Print format commands" chapter for information on the ILIST parameter.

Inknames

You may specify either long names or alphanumeric names in your JSLs, depending on the palette you have selected. The XEROX1 catalog differs from the DFAULT and XEROX catalogs.

The XEROX1 ink catalog allows you to specify one inkname and use it across XEROX1 palettes to achieve equivalent results across primaries. Inks are represented by the percent of highlight color and the percent of black, regardless of what the color is. For example, H10K20 yields the same tone of red as it does for blue when it is specified in the RED and BLUE XEROX1 palettes, respectively.

The XEROX1 catalog HK inknames cannot be referenced in other ink catalogs. Using the HK inknames in the XEROX1 ink catalog makes ink referencing easier in several ways:

- Inknames are easy to remember, especially when you are using several different palettes.
- You can easily change color specifications in the print job by sending a DJDE PALETTE through the input data stream. This allows you an easy method of ink substitution and takes the guesswork out of finding equivalent color attributes across the XEROX1 palettes. By using the HK convention, the percentage of highlight color and the percentage of black are consistent throughout these palettes.

In the DFAULT and XEROX ink catalogs, the alphanumeric inknames identify whether the shade has more black than the highlight color by percent. For example, in the XEROX ink catalog, the shade G12K67 is 12 percent green and 67 percent black and thus prints as black. The long names, on the other

hand, reflect what the color looks like rather than how it is created. Thus, long names also identify whether the shade has more black than the highlight color by description, rather than by percent. For example, the shade 'GRAYISH RED' tells you that it consists of mostly red with some gray, whereas 'REDDISH GRAY' is comprised of gray with a little red. Therefore, text is printed on the page in black rather than red.

As in the inkindex example, you want to override the inks specified in the ink list of the image and page number that are called out for the job. Given that red is called out for the image and black for the page number, you can override each of these in the IMAGE and NUMBER parameters of the OUTPUT command as follows:

```
JOB1:    JOB;
         OUTPUT    IMAGE = (1 IN, 2 CM, (INK, 'BLACK')),
                   NUMBER = (1, 1, 40, 1, (INK, 'RED'));
```

Inknames must be referenced from the IDR command ILIST parameter that is typically specified at the ID command level in the JSL. In this example, both 'BLACK' and 'RED' are listed in the ILIST parameter of the IDR command.

Adding highlight color to existing applications

There are two primary ways of adding highlight color to existing applications: by modifying applications or by modifying JSLs.

The easiest method of adding color is through modifying the JSL for the application:

- Add copy modification entries (CMEs) to do the following:
  - Change the color of text within a rectangular area of a page using the INK parameter, similar to a font change using CMEs. Refer to the "CME command" section in the "Print format commands" chapter for information on the CME command.
  - Print constant and/or colored text on specific copies or on all copies, using the CONSTANT and INK parameters. The text need not be color since INK is optional.
- Print selected IMG graphics in color by converting them to monochrome image format. Refer to your *Xerox 4850/4980 HighLight Color LPS System Programming and Administration Guide* for information on converting files to color format.
- Print selected forms or form elements in color by modifying the forms source library (FSL) of the form and recompiling it. Refer to your *Xerox 4850/4890 HighLight Color LPS Forms Creation Reference* for information on adding highlight color to forms.
- If the application currently uses a font index byte to change fonts, you can use that byte as an inkindex to change text color when the font is changed.
- Use the LMODIFY command to selectively print parts of individual print lines using a different ink. For more information on the LMODIFY command, refer to the "Logical processing commands" chapter.

If you wish to modify the application, you have the following additional capabilities:

- Modify the application to send a separate inkindex byte with each record to specify the highlight color used to print that record.
- Add dynamic job descriptor entries (DJDEs) to do the following:
  - Change the color of text using the DJDE ILIST or IDFAULT command,
  - Print IMG graphics in color without converting them to monochrome image format using the INK override option of the DJDE image (or GRAPHIC) command, or
  - Print forms in color without modifying the FSL of the form with INK overrides in the DJDE FORMS command. This is also possible using the OUTPUT FORMS command in the JSL.

Print line modification features

There are three PDL/DJDE commands that provide alternative means of print line modification:

- MODIFY;
- OVERPRINT=MERGE applied in conjunction with the FONTINDEX and, optionally, the INKINDEX command; and
- PDL LMODIFY.

The MODIFY command is used to specify a copy modification entry (CME) which can be designed to selectively change the FONT, INK, or text (CONSTANT) in a print line on a copy basis.

The FONTINDEX (and optionally INKINDEX) command when applied in conjunction with the OVERPRINT=MERGE command is used to specify the fonts and inks to be used to print the merged text of overprinted lines.

A third alternative means of print line modification, the PDL LMODIFY statement allows the user to specify the conditions for changing the ink and to select either portions or the entire print line to be printed with a specified ink.

These print line modification features operate exclusively. If two or all three of these features are concurrently applied on a page, the resultant print lines will exhibit the modifications called for by only one of these features. The other applied features will remain suspended until the superseding feature is terminated.

Since the MODIFY and FONTINDEX (and optionally INKINDEX) with OVERPRINT=MERGE commands can be specified as page-oriented DJDEs which are applied at logical page boundaries, these print line modification features can be instituted and terminated at logical page boundaries.

The LMODIFY statement may be specified only within a JDE. Thus, this mode of print line modification may be initiated or terminated via a DJDE selected JDE and/or JDL only at physical page boundaries because the application of DJDE selected JDE and/or JDLs force a transition to a new physical page.

The precedence or hierarchy of application of these features is:

(1) FONTINDEX (and optionally INKINDEX) with OVERPRINT=MERGE,

(2) CMEs, and

(3) LMODIFY.

Highlight color coding tips

There are many ways in which the various highlight color-related PDL commands and DJDEs can help you create highly attractive and effective applications. Keep the following guidelines to keep in mind when creating your JSLs:

- Accurate inknames: Make sure you include the inkname you want to use in addition to the ink catalog and palette. Generally, inkindexes are faster and easier to use in your jobs than inknames.

  If you encounter an inkname error, check to make sure you have used the correct inkname type for the palette you have specified. For example, the 'DFAULT.DFAULT' palette uses only long names, such as 'LIGHT GRAYISH RED'. If you have specified 'R25K18' as an inkname for this palette, you will get an error. Conversely, the 'DFAULT.RED' palette uses only alphanumeric names, so if you have specified the inkname 'LIGHT PINK' for it, you will also get an error.

- Specifying default palettes: As with inknames, specify a default palette. This is particularly important with the XEROX1 catalog because it uses a generic ink naming convention and it has no DFAULT palette. You get an error message if you do not specify a palette for the XEROX1 ink catalog.

- Applications: Color documents add a new dimension to the way in which you develop applications. When adding highlight color to a monochrome black application, focus on what is important information and how the highlight color ink could enhance that information.

- Logos: When your application includes a "colorized" form, the current logos also need to be updated to include a color ink list in the logo header. This process can be done through the file conversion utility (FCU) provided on the printer. If you want to use current black format logos in color forms, you still need to upgrade the logo to the 4850 or 4890 color format. FCU includes a color ink list in the logo header file. Refer to your *Xerox 4850/4890 HighLight Color LPS System Programming and Administration Guide* for information on FCU.

- Primary colors: In PDL, only solid primary colors (e.g., black, red, blue, green, magenta or cyan) may be incorporated in any part of a form or variable data. Shading of boxed areas can be done in the forms description language (FDL). Refer to your *Xerox 4850/4890 HighLight Color LPS Forms Creation Reference* for information.

- Single quotation marks: All palette and inknames must be enclosed in single quotation marks (''). Some ink color names have embedded space within their names, such as 'VIVID GREEN'. Single quotation marks are used as delimiters when specifying inknames (for example, 'XEROX.DFAULT.VIVID GREEN'). Single quotation marks are optional for ink catalog names.

- Ink catalogs: The default ink catalog for PDL (and FDL) is DFAULT. This catalog is recommended for all PDL (and FDL)

applications. The default ink catalog for Interpress is Xerox.ICT. This ink catalog should be used for all Interpress applications.

When creating your print jobs, however, you can use any or all of the DFAULT, XEROX, or XEROX1 ink catalogs. In your JSLs, you can list several IDR identifier commands that specify different ink catalogs and then call out whichever one you wish to use for a particular job at the job command level. Refer to the "JSL structure" section in the "Overview" chapter for information on command levels.

- Changing commands: If you change a command to incorporate color in your job and text disappears, check the IRESULT parameter of the OUTPUT command.
- Variable text: If a line of variable text is not printing as expected, check the IDFAULT parameter of the OUTPUT command. Also, when using the INKINDEX parameter of the LINE command, you must specify an inkindex on every line requiring highlight color.

7. Graphics

This chapter describes the various formats for graphics processing.

