## Related publications

The Xerox 4635 Laser Printing System PDL/DJDE Reference is part of the ten-manual reference set for your laser printing system. The entire reference set is listed in the table below. Several other related documents are also listed for your convenience. For a complete list and description of available Xerox documentation, refer to the Xerox Documentation Catalog (Publication number 610P17417) or call the Xerox Documentation and Software Services (XDSS) at 1-800-445-5554.

<table>
<thead>
<tr>
<th>Publication</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xerox Dynamic Document Interface Operator Guide</td>
<td>720P13670</td>
</tr>
<tr>
<td>Xerox Dynamic Document Interface Command Summary</td>
<td>720P13680</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Operator Guide</td>
<td>721P83010</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Operations Reference</td>
<td>721P83020</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Sysgen Guide</td>
<td>721P83030</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Message Guide</td>
<td>721P83040</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System PDL/DJDE Reference</td>
<td>721P83050</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Forms Creation Guide</td>
<td>721P83060</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Product Reference</td>
<td>721P83070</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Installation Planning Guide</td>
<td>721P83080</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System Operator Command Summary Card</td>
<td>721P83090</td>
</tr>
<tr>
<td>Xerox 4635 Laser Printing System PC UI Reference</td>
<td>721P83100</td>
</tr>
<tr>
<td>Xerox Laser Printing Systems Tape Formats Manual</td>
<td>600P86175</td>
</tr>
<tr>
<td>Helpful Facts About Paper</td>
<td>721P82490</td>
</tr>
</tbody>
</table>
Notice

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

## Introduction

- Related publications iii
- Notice iv
- About the reference set xxii
- 4635 Laser Printing System document set xxii
- About this manual xxiv
- PDL syntax conventions used in this manual xxv

## 1. Overview

### PDL features and functions 1-1

#### LPS component types 1-2

- Hardware 1-2
- Software 1-2

#### LPS hardware components 1-2

- Advanced Image Subsystem (AIS) 1-2
- Raster image processor (RIP) 1-2

#### LPS software components 1-3

- PDL related programs and tasks 1-5

#### Preparing for a print job 1-6

- Operating system executive task (OSEXEC) 1-5
- File control program (FCP) 1-5
- Operator communication subsystem (OCS) 1-5
- Editor task 1-5
- Font editor task 1-5
- Input processing task 1-5
- Report task (RPT) 1-5
- Dynamic job descriptor (DJD) task 1-6
- Output processing task 1-6
- Print description language (PDL) processor 1-6
- Forms description language (FDL) processor 1-6

### Preparing for a print job 1-6

- Job flow process 1-7
- Input data types 1-8

#### Unformatted data 1-8

- Formatted data 1-10
# 2. Print Description Language (PDL)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose of PDL</td>
<td>2-1</td>
</tr>
<tr>
<td>PDL command structure</td>
<td></td>
</tr>
<tr>
<td>Command line length</td>
<td>2-2</td>
</tr>
<tr>
<td>Command components</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>2-3</td>
</tr>
<tr>
<td>Command keyword</td>
<td>2-3</td>
</tr>
<tr>
<td>Parameter</td>
<td>2-4</td>
</tr>
<tr>
<td>Comments</td>
<td>2-4</td>
</tr>
<tr>
<td>Syntax rules</td>
<td>2-4</td>
</tr>
<tr>
<td>Right-part constants</td>
<td>2-5</td>
</tr>
<tr>
<td>Value constants</td>
<td>2-5</td>
</tr>
<tr>
<td>String constants</td>
<td>2-6</td>
</tr>
<tr>
<td>Job source library (JSL) structure</td>
<td>2-9</td>
</tr>
<tr>
<td>Command levels</td>
<td>2-10</td>
</tr>
<tr>
<td>ID level</td>
<td>2-11</td>
</tr>
<tr>
<td>System or JDL level</td>
<td>2-11</td>
</tr>
<tr>
<td>Catalog level</td>
<td>2-11</td>
</tr>
<tr>
<td>Job or JDE level</td>
<td>2-12</td>
</tr>
<tr>
<td>Examples-JOB command</td>
<td>2-12</td>
</tr>
<tr>
<td>END command</td>
<td>2-13</td>
</tr>
<tr>
<td>Creating separate files for grouping PDL commands</td>
<td>2-16</td>
</tr>
<tr>
<td>Hierarchy of replacement</td>
<td>2-16</td>
</tr>
<tr>
<td>Hierarchy within a job descriptor library (JDL)</td>
<td>2-16</td>
</tr>
<tr>
<td>Non-JDL hierarchy — START command</td>
<td>2-17</td>
</tr>
<tr>
<td>Start command</td>
<td>2-17</td>
</tr>
<tr>
<td>Points to note</td>
<td>2-20</td>
</tr>
<tr>
<td>Hierarchy of replacement in an errored job descriptor library (JDL)</td>
<td>2-20</td>
</tr>
</tbody>
</table>