Processing modes

The software handles graphics 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 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 further information, refer to the “Page interleaved” section of 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, 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 is invoked.

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 *filename* 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 indicated below.

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 | FB | V | VB

It may not be an undefined block or record format:

RECORD STRUCTURE = U | 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 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 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 indicated below.

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 | FB | V | VB

It may not be an undefined block or record format:

RECORD STRUCTURE = U | 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 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 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 indicated below.

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 | FB | V | VB

It may not be an undefined block or record format:

RECORD STRUCTURE = U | 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 format on your laser printing system:

- Noninterleaved
- Document interleaved
- Page interleaved
- Batch mode.

Noninterleaved

HOSTCOPY is the mechanism for transferring noninterleaved graphics through the online channel to disk. The HOSTCOPY utility supports .IMG files in Xerox system tape format.

Document and page interleaved

Online report processing, which includes document and page interleaved graphics, functions as tape input processing functions.

The rules for online interleaved graphics are indicated below.

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 nonoptimize 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 allow 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 available for offline use also.

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 graphics may be used in current job.

IMAGE
: Specifies initial scaling and positioning parameters for batch mode.

PURGE
: Specifies default disposition of document interleaved graphics.

UNITS
: Specifies user-selected units for positioning graphics.

DJDE definitions for graphics

ALTER
: Specifies new imaging parameters for graphics previously referenced using HOLD statement.

BATCH
: Marks beginning and end of online batch mode graphic data.

CANCEL
: Terminates HOLD statement.

GRAPHICS
: Precedes interleaved graphic in data stream.

IMAGE
: Specifies name, scaling, and positioning of a graphic printed.

SAVE
: Identifies document interleaved graphics not to be deleted (purged).

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 feature 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 | FB | V | 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 for OSS software version 5.0 is 32. 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 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 that is being printed.
- Forms are not supported for batch mode processing.

8. Paper stock and clusters

Cluster printing allows the 4850 or 4890 to print jobs continuously between all four input paper trays. The system also provides added error checking to ensure the jobs are being printed on the correct paper stock.

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 4850 and 4890 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 4850 and 4890 LPS to determine what size paper to use to a print job.

Programming tasks

As the programmer, you group any combination of trays into a cluster. Group the trays together because they hold the same stock or they hold stock 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:

- Clusters needed by the report (that is, the “stockset”)
- Particular cluster to use to print each particular page
- Recommended trays for loading the stock.

You store the defined clusters in the CLUSTR.LIB file using the cluster (CLP or CLU) commands. CLUSTER.LIB, the cluster database for the printer, 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.

Paper stock commands and DJDEs

You use the PDL commands and parameters listed in table 8-1 to specify paper stock.

Table 8-1. **PDL commands specifying paper stock**

Commands	Parameters	Options
OUTPUT	FEED =	AUX MAIN OPR *stock-name* *stock-reference*
	STOCKS =	*stockset-name*
STOCKSET	ASSIGN =	*stock-descriptor* (*stock-descriptor*$_1$[,*stock-descriptor*$_2$][,...[)
	INIFEED =	*stock-name* *stock-reference*
	SYSPAGE =	*stock-name* *stock-reference*

You use the DJDEs listed in table 8-2 to modify paper stock specifications.

Table 8-2. **DJDEs modifying paper stock**

DJDE	Options
FEED =	AUX MAIN OPR *stock-name* *stock-reference*
STOCKS =	*stockset-name*

Cluster guidelines

When creating clusters, follow the guidelines in this section.

Cluster names and references

Assign each cluster a unique logical name that begins with a letter and consists of 1 to 6 alphanumeric characters. Make certain that the cluster name is not a keyword, parameter, or option used by the LPS. Group the clusters used by each application into a stockset.

Stocksets convert cluster references to cluster names. Cluster references are recommended but are not required. The cluster name appears first in quotes, followed by its cluster reference. The cluster is identified to the operator by cluster name. Define stocksets with a maximum of one cluster for each feeder tray on the LPS since the printer cycles 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.

Specifying paper stocksets

A stockset is a set of paper clusters used in a report. To create a stockset, determine the different types of paper stock required by the job. For example, to create a billing application, you need:

- Cover letter stock
- Standard billing stock
- Late notice stock
- Disconnection notice stock.

Your job description looks like this:

```
BILLS: STOCKSET ASSIGN=('CV102', COVER),
                      ('CL231', BILPAG),
                      ('LT329', LATE),
                      ('DS999', DISCON);
```

In this case, 'CV102', 'BL231', 'LT329', and 'DS999' are the actual names of the paper stock used during printing. COVER, BILPAG, LATE, and DISCON refer to the stock names. You can create a JSL and encode it in the stockset within a compiled JDL. You may also compile the stockset separately, similar to a PDE, which produces an STK file that may be used globally by other JDLs.

For the application to use the stockset, use the OUTPUT command:

OUTPUT STOCKS = BILLS;

You may also call out the stockset in the data stream using the following DJDE command:

DJDE STOCKS = BILLS, END;

When the stockset is created and referenced, you can use the FEED command of the OUTPUT statement to specify which cluster to feed. A DJDE FEED may be embedded in the data stream to change the active cluster dynamically.

For example, the data might look like the following:

```
DJDE FEED = COVER, END;
...data for cover letter...
DJDE FEED = BILPAG, END;
...data for actual bill...
```

Each physical sheet in a job has a cluster associated with it. This is true even if no stockset in an OUTPUT FEED command is in effect, since the 4890 LPS defaults to cluster main.

Using clusters in print jobs

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

- 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, then it uses the default stock for the site. If the application is offline, and the operator knows from the START command which stock to load, the operator loads the stock. Simple applications are fed stock from the MAIN and AUX clusters. MAIN and AUX are defined by the programmer to improve printer performance for the jobs run by the site, ensuring that paper can be loaded during printing.

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 page called.

Stockset changes require the cluster checks listed in table 8-3.

Table 8-3. **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 an “in use” status. That is, it has trays and is in the stockset last called by the job. Any one of these errors in any clusters in the stockset gives “in error” status to the stockset.

As jobs print, the output task uses the 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:

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

Mixing applications

When a site runs simple and OTEXT applications with stockset applications, the LPS executes two additional stockset checks to help ensure the correct paper is loaded. They are as follows:

- If a report has no stockset information, 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 and DJDE command summary

This appendix lists all PDL commands and DJDEs. 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, 4650, 4850, and 4890)

The symbol *ac* is used to represent command identifiers The *ac* identifier consists of one to six alphanumeric characters (A to Z, 0 to 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 and DJDEs**

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
ABNORMAL	ERROR	STOP	All	Y	Y	—
	IMISMATCH	STOP	4850, 4890	Y	Y	—
	ISUBSTITUTE	ANY	4850, 4890	Y	Y	—
	OTEXT	NOWAIT	All	Y	Y	—
	SECURITY	NO	All	Y	Y	—
ACCT	DEPT	*jdl-name*	All	Y	Y	Page
	USER	BIN	All	Y	Y	—
BANNER	HCOUNT	0	All	N	Y	—
	HJOBNO	NONE	All	N	Y	—
	HRPTNA	NONE	All	Y	Y	—
	TCOUNT	0	All	N	Y	—
	TEST	—	All	N	Y	—
	TYPE	BANNER	All	N	Y	—
BDELETE	TEST	—	All	Y	N	—
BLOCK	ADJUST	0	All	Y	N	—
	CONSTANT	—	All	Y	N	—
	FORMAT	BIN	All	Y	N	—
	LENGTH	1330	All	Y	N	—
	LMULT	1	All	Y	N	—
	LTHFLD	0	All	Y	N	—
	OFFSET	0	All	Y	N	—
	POSTAMBLE	0	All	Y	N	—
	PREAMBLE	0	All	Y	N	—
	ZERO	NO	All	Y	N	—

Table A-1. **PDL commands and DJDEs** (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
BSELECT	TEST	—	All	Y	N	—
ac:CATALOG	—	—	All	Y	Y	—
ac:CME	CONSTANT	—	All	Y	Y	—
	FONTS	—	All	Y	Y	—
	INK	—	4850, 4890	Y	Y	—
	LINE	—	All	Y	Y	—
	POSITION	1	All	Y	Y	—
ac:CODE	ASSIGN	—	All	Y	Y	—
	DEFAULT	EBCDIC	All	Y	Y	—
ac:CRITERIA	CHANGE	—	All	Y	Y	—
	CONSTANT	—	All	Y	Y	—
	LINENUM	ALL LINES	All	Y	Y	—
	VALUE		All	Y	Y	—
END	—	—	All	Y	Y	—
EXPORT	SEPARATORS	NONE	4850, 4890, 4135	Y	Y	—
	SNUMBER	—	4850, 4890, 4135	Y	Y	Page
	SPLIT	—	4850, 4890, 4135	Y	Y	Page
	SRECOVER	PAGE	4850, 4890, 4135	Y	Y	Page
	STIMING	—	4850, 4890, 4135	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	—
ac:IDR	ICATALOG	DFAULT	4850, 4890	Y	Y	Page
	ILIST	—	4850, 4890	Y	Y	Page
	PALETTE	DFAULT	4850, 4890	Y	Y	Page
JDE	INCLUDE	—	All	Y	Y	Page
JDL	—	—	All	Y	Y	Page
ac:JOB	INCLUDE	—	All	Y	Y	—
LINE	DATA	offline=1,132	All	Y	Y	Record
		online=0,150	All	Y	Y	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	All	Y	Y	—