# 3. Creating a job source library (JSL)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>3-1</td>
</tr>
<tr>
<td>What your JSL specified for LPS processing</td>
<td>3-1</td>
</tr>
<tr>
<td>Decisions to make before creating your JSL</td>
<td>3-2</td>
</tr>
<tr>
<td>Input data</td>
<td>3-2</td>
</tr>
<tr>
<td>Output specifications</td>
<td>3-3</td>
</tr>
<tr>
<td>Type of application to create</td>
<td>3-3</td>
</tr>
<tr>
<td>Special features</td>
<td>3-4</td>
</tr>
<tr>
<td>Interactions between JSLs, catalogs, and jobs</td>
<td>3-4</td>
</tr>
<tr>
<td>Review of PDL components and syntax</td>
<td>3-4</td>
</tr>
<tr>
<td>Command levels</td>
<td>3-4</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Command components</td>
<td>3-5</td>
</tr>
<tr>
<td>Command identifiers</td>
<td>3-5</td>
</tr>
<tr>
<td>Command keywords</td>
<td>3-5</td>
</tr>
<tr>
<td>Command parameters</td>
<td>3-5</td>
</tr>
<tr>
<td>Comments</td>
<td>3-6</td>
</tr>
<tr>
<td>PDL syntax</td>
<td>3-6</td>
</tr>
</tbody>
</table>

Hints and tips | 3-6 |
| Steps in creating a JSL | 3-8 |

- Using the Editor | 3-8 |
- Name the JDL identifier | 3-8 |
- Specifying VFUs | 3-8 |
- Setting up input parameters | 3-8 |
- Specifying LINE command parameters | 3-9 |
- Specifying ACCT command parameters | 3-9 |
- Specifying use of DJDEs | 3-9 |
- Adding logical processing specifications | 3-9 |
- Specifying formats | 3-10 |
- Using copy modification entries | 3-10 |
- Defining paper requirements | 3-10 |
- Specifying output requirements | 3-11 |
- Ending a JSL | 3-11 |
- Finished JSL | 3-11 |

Compiling the JSL | 3-13 |

Printing the job | 3-14 |

Page considerations | 3-15 |

- Paper sizes | 3-15 |
- System page | 3-15 |
- Physical page | 3-15 |
- Edgemarking | 3-15 |
- Non-imaged elements | 3-15 |

Page orientations | 3-16 |

- Landscape orientation | 3-17 |
- Portrait orientation | 3-18 |

Registration shift and skew | 3-19 |

Fonts | 3-21 |

- Font and graphic memory | 3-22 |
- 4635 compatibility with the 4850, 4135 and 4050/4090/4650 LPS | 3-22 |
- Xerox 4850 and 4890 HighLight Color LPS | 3-24 |
- Running 4850/4890 applications on your 4635 LPS | 3-24 |
4. Specifying input parameters

Overview

BLOCK command

BLOCK command parameters
ADJUST
CONSTANT
FORMAT
LENGTH
LMULT
LTHFLD
OFFSET
POSTAMBLE
PREAMBLE
ZERO
Points to note—BLOCK command
Online versus offline JDLs
Example—BLOCK command usage

CODE command

CODE command parameters
ASSIGN
DEFAULT
Point to note—CODE command
Examples—CODE command usage