Table A-1. **PDL commands and DJDEs** (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
MESSAGE	ITEXT	NONE	All	Y	Y	Page
	OTEXT	NONE	All	Y	Y	Page
OUTPUT	BFORM	NONE *init* defaults to first or next copy	All	Y	Y	Page
	COLLATE	YES	All	Y	Y	Page
	COPIES	1	All	Y	Y	Page
	COVER	NONE	All	Y	Y	—
	CYCLEFORMS	NONE	All	Y	Y	—
	DENSITY	DEFAULT	All	Y	Y	—
	DESTINATION	NO	All	Y	Y	—
	DUPLEX	NO	All	Y	Y	Page
	FACEUP	NO;	All	Y	Y	—
	FEED	OPR	All	Y	Y	Page
	FORMAT	FMT1	All	Y	Y	Page
	FORMS	NONE *init* defaults to first or next copy; FOR defaults to FORMAT if FORMS not fully keyed	All	Y	Y	Page
	GRAPHICS	NO	All	Y	Y	Record
	IDFAULT	first ink in ILIST parameter	4850, 4890	Y	Y	Page
	IDR	DFIDR.IDR	4850, 4890	Y	Y	Page
	IMAGE	inches, top left corner scaled at 1/1	All	Y	Y	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	—
	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	—
	SF1FUNCTION	NO	4850, 4890, 4135	Y	Y	Page
	SF2FUNCTION	NO	4850, 4890, 4135	Y	Y	Page
	SHIFT	NO	All	Y	Y	Page
	SIZING	—	All	Y	Y	—
	STAPLE	NO	All	Y	Y	—

Table A-1. **PDL commands and DJDEs** (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
	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	Y	—
RECORD	ADJUST	0	All	Y	N	—
	CONSTANT	—	All	Y	N	—
	FORMAT	BIN	All	Y	N	—
	LENGTH	offline=133, online=150	All	Y	Y	—
	LMULT	1	All	Y	N	—
	LTHFLD	0	All	Y	N	—
	OFFSET	0	All	Y	N	—
	POSTAMBLE	0	All	Y	N	—
	PREAMBLE	0	All	Y	N	—
	STRUCTURE	FB	All	Y	N	—
RFEED	TEST	no default for *clu-def*	4050, 4090, 4650 (V3.8), 4850 (V5.0), 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
RPAGE	SIDE	NUFRONT, NOFFSET	All	Y	Y	Page
	TEST	—	All	Y	Y	—
	WHEN	TOP	All	Y	Y	—
RRESUME	BEGIN	NEXT	All	Y	Y	—
	TEST	—	All	Y	Y	—
RSELECT	TEST	—	All	Y	Y	—

Table A-1. **PDL commands and DJDEs** (continued)

Command	Parameter	Default	LPS	Offline	Online	DJDE orientation
RSTACK	ACCTINFO	—	All	Y	Y	—
	DELIMITER	NO	All	Y	Y	—
	HRPTNA	NONE	All	Y	Y	—
	PRINT	NONE	All	Y	Y	—
	TEST	—	All	Y	Y	—
RSUSPEND	BEGIN	NEXT	All	Y	Y	—
	TEST	—	All	Y	Y	—
SEFFNT	MAP	—	4135	Y	Y	Page
	SEFMAP	—	4135	Y	Y	Page
ac:STOCKSET	ASSIGN	—	All	Y	Y	—
	INIFEED	first *stock-name*	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, NOCHECK, NOBREAKPAGE, SIMPLE, INTEGRAL, PERFORMANCE, RIP	All	Y	N	—
	LABEL	STANDARD	All	Y	N	—
	LCODE	EBCDIC	All	Y	N	—
	LPACK	NO	All	Y	N	—
	MAXLAB	81	All	Y	N	—
	MINLAB	80	All	Y	N	—
	OPTIMIZE	NONE	All	N	Y	—
	OSCHN	9	All	Y	N	—
	OSHDP	0	All	Y	N	—
	OSTLP	0	All	Y	N	—
	PLABEL	NO	All	Y	N	—
	RMULT	1	All	Y	N	—
	RSAT	REMOUNT	All	Y	N	—
	TCODE	EBCDIC	All	Y	Y	—
	UNPACK	NONE	All	Y	N	—

B. Character code assignments

IBM BCD code set

Table B-1. **IBM BCD code set**

			Most Significant Bits							
	Octal	(columns)	0	1	2	3	4	5	6	7
	(rows)	Binary	000	001	010	011	100	101	110	111
Least Significant Bits	0	000		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.

[t]Corresponds to BCD code set used by IBM users and defined by PDL command CODE = IBMBCD.

Honeywell 200/2000 BCD code set

Table B-2. **Honeywell 200/2000 BCD code set**

			Most Significant Bits							
	Octal	(columns)	0	1	2	3	4	5	6	7
	(rows)	Binary	000	001	010	011	100	101	110	111
Least Significant Bits	0	000	0	8	+	H	–	Q	<	Y
	1	001	1	9	A	I	J	R	/	Z
	2	010	2	'	B	;	K	#	S	@
	3	011	3	=	C	.	L	$	T	,
	4	100	4	:	D	)	M	*	U	(
	5	101	5		E	%	N	"	V	{
	6	110	6	>	F	[	O	]	W	}
	7	111	7	&	G	?	P	!	X	

Notes: 0'15' is the official blank character; 0'77' is the padding character.

[t]Corresponds to BCD code set used by Honeywell 200/2000 users and defined by PDL command CODE = H2BCD.

Honeywell 6000 BCD code set

Table B-3. **Honeywell 6000 BCD code set**

			Most Significant Bits							
	Octal	(columns)	0	1	2	3	4	5	6	7
	(rows)	Binary	000	001	010	011	100	101	110	111
Least Significant Bits	0	000	0	8	space	H	¬ 1 / ∧ 2	Q	+	Y
	1	001	1	9	A	I	J	R	/	Z
	2	010	2	[	B	&	K	-	S	_
	3	011	3	#	C	.	L	$	T	,
	4	100	4	@	D	]	M	*	U	%
	5	101	5	:	E	(	N	)	V	=
	6	110	6	>	F	<	O	;	W	"
	7	111	7	?	G	\	P	'	X	!

Notes: 1. Usual BCD Character
2. Printer Character

[t]Corresponds to BCD code set used by Honeywell 600/6000 series SSF tapes and defined by PDL command CODE = BCD or CODE = H6BCD.

Fieldata translation

Table B-4. **Fieldata translation**

		Most Significant Bits							
Octal (columns)		0	1	2	3	4	5	6	7
(rows)	Binary	000	001	010	011	100	101	110	111
0	000	@	C	K	S	)	*	0	8
1	001	[	D	L	T	-	(	1	9
2	010	]	E	M	U	+	%	2	'
3	011	#	F	N	V	<	:	3	;
4	100	^	G	O	W	=	?	4	/
5	101	(blank)	H	P	X	>	!	5	.
6	110	A	I	Q	Y	&	,	6	"
7	111	B	J	R	Z	$	\.	7	_

Least Significant Bits

UNIVAC ASCII character set

Table B-5. **UNIVAC ASCII character set**

Octal	Character	Octal	Character	Octal	Character
040	(blank)	100	@	140	\
041	!	101	A	141	a
042	"	102	B	142	b
043	#	103	C	143	c
044	$	104	D	144	d
045	%	105	E	145	e
046	&	106	F	146	f
047	/	107	G	147	g
050	(	110	H	150	h
051	)	111	I	151	i
052	*	112	J	152	j
053	+	113	K	153	k
054	,	114	L	154	l
055	-	115	M	155	m
056	.	116	N	156	n
057	/	117	O	157	o
060	0	120	P	160	p
061	1	121	Q	161	q
062	2	122	R	162	r
063	3	123	S	163	s
064	4	124	T	164	t
065	5	125	U	165	u
066	6	126	V	166	v
067	7	127	W	167	w
070	8	130	X	170	x
071	9	131	Y	171	u
072	:	132	Z	172	z
073	;	133	[	173	{
074	<	134	\	174	¦
075	=	135	]	175	}
076	>	136	^	176	~
077	?	137	_	177	(null)

Standard ASCII character set

Table B-6. **Standard ASCII character set**

			MOST SIGNIFICANT BITS															
	Hexadecimal		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LEAST SIGNIFICANT BITS	0	0000	RESERVED FOR XEROX USE ONLY			0	@	P	c	p								
	1	0001			!	1	A	Q	a	q								
	2	0010			"	2	B	R	b	r								
	3	0011			#	3	C	S	c	s								
	4	0100			$	4	D	T	d	t								
	5	0101			%	5	E	U	e	u								
	6	0110			&	6	F	V	f	v								
	7	0111			'	7	G	W	g	w								
	8	1000			(	8	H	X	h	x								
	9	1001			)	9	I	Y	i	y								
	A	1010		{	*	:	J	Z	j	z								
	B	1011		|	+	;	K	[	k								
	C	1100		}	,	<	L	\	l									
	D	1101		~	-	=	M	]	m									
	E	1110		¼	.	>	N	^	n									
	F	1111		½	/	?	O	_	o									

Standard EBCDIC character set

Table B-7. **Standard EBCDIC character set**

		MOST SIGNIFICANT BITS															
	Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LEAST SIGNIFICANT BITS	0 0000						&	-		¼	½			{	}	\	0
	1 0001							/		a	j			A	J		1
	2 0010									b	k	s		B	K	S	2
	3 0011									c	l	t		C	L	T	3
	4 0100									d	m	u		D	M	U	4
	5 0101									e	n	v		E	N	V	5
	6 0110									f	o	w		F	O	W	6
	7 0111									g	p	x		G	P	X	7
	8 1000									h	q	y		H	Q	Y	8
	9 1001								¢	i	r	z		I	R	Z	9
	A 1010					¢	!	^	:								¦
	B 1011					.	$	,	#	{	}						
	C 1100					<	*	%	@								
	D 1101					(	)	_	'			[	]				
	E 1110					+	;	>	=								
	F 1111					¦	~	?	"								

Xerox EBCDIC to extended ASCII hexadecimal translation values

Xerox EBCDIC to extended ASCII hexadecimal translation values

VALUES ON PERIMETER ARE EBCDIC. VALUES WITHIN BLOCKED RECTANGLE ARE ASCII. EXAMPLE: EBCDIC VALUE OF AB IS EQUIVALENT TO AN ASCII VALUE OF 7D.

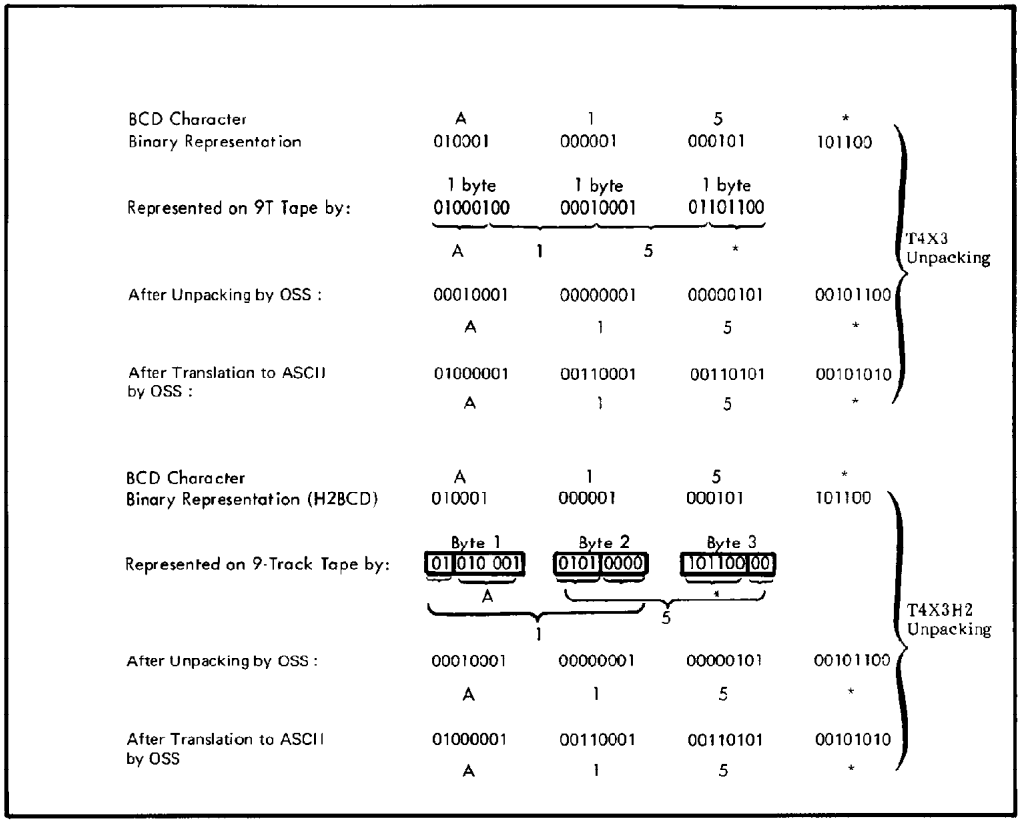
		EBCDIC MOST SIGNIFICANT DIGIT															
	Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
EBCDIC LEAST SIGNIFICANT DIGIT	0 0000	20	FA	EA	DA	20	26	2D	BO	1E	1F	8C	8O	1A	1C	5C	3O
	1 0001	20	F9	E9	D9	CA	C1	2F	AF	61	6A	7B	B1	41	4A	9F	31
	2 0010	20	F8	E8	D8	C9	CO	B8	AE	62	6B	73	82	42	4B	53	32
	3 0011	20	F7	E7	D7	C8	BF	B7	AD	63	6C	74	83	43	4C	54	33
	4 0100	20	F6	E6	D6	C7	BE	B6	AC	64	6D	75	84	44	4D	55	34
	5 0101	20	F5	E5	D5	C6	BD	B5	AB	65	6E	76	85	45	4E	56	35
	6 0110	20	F4	E4	D4	C5	BC	B4	19	66	6F	77	86	46	4F	57	36
	7 0111	20	F3	E3	D3	C4	BB	B3	18	67	7O	78	87	47	5O	58	37
	8 1000	20	F2	E2	D2	C3	BA	B2	AA	68	71	79	88	48	51	59	38
	9 1001	20	F1	E1	D1	C2	B9	B1	6O	69	72	7A	89	49	52	5A	39
	A 1010	20	FO	EO	DO	6O	21	5E	3A	A9	9O	91	A8	A7	A3	9E	1B
	B 1011	FF	EF	DF	CF	2E	24	2C	23	1A	1C	7D	17	A6	A2	9D	99
	C 1100	FE	EE	DE	CE	3C	2A	25	4O	1O	12	7E	7F	13	8F	15	98
	D 1101	FD	ED	DD	CD	28	29	5F	27	8A	8D	5B	5D	A5	8E	9C	97
	E 1110	FC	EC	DC	CC	2B	3B	3E	3D	8B	7C	11	95	14	A1	9B	96
	F 1111	FB	EB	DB	CB	1B	1D	3F	22	94	93	92	16	A4	AO	9A	2O

C. Offline specifications

This appendix provides offline specifications.

Input unpacking examples

Figure C-1. **Input unpacking examples**



Valid host computer and label specifications

Table C-1. **Valid host computer and label specifications**

Host type	Label specification					
	Unlabeled	ANSI	Standard	System print	COBOL	Undef
IBMOS	X	X	X			
IBMDOS	X	X	X			
GRASP POWER POWERVS	Label specification is ignored					
OSWTR		X	X			
US70	X	X	X			
XEROX	X	X	X			
B2500 Burroughs medium systems	X	X	X			
B2700	X	X	X			
B3500	X	X	X			
B3700	X	X	X			
B4700	X	X	X			
B6700 Burroughs large system	X	X	X			
H2000 Honeywell 200/2000 series			X	X	X	
H6000 Honeywell 600/6000 series			X			
DUMP	Label specification is ignored					
OCTDUMP	Label specification is ignored					
UNIVAC	X	X	X			
ANSI		X				
NCR	X		X			
RSX11	X					
ICL2900			X			
OLDUMP	Label specification is ignored					
Undef						X

Host system JDLs on system software tape

Job description library (JDL) source files supplied on operating system software (OSS) tape are summarized below.

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: **JDE** — **Characteristics** 1-or-H6BCD — H6000 BCD 2-or-EBCDIC — EBCDIC 3-or-ASCII — ASCII 4-or-H2BCD — H2000 BCD 5-or-IBMBCD — IBM BCD 6-or-UNIVAC — UNIVAC Fieldata 7-or-ICL — ICL 2900 VME/B
H2000	Honeywell 2000 SPR-labeled tapes Honeywell 2000 standard, COBOL, and SPR BCD-formatted tapes
H6000	Honeywell 6000 SSF BCD- and ASCII-formatted tapes
IBMRCA	IBM OS and DOS standard-labeled tapes IBM ANSI-labeled and OS Writer tapes US70 RCA-labeled tapes
ICL	ICL 2900-labeled tapes
NCR	NCR-labeled tapes
OLDUMP	Online dumps of host transmissions
ONLINE	Online with or without banner or trailer pages
POWGRP	IBM POWER VS, POWER 4.0 POWER 4.1 or 4.2; IBM DOS GRASP (including Interpress tapes)
RSX	RSX-11 (FLX) ASCII tapes (including Interpress tapes)
UNIVAC	Univac DF-formated tapes
XEROX	ANSI-labeled tapes Unlabeled tapes Xerox ANSI-labeled tapes JDEs to print unknown tapes Printing tapes without attempting to deblock the tape properly

After an LPS software system is created by the user, the source for these JDLs is resident in the JSL directory. Use the Editor PRINT parameter or the PDL compiler to obtain a printout of the JDL(s) desired.

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 channel positions and forms length. It is analogous to a carriage tape on an impact printer. Each byte in the buffer corresponds to one line on the form.

Table 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 nnnn	Shift right n- 1positions
010N nnnn	Shift left n- 1 positions (first n- 1 bytes of each record is not printed by the LPS)

Online input processing modifies the size and address of the print line based on the PPI byte before processing the DATA command of the LINE statement. If the resultant size of the print line is zero or negative (after PPI modification), only PCC action takes place.

Function of UCSB byte locations

The correspondences between location in the associative field hexadecimal values, characters, and the associative bits that determine the character printability are listed in tables D-1 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.

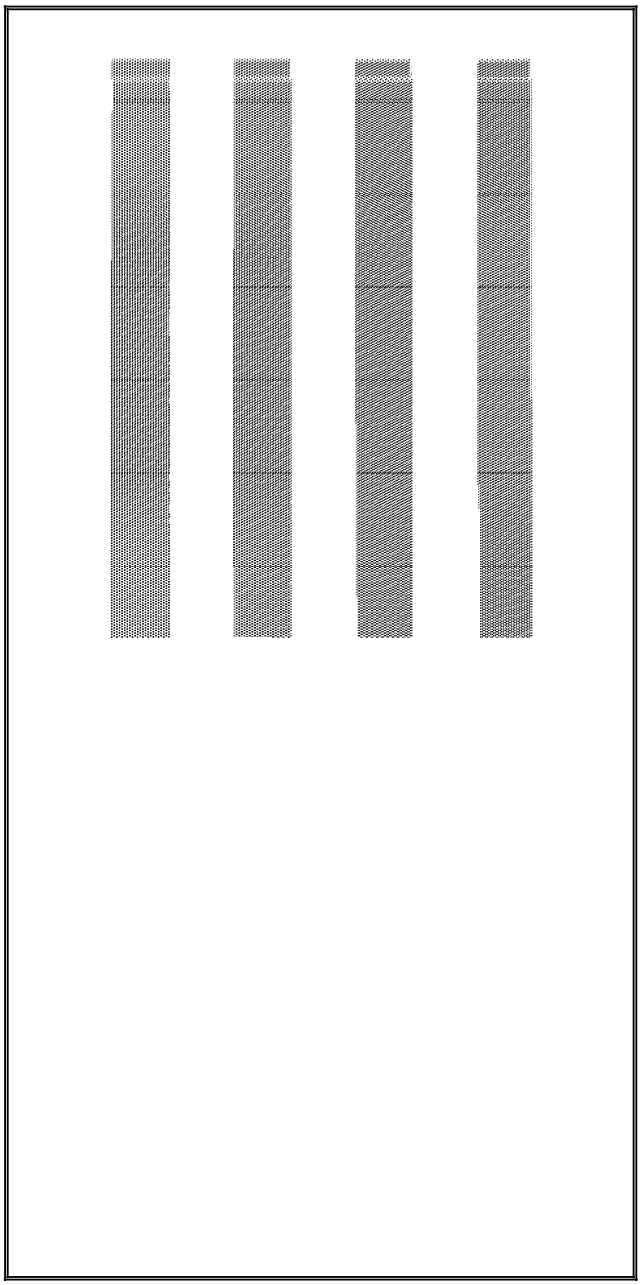
Characters associated with UCSB field

Table D-4. **Characters associated with UCSB field**

UCSB Location								
Byte	Bit 0		Bit 1		Bit 2		Bit 3	
	Hex	Char	Hex	Char	Hex	Char	Hex	Char
448	00		40	Sp	80	¼	C0	{
449	01		41		81	a	C1	A
450	02		42		82	b	C2	B
451	03		43		83	c	C3	C
452	04		44		84	d	C4	D
453	05		45		85	e	C5	E
454	06		46		86	f	C6	F
455	07		47		87	g	C7	G
456	08		48		88	h	C8	H
457	09		49		89	i	C9	I
458	0A		4A	¢	8A		CA	
459	0B		4B	.	8B	{	CB	
460	0C		4C	<	8C		CC	
461	0D		4D	(	8D		CD	
462	0E		4E	+	8E		CE	
463	0F		4F	¦	8F		CF	
464	10		50	&	90	½	D0	}
465	11		51		91	j	D1	J
466	12		52		92	k	D2	K
467	13		53		93	l	D3	L
468	14		54		94	m	D4	M
469	15		55		95	n	D5	N
470	16		56		96	o	D6	O
471	17		57		97	p	D7	P
472	18		58		98	q	D8	Q
473	19		59		99	r	D9	R
474	1A		5A	!	9A		DA	
475	1B		5B	$	9B	}	DB	
476	1C		5C	*	9C		DC	
477	1D		5D	)	9D		DD	
478	1E		5E	;	9E		DE	
479	1F		5F	~	9F		DF	

Note: Characters in the “Char” column are shown for illustration. 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. 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	D3	Skip immediate to channel	73	Block data check
D9	10	DB	10	7B	Allow data check
E1	Write and skip to channel	E3	Skip immediate to channel	FB	Load UCSB
	11			E4	Sense ID (4245 only)
				87	Clear Printer (4245 only)

Glossary

4890 palette A predefined set of colors or inks provided with the 4890 HighLight Color LPS. Different versions are provided with the printer and with host- or PC-based application software.

A3 International paper size measuring 297 mm by 420 mm or 11.69 by 16.54 inches.

A4 International paper size measuring 210 by 297 mm or 8.27 by 11.69 inches.

additive primary color There are five additive primary colors: red, green, blue, magenta and cyan. When light of these five colors is combined in equal amounts, the result is white light.

American Standard Code for Information Interchange (ASCII) Standard 7-bit code that represents alphanumeric information Each alphanumeric character and several nonprinting characters are assigned a binary number, covering 128 possible characters It is used for information interchange among data processing systems, data communication systems, and associated equipment.

ANSI American National Standards Institute.

application software Host- or LPS-resident software that directs the computer to perform specific tasks or functions. Common business applications include payroll, accounting, and inventory.

argument Independent variable of a function, such as a parameter.

ascender Portion of a lowercase alphabetic character that extends above the main body of characters such as b, d, or h. See also *descender*.

asynchronous Transmission in data communications controlled by start and stop characters; thus, time intervals between transmitted data blocks may be unequal in length.

ASCII American Standard Code for Information Interchange. Standard 7 or 8--bit code that represents alphanumeric characters and several nonprinting characters by assigning each a binary number covering 128 possible characters. It is a common standard for information interchange among data processing systems, data communication systems, and associated equipment.

B4 International paper size measuring 250 mm by 353 mm or 9.84 by 13.9 inches.

batch processing Process that allows for repetitive operations to be performed sequentially on batched data without much involvement from the computer operator.

background job Low-priority job, usually batched, which is executed automatically as system resources become available.

baud Measurement of data flow between devices in bits per second. Common baud rates are 110, 300, 1200, 2400, 4800, and 9600.

BCD Binary coded decimal.

binary Numbering system with a base of 2. All numbers are represented by combinations of 1 and 0.

binary synchronous communication See *BSC*.

binary synchronous transmission Data transmission where synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

bit Abbreviation for binary digit, the smallest unit of information recognized by a computer.

bitmap 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.

bitmapped Display image generated bit by bit for each point or dot. A software-driven scanner is used to create characters and/or graphics.

bits per second See *bps*.

block length Number of characters or bytes contained in a block of data (the block is treated as a unit within the computer). Block length is usually invariable within a system and may be specified in units such as records, words, computer words, or characters.

blocking 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.

BOF Bottom of form.

boot To load the initial instructions of a program into memory. These instructions direct the loading of the operating system and application software.

BOT Beginning of tape.

bpi Bits per inch.

bps Bits per second. In serial communication, the instantaneous bit speed a device or channel transmits a character.

BSC Binary synchronous communications. 1. Data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations. 2. Communication using binary synchronous line discipline. 3. Uniform procedure using a standardized set of control characters and control character sequences for synchronous transmission of binary-coded data between stations.

buffer Area of memory in which data is stored during transfer from one device to another. Used for: 1. Accumulating data into blocks before storage or processing. 2. Adjusting differences of speed between devices, or between a device and a communicating facility.

byte Fixed number of bits (in data processing, usually 8) processed as a single binary value.

cache memory A fast, small memory used to enhance CPU performance, separate from the main processor memory.

CCID 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.

CCU Customer Changeable Unit. The color housing that fits inside the printer.

central processing unit Interprets and executes instructions, performs all operations and calculations, and controls input and output units and auxiliary attachments.

channel 1. In data communications, a path or line that enables two or more devices to communicate (sometimes called a circuit, facility, or link). 2. In computers, a path for communication between the central processing unit (CPU) and input and output units, or between the CPU and peripheral devices.

character cell Area defined by the outside dimensions of a character plus all horizontal and vertical spacing. In electronic printing, the character cell is defined by a bitmap of dots.

character code identifier See *CCID*.

character set Set of all characters defined in a font, including alphabetic, numeric, and special characters such as symbols.

characters per inch See *cpi.*

clearing house service Directory of registered users, services, and other resources, allowing relevant information about each item to be retrieved by name.

clocking A method of synchronizing sending and receiving data communications devices. Clocking allows synchronous transmission at high speeds.

cluster 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.

CME Entry modifying the output printing characteristics of a report on a copy-to-copy basis.

CMT Character mapping table.

CMYK A printing industry standard color definition model where all colors are defined in terms of values for the four-color process primary colors: cyan, magenta, yellow, and black. Refer to color definition model.

Code 1. A set of symbols representing data or instructions to a computer. 2. To write a program or routine that instructs a computer to perform specified operations.

code conversion Translation of one type of character or symbol code to another.

color definition model A standardized format for defining colors using quantifiable values. The most popular such models are RGB, CMYK, and HSB.

color substitution The 4890 HighLight Color LPS uses an algorithm to translate a color specified in a printing industry standard color definition model to a tone of the Highlight primary color using a 4890 palette.

command language A language used to instruct an operating system.

communication line Telecommunication line connecting devices at one location with devices at other locations in order to transmit and receive information.

communication link Physical means connecting one location to another to transmit and receive information, such as a data link

compiler Software that translates instructions written in high-level language into machine language for execution by a system.

composite character Character that has multiple bitmap references combined into one and stored on disk in font memory.

continuous printing Refers to Interpress job integrity under any of the following conditions: excessive graphics, forms, or font use problems.

continuous tone A photograph, rendering, or other similar image that is made of blended gray tones or values that flow into each other gradually and without hard edges.

control program An operating system program that manages job flow, input/output processing, and other overall system functions and resources.

Copy Modification Entry See *CME*.

copy-sensitive Job in which multiple copies of a report contain different data, such as paychecks and banking statements.

cpi Characters per inch. Measure of the size of a fixed-pitch font expressed as the number of characters of the font that can be set in one horizontal inch. See also *pitch.*

CSI Command status interface.

Data Capture Utility See *DCU.*

data communications Transmission and reception of encoded information over telecommunication lines.

data file Collection of related data records organized in a specific manner so that each record is similarly structured. An example of this would be a payroll file set up with one record for each employee, last name first, indicating the rate of pay and all deductions.

data link Physical means of connecting one location with another for communication. This might include the communications lines, modems, and controls that transmit information between two or more stations.

DCU Data capture utility. LPS function that saves and prints all system controller activity and provides tracing and event logging facilities.

DDCMP Digital Data Communication Message Protocol.

default Value assigned to a field by the system if no input is received from the operator.

defined ink The tone (defined in the ink catalog file and reproduced for your reference on the printed samples of 4890 HighLight Color LPS palettes) as it is applied to the page.

descender Portion of the lowercase alphabetic character that extends below the main body of characters such as g, p, or y. See also *ascender*.

DEC Digital Equipment Corporation.

device Any piece of hardware other than the CPU (Central Processing Unit).

digitize To express or represent data in digital (binary) form so that it can be processed electronically.

DJDE Dynamic job descriptor entry. Command within an input data stream used to modify the printing environment dynamically.

DMA Direct memory access.

DMR Data mode ready. Command parameter that designates a Digital Equipment Corporation host system.

document 1. Data medium and the data recorded on it, usually permanent, which can be read by you or a computer. 2. Collection of information pertaining to a specific subject or related subjects.

dot Picture element (pixel) imaged by a printer. The number of dots imaged per inch measures printer resolution, for example, 300 dots per mich (dpi). See also *spot*.

dpi Dots per inch. Indicates the number of dots per inch displayed on a terminal screen or printed to form a character or graphic.

dry ink 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.

DSDD Double sided double density.

DSSD Double sided single density.

DSU Digital signal unit.

duplex 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.

dynamic Job descriptor entry See *DJDE.*

EBCDIC Extended binary coded decimal interchange code. Coded character set consisting of up to 256 8-bit coded characters.

edge marking Use of graphic objects (usually lines or boxes) that bleed off the edge of the physical page. See also *physical page*.

electronic publishing The integrated production of documents on demand, using digitally stored documents, computerized composition, and electronic printing systems.

elite Smallest size standard typewriter type which is 12 characters per horizontal inch.

embedded blanks Blank spaces within a command line.

EMT Emulator trap. Language instruction.

enabler Hardware devices or software packages that come with the printer and allow it to perform as specified.

ENET Ethernet network.

EOF End of file.

EOT End of tape.

escapement Lateral positioning of characters or font families.

Ethernet Xerox developed local area network (LAN) that allows transmission of data by cable from one device to another. A modified version of the Ethernet specification has been approved as IEEE standard 802.3.

Extended binary coded decimal interchange code See *EBCDIC*.

extended metrics Measurements used in Interpress to alter the size of fonts, allowing more precision with character escapement. Used for rendered characters.

FCB Forms control buffer. Controls the vertical format of printed output.

FCP File control parameter.

FCG Finishing Configuration Utility.

FCU File Conversion Utility.

FDL Forms description language. LPS-resident source language used to design electronic forms. See also *FSL* and *form.*

FDR File directory.

feedback Portion of an output signal that is returned, directly or indirectly, to be compared to a reference signal to maintain the quality of the output signal.

FFM Font file management.

firmware Permanent programs stored in read-only memory (ROM).

fixed font Font containing characters with fixed spacing.

fixed pitch Font set with every character cell having the same width. In reference to character sets, this term describes typefaces with all character cells having equal width. Monospaced as opposed to proportional spaced.

fixed spacing Arrangement of characters on a line so that all characters occupy the same amount of horizontal space.

flag Small indicator marking the occurrence of an event or the existence of a certain condition while the program is executing.

floating accent Nonspacing accent characters that can be combined with characters and printed as a composite.

font Complete set of characters of a particular font family having the same point size, weight, stress, and orientation.

Font Interchange Standard (FIS) A standard that defines the digital representation of fonts and character metrics for the generation of an entire series of Interpress fonts.

form 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 *FDL* and *FSL*.

format 1. Layout of a document, including margins, page length, line spacing, and typeface. 2. In data storage, the way the surface of a disk is organized to store data. 3. To prepare the surface of a disk for acceptance of data.

format line Line preceding the message text in a formatted message display.

form feed Keyboard and printer control character that causes the printer to skip to the top of the next page.

forms control buffer See *FCB.*

forms description language See *FDL.*

forms source library See *FSL.*

FPS Formatting print service.

FSL Forms source library. Uncompiled collection of user-created files containing FDL commands. See also *FDL* and *form*.

FST Font specification table.

fuse To affix dry ink to paper by heat or pressure or a combination of both.

GB Gigabyte. Unit of approximately one billion bytes.

GCR Group code recording.

GHO Graphics handling option.

gigabyte See *GB*.

group code recording (GCR) mode Refers to the specific density of data (such as 6250 bpi) as it is recorded on tape, which is measured in bits per inch (bpi).

gsm Grams per square meter.

GVG Graphics video generator.

halftone screen A tool used in offset printing, typesetting, and laser printing to convert a continuous tone (such as photographic) image to dots, which allows the image to be rendered accurately in these printing processes.

hardcopy Machine output in permanent form, such as printed reports and listings. Output in a permanent form (usually on paper or paper tape) rather than in temporary form, as on a display. Contains readable printed copy of machine (such as computer) output.

hardware Physical components (mechanical, magnetic, electronic, and so on) of a system, as opposed to programs, procedures, rules, and associated documentation. The hardware is operated by software and firmware.

HCF High-capacity feeder.

HCSS High-capacity stitcher/stacker; also referred to as stitcher/stacker.

hexadecimal Numbering system with a base of 16. The numbers 10 through 15 are represented by A through F.

HFDL Host forms description language.

hierarchy Relative priority assigned to arithmetic or logical operations that must be performed.

high-level language Programming language consisting of words and symbols that are close to normal English and, also readily understandable by the user. High-level source languages are used for most commercial programs.

highlight color 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.

HIP Host interface processor.

host Computer accessed by users that serves as a source of high-speed data processing for workstations with less computer power. Commonly referred to as mainframe.

host interface Connection between network and host computer.

HSB A printing industry standard color definition model in which all colors are defined in terms of values for hue, saturation, and brightness. Refer to *color definition model*.

hue The name that describes a color's general appearance, such as red, green, and blue.

identifier (id) Character(s) used to identify or name data and possibly to indicate certain properties of that data image area on a physical page that may contain text or graphics.

IFU Interpress font utility. A utility for managing Interpress fonts.

image area Area on a physical page that may contain text or graphics.

initialize 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.

initial program load See *IPL.*

ink catalog File containing inks that is referenced by all software for color information. The ink catalog file has the extension .ICT.

ink referencing Process by which inks within the ink catalog are referenced in PDL and FDL commands, and by page description language interpreters.

ink substitution Option of aborting the printing process or continuing with a substitute ink if the referenced ink is not found.

input devices Keyboards, magnetic media, or any device used to give a system information.

input/output General term encompassing the flow of data into and out of a system. Also referred to as I/O.

interface The device that connects two systems to allow them to communicate.

Interpress Industry-standard page description language developed by Xerox. Interpress documents can be printed on any sufficiently powerful printer equipped with Interpress print software.

Interpress font utility (IFU) Utility for managing fonts.

Interpress font utility (IFU) program Program used to convert FIS fonts to LPS fonts.

Interpress master File written according to the Interpress standard.

IPD Interpress decomposer.

IPL Initial program load. For the optional 9-track magnetic tape drive, the internal initialization sequence whereby certain functions are loaded into random access memory (RAM).

IPM Interpress mapping.

ips Inches per second.

JCB Job control block.

JCL Job control language.

JDE Job descriptor entry. Collection of job descriptions.

JDL Job description library. Collection of compiled job descriptions. See also *JSL*.

JID Job identifier.

job descriptor entry See *JDE.*

job descriptor library See *JDL*.

job source library See *JSL*.

JSL Job source library. Collection of uncompiled job descriptions. See also *JDE* and *JDL*.

keyword Required part of a command.

label Reference to a file saved on tape or disk, a record indicating the file name or date created, or other control information.

LAN Local area network. LAN is the lower two layers of the network architecture: the physical layer and the data link layer

landscape Orientation in which text and images are positioned parallel to the long edge of the paper.

laser printing Technology that uses a laser to transfer character forms to a page by direct or indirect means.

latent image Static charge present on the photo conductor before contact with dry ink particles.

leading (pronounced *ledding*) Vertical distance between lines of type (also called line space), measured from a baseline of one line to the baseline of the next.

legal size Paper size measuring 8.5 by 14 inches or 216 by 356 mm.

letter size Paper size measuring 8.5 by 11 inches or 216 by 279 mm.

light emitting diode (LED) Solid substance that glows when a current is passed through it. Often used for indicator lights on disk drives or modems, as well as for displays on other electronic equipment.

LF Line feed character.

line feed Control character that (unless set to be interpreted as a line end) causes the printing system to begin printing in the current character position of the next line.

literal Alphanumeric character beginning with a letter, including an asterisk, period, colon, or slash, and not enclosed in single quotes.

logical page 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.

logo Small illustration or design, usually simple, typically used to identify a company.

long-edge feed See *LEF*.

lpi Lines per inch.

LPS Laser printing system.

LUN Logical unit number. A number that identifies a peripheral device to the OSS.

magnetic media Term for all storage devices (disks, tape, and so on) on which data is stored in magnetic form.

map To establish a set of values having a defined correspondence with the quantities or values of another set.

mask 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.

master file File serving as a general reference point for a particular application system and providing information to be used by the program. It is usually updated and maintained to reflect the results of current or daily processing operations.

MB Megabyte. Unit of approximately one million bytes.

metacode 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.

MHz Megahertz. One million cycles per second. Used to measure electromagnetic waves.

mixed environment Multiple printers or printing systems in the same location.

modem Device that converts digital information into an analog signal suitable for sending over analog telecommunication lines. Also converts an analog signal from telecommunication lines into digital information.

monochrome Printing in one color only.

nesting Subroutine or set of data, such as a comment, contained sequentially within another set of data.

network System of hardware and software that manages communication and sharing of common resources (such as printers) between computers and terminals for multiple users at once.

node Station, terminal or computer operating in a network environment.

object file Source file converted into machine language (binary code).

octal System of representing numbers based on 8.

offline Devices not under the active control of a central processing unit. For example, a computer makes output to a magnetic tape. The tape is then used by an offline printing system to produce printed data. Offline operations are much slower than online operations. Refer to *online.*

offset To place printed output sets in slightly different positions from each other in an output bin for easy separation of collated sets.

offset printing Widely-used method of commercial and corporate printing, in which ink is picked up by a metal or paper plate, passed to an offset drum, then passed to the paper.

online Devices under the direct control of a central processing unit, such as a printing system in interactive communication with a mainframe. Refer to *offline.*

operand That which is acted upon, for example, data, in an operation or process.

operating system 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.

orientation 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).

origin In reference to image area, the upper left corner of a sheet.

output 1. Material produced by a peripheral device of a computer, such as a printout or a magnetic tape. 2. The result of completed operations.

overprint lines Print lines whose carriage control specifies printing with no line spacing after the last printed line.

overprint ratio Maximum number of variable data and form characters that can be intersected by a single scan line.

page end Command character (form feed) to terminate the current page.

page orientation Direction that data is printed on a report. Refer to *landscape page orientation* and *portrait page orientation*.

pagination Process of separating text into pages.

palette Predefined set of colors or inks. Different versions are provided with the printer and with some application software packages.

parameter Part of a command, other than the keyword. See *keyword*.

pass-through job On systems with XPAF, a job that is sent directly from a host to a Xerox printer through XPAF without undergoing XPAF processing.

parse To read or interpret a command; to build up a parameter list from information within a command.

PCC Printer carriage control.

PDE Page description entry.

PDL 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).

PE Phase encoded.

physical page Actual page size your printer uses to print a form.

pica 1. Unit of measurement equal to twelve points or approximately 1/6 inch. 2. A 10-pitch typeface having ten characters per inch and 12 points in height. See also *point*.

pitch Width of a fixed-pitch font expressed in characters per horizontal inch.

pixel Acronym for picture element. Smallest addressable point of a bitmapped screen that can be independently assigned color and intensity.

point 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. See also *pica*.

point size Height of character set from the top of its ascenders to the bottom of its descenders in units (points). Point size does not always include leading.

portrait Orientation in which text and images are positioned parallel to the short edge of the paper.

PostScript Proprietary page definition language, compatible with the Xerox 4890 Highlight Color LPS when a front-end conversion utility is installed.

ppm Pages per minute.

print quality adjustment See *PQA*.

primary color A color that, when combined with one or more other primary colors in a color model system in varying quantities, produces the palette of colors described by that model.

print description language See *PDL.*

print ratio The maximum number of variable data and form characters that may be intersected by a single scan line.

printer subsystem controller See *PSC*.

print file 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.

PQA The process the 4890 HighLight Color LPS uses to maintain the electrostatic state of the xerographic subsystem to ensure optimal print quality.

printout Informal expression referring to almost anything printed by a computer peripheral device.

PROM Programmable read-only memory.

prompt Message or symbol displayed on a system console requiring the operator to take action.

process color printing In process color printing, every color in the spectrum is printed by combining various percentages of the four-color process primary colors–cyan, magenta, yellow, and black.

processor Applies to any system that is capable of receiving and performing operations upon data and supplying the results of those operations.

protocol Formal set of conventions governing format of data and control of information exchange between two communication devices.

proportional font Font containing characters that vary in width.

proportional spacing Text where each alphanumeric character is given a weighted amount of space. Such output has print-like appearance. Proportional spacing allows more space for wide characters and less space for narrow characters.

proportional type Characters that vary in width.

PSC Printer subsystem controller. Command and task that download the printer and/or raster image processor (RIP) software if selected.

query Request for data or other information, entered by an operator while the system is processing.

queue Managed database of documents waiting to be processed.

RAM Random access memory.

raster data Binary data, usually consisting of dots arranged in scan lines, according to the print order.

raster image processor See *RIP*.

rasterization The creation of a bitmap image on a page for printing.

read-only memory See *ROM*.

read/write head A small electromagnet that reads, writes, and erases data in the form of magnetic dots on the surface of an external storage medium, such as a magnetic disk.

real time To process transactions as they are entered into a system. User and system interchange allow transactions to be processed and the results returned to the user.

remote access Access to a central computer by terminals or devices geographically separated from that computer.

RGB A printing industry standard color definition model where all colors are defined in terms of values for the three additive primary colors: red, green and blue. Refer to *color definition model*.

ROM Solid-state memory for programs. It cannot be rewritten.

rendered character Character whose bitmap is a combination of other bitmaps positioned relative to each other, such as accented characters and single characters consisting of two or more characters. A rendered character does not require additional disk space or font memory.

report In setting a separation boundary, report refers to a subset of a job. A job may consist of one or more reports.

resolution 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.

RIP Raster image processor. LPS option that supports the Interpress V3.0 Publications Set.

scale To adjust font or image size according to given proportions.

sci START command index.

scroll Manipulation of a display to bring upper or lower portions of a document page into view when available space is insufficient to view the entire document at once.

SCSI Small computer system interface. Common standard for connecting devices to computers.

SDI System diagnostics interface. OSS task used by service personnel to diagnose system problems.

secondary storage Form of storage external to a system, such as magnetic tapes or floppy disks.

security 1. Procedure for limiting access to the system's resources, programs, or files, to authorized personnel. 2. Protecting programs and files from unintentional or undesired modification.

SEF See *short-edge feed.*

sequential 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.

set Multiple copies of the same report.

SFC Status file converter.

SFS Status file service.

shade Color resulting from adding black to a pure hue. See also *tint* and *tone*.

short-edge feed Movement of paper through the printer in the direction of the paper width (the shorter side of a sheet of paper).

SIF Sequence insert file.

simplex printing Printing on one side of the page.

small computer system interface See *SCSI.*

SNA System network architecture. Total description of logical structure, formats, and protocols of operation sequences for transmitting an information unit through the communication system.

special processing Commands allowing the user to process special reports such as printing certain records, and printing on special paper

spi Spots per inch. See *resolution*.

spooling Process of releasing data from main memory and storing it temporarily until a peripheral device is ready to accept it, for example, storing print data before sending it to a printer.

spot A picture element imaged by the printer. Synonymous with *dot* and *pixel*.

statement Detailed instructions in a program step, written according to specific rules called syntax.

static data Information usually found on preprinted forms or overlays.

station 1. In data communications, a terminal device connected to a data link. 2. In computer networks, any of the numerous terminal devices that form a network.

stock User-defined name in the JSL that specifies a certain type of paper for printing a job.

stock descriptor A stock reference or a stock name.

stock name The name of a cluster as defined by a system command.

stock reference An application specific pseudonym for a cluster.

stockset Collection of stocks to be used on a print job. See also *stock*.

string Connected sequence of alphanumeric characters treated as one unit of data by a program.

symbiont Device which shares a common purpose and close association with other devices. This term is used to identify the Xerox print software which resides in the Digital host system as part of the XPMF-VMS communications link.

symbol Character used in a computer language to specify a particular function.

synchronous Efficient encoding of data suitable for high-speed, block-oriented data transmission by using equal time elements.

syntax Rules governing the structure of expressions in a programming language.

syntax error System response to a mistake in a command's entry.

sysgen Abbreviated form of system generation.

System 1. In data processing, a collection of computer components and procedures organized to accomplish a set of specific functions. 2. Assembly of components united by some form of regulated interaction to form an organized whole. 3. Operations or procedures used to accomplish a business activity.

System controller The part of the LPS that provides interfacing capability, data handling, formatting, buffering, and operator control for the system. Also called the system controller.

system disk Magnetic disk storage medium, usually of large capacity, that is not removable as opposed to floppy disk or disk packs.

system file Master software program that keeps all components working together.

system generation Process whereby the system is made ready to operate. Typically involves selecting the operative parameters and activating the relevant software.

System Network Architecture See *SNA.*

system page Maximum area in which text and graphics can be imaged on a printing system.

tape density Expression of the format of a magnetic tape measured in number of bytes that can be stored per inch of tape.

tape drive Input/output device that controls the movement of magnetic storage tape past the read/write head while data is accessed or stored.

telecommunications The transfer of data through telephone lines.

teleprocessing Data transfer through telecommunication lines for processing among various remote terminals and the central processing unit (CPU).

templates 1. Preset document formats, usually furnished along with application software, such as electronic spreadsheets or data base programs. 2. Also applies to keyboard overlays showing function keys for particular software packages.

temporary storage Main memory locations reserved for intermediate results of processing, control values, or other information that needs to be kept on hand as a program proceeds.

terminal Interface device connected to a computer or network. A terminal has no processing capability of its own.

throughput Measure of the number of pages printed during a given unit of time, usually expressed as pages per minute.

tint Color resulting from adding white to a pure hue. See also *shade* and *tone*.

TOF Top of form.

toggle In an LPS system, to switch (alternate) from one tray to another. The system will switch from an active feeder or stacker tray to an inactive one to allow continuous printing when the proper commands are invoked.

token A coded character representing a word used in programs. For instance STOP is a word and the one-byte token for it in a BASIC dialect is 250.

tone Primary color in its pure form, or mixed with black or white to create a lighter tint or darker shade of the primary color. See also *shade* and *tone*.

total xerographic convergence (TXC) For the 4890 HighLight Color LPS, a function of the operating system software (OSS) that monitors the state of the color and black developer housings. If TXC detects a condition that may affect print quality, TXC requests that the operator initiate Print Quality Adjustment (PQA) to adjust the xerographic subsystem. Refer to *print quality adjustment.*

tpi Tracks per inch.

transaction processing Method of data processing that updates files and results are generated immediately after data entry.

translation 1. In data communications, the conversion of one code to another on a character-by-character basis. 2. In programming, the function of a language processor that converts a source program from one programming language to another.

transmission speed In data communications, the rate data is passed through communication lines, usually measured in bits per inch (bpi).

Tri-level xerography The feature of the 4890 HighLight Color LPS that enables the rasterization of both the black and the color images in a single pass by the creation of a third charge state.

truncated Cut off before completion, as when data transfer from a host to a printer is cut off before all data has been transmitted.

two-up Printing two logical pages on one side of a physical page.

TXC Total xerographic convergence.

type style Italic, condensed, bold, and other variations of typeface that form a type family.

typeface Set of fonts having identical design features that give the type a uniform appearance.

UCS Universal Character Set. Printer feature that permits the use of a variety of character arrays.

UCSB Universal Character Set Buffer.

UI User interface.

Universal Character Set See *UCS*.

utility program General-purpose program that performs activities, such as initializing a disk or sorting, which are not specific to any application.

validation Process of testing a system's ability to meet performance objectives by measuring or monitoring its performance in a live environment.

variable data 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.

variable text Text of changing nature (such as names and addresses) combined with a form letter to make a complete document.

verification Process of testing a system's ability to meet performance objectives by running programs in a simulated environment.

virtual page Page area selected by a forms designer for printing.

vpos Vertical positioning.

WAN Wide area network.

weight Perceived blackness of a character affected by varying the width of the stroke. Weight is expressed in general terms as either bold or roman.

wildcard 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.

write protection Data protection feature implemented on magnetic media (for example, floppy disk, 9-track tape) to prevent stored data from being modified, written over, or erased.

x axis Horizontal axis on a forms grid.

x height Height of lowercase letters without their ascenders or descenders (height of letter “x”).

xdot 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).

xerographic engine Component of a printer that develops an image, transfers it to paper, and fuses it for output as hardcopy.

xerographic mode 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.

xerographic mode persistence (XMP) For the 4890 HighLight Color LPS, a value specified during sysgen, in JDL or in DJDE code that specifies on what basis xerographic mode switching (XMS) may take place. Refer to *xerographic mode switching*.

xerographic mode switching (XMS) A function of the operating system software (OSS) that controls the xerographic mode of the 4890 HighLight Color LPS. The operator or programmer controls XMS through the xerographic mode persistence (XMP) setting. Refer to *xerographic mode*.

XICS Xerox Integrated Composition System.

XJCF Xerox Job Control Facility.

XNS Xerox Network Systems is the network architecture and protocol used with Xerox equipment.

XPAF, XPF Xerox Printer Access Facility.

XPPI Xerox Pen Plotter Interface.

XPS (Xerox Print Service) Xerox Print Service (XPS) Manager is a printer front end to other hosts. XPS Manager server communicates with the printer via a layered communication protocol, based on the XNS protocol suite but enhanced with “online“ capability.

XPS (Xerox Publishing System) Xerox Publishing System.

XPMF-VMS Xerox Print Management Facility-VMS Version.

y axis Vertical axis on a forms grid.

Index

D

S

Z