PCC command

PCC command parameters
ADVTAPE
ASSIGN
DEFAULT
INITIAL
MASK
Points to note—PCC command
Example—PCC command usage
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD command</td>
<td>4-13</td>
</tr>
<tr>
<td>RECORD command parameters</td>
<td>4-13</td>
</tr>
<tr>
<td>ADJUST</td>
<td>4-14</td>
</tr>
<tr>
<td>CONSTANT</td>
<td>4-14</td>
</tr>
<tr>
<td>FORMAT</td>
<td>4-14</td>
</tr>
<tr>
<td>LENGTH</td>
<td>4-14</td>
</tr>
<tr>
<td>LMULT</td>
<td>4-15</td>
</tr>
<tr>
<td>LTHFLD</td>
<td>4-15</td>
</tr>
<tr>
<td>OFFSET</td>
<td>4-15</td>
</tr>
<tr>
<td>POSTAMBLE</td>
<td>4-16</td>
</tr>
<tr>
<td>PREAMBLE</td>
<td>4-16</td>
</tr>
<tr>
<td>STRUCTURE</td>
<td>4-16</td>
</tr>
<tr>
<td>Points to note—RECORD command</td>
<td>4-17</td>
</tr>
<tr>
<td>Example—RECORD command usage</td>
<td>4-17</td>
</tr>
<tr>
<td>SEFFNT command</td>
<td>4-20</td>
</tr>
<tr>
<td>Mapping files</td>
<td>4-20</td>
</tr>
<tr>
<td>SEFFNT command parameters</td>
<td>4-21</td>
</tr>
<tr>
<td>[label:] SEFFNT SEFMAP</td>
<td>4-21</td>
</tr>
<tr>
<td>SEFMAP</td>
<td>4-21</td>
</tr>
<tr>
<td>MAP</td>
<td>4-22</td>
</tr>
<tr>
<td>Points to note—SEFFNT command</td>
<td>4-22</td>
</tr>
<tr>
<td>Examples—SEFFNT command usage</td>
<td>4-23</td>
</tr>
<tr>
<td>TCODE command</td>
<td>4-24</td>
</tr>
<tr>
<td>TCODE command parameters</td>
<td>4-24</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>4-24</td>
</tr>
<tr>
<td>TASSIGN</td>
<td>4-25</td>
</tr>
<tr>
<td>TRESET</td>
<td>4-26</td>
</tr>
<tr>
<td>Points to note—TCODE command</td>
<td>4-26</td>
</tr>
<tr>
<td>Examples—TCODE command usage</td>
<td>4-27</td>
</tr>
<tr>
<td>VOLUME command</td>
<td>4-28</td>
</tr>
<tr>
<td>VOLUME command parameters</td>
<td>4-28</td>
</tr>
<tr>
<td>BMULT</td>
<td>4-28</td>
</tr>
<tr>
<td>CODE</td>
<td>4-29</td>
</tr>
<tr>
<td>EOV</td>
<td>4-29</td>
</tr>
<tr>
<td>HOST</td>
<td>4-30</td>
</tr>
<tr>
<td>INTERPRESS</td>
<td>4-31</td>
</tr>
<tr>
<td>LABEL</td>
<td>4-32</td>
</tr>
<tr>
<td>LCODE</td>
<td>4-32</td>
</tr>
<tr>
<td>LPACK</td>
<td>4-33</td>
</tr>
<tr>
<td>MAXLAB</td>
<td>4-33</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>MINLAB</td>
<td>4-33</td>
</tr>
<tr>
<td>OPTIMIZE</td>
<td>4-34</td>
</tr>
<tr>
<td>OSCHN</td>
<td>4-35</td>
</tr>
<tr>
<td>OSHDP</td>
<td>4-35</td>
</tr>
<tr>
<td>OSTLP</td>
<td>4-35</td>
</tr>
<tr>
<td>PLABEL</td>
<td>4-36</td>
</tr>
<tr>
<td>R MULT</td>
<td>4-36</td>
</tr>
<tr>
<td>RSAT</td>
<td>4-36</td>
</tr>
<tr>
<td>T CODE</td>
<td>4-37</td>
</tr>
<tr>
<td>UNPACK</td>
<td>4-37</td>
</tr>
<tr>
<td>Points to note—VOLUME command</td>
<td>4-38</td>
</tr>
<tr>
<td>Examples—VOLUME command usage</td>
<td>4-38</td>
</tr>
<tr>
<td>Input sources</td>
<td>4-39</td>
</tr>
<tr>
<td>Online printing systems</td>
<td>4-39</td>
</tr>
<tr>
<td>Channel-attached LPS</td>
<td>4-39</td>
</tr>
<tr>
<td>Online 3211/4245 mode</td>
<td>4-39</td>
</tr>
<tr>
<td>Online-specific commands</td>
<td>4-40</td>
</tr>
<tr>
<td>Creating a JDE or JDL</td>
<td>4-40</td>
</tr>
<tr>
<td>DJDE processing</td>
<td>4-40</td>
</tr>
<tr>
<td>Online optimization</td>
<td>4-41</td>
</tr>
<tr>
<td>Copy-sensitive copy modification entries (CME)</td>
<td>4-41</td>
</tr>
<tr>
<td>Report separation</td>
<td>4-41</td>
</tr>
<tr>
<td>Universal character set buffers (UCSBs)</td>
<td>4-41</td>
</tr>
<tr>
<td>UCSB processing</td>
<td>4-42</td>
</tr>
<tr>
<td>Forms control buffer (FCB)</td>
<td>4-42</td>
</tr>
<tr>
<td>Vertical format control processing</td>
<td>4-42</td>
</tr>
<tr>
<td>Online record length</td>
<td>4-43</td>
</tr>
<tr>
<td>Points to note</td>
<td>4-44</td>
</tr>
<tr>
<td>Online recovery</td>
<td>4-46</td>
</tr>
<tr>
<td>Online dump</td>
<td>4-46</td>
</tr>
<tr>
<td>Starting and ending dump sessions</td>
<td>4-46</td>
</tr>
<tr>
<td>Dump format</td>
<td>4-46</td>
</tr>
<tr>
<td>Points to note</td>
<td>4-47</td>
</tr>
<tr>
<td>Downloading files from the host to the LPS</td>
<td>4-47</td>
</tr>
<tr>
<td>Valid download file types</td>
<td>4-48</td>
</tr>
<tr>
<td>DJDE FILE command</td>
<td>4-48</td>
</tr>
<tr>
<td>Offline mode</td>
<td>4-49</td>
</tr>
<tr>
<td>Tape structure concepts</td>
<td>4-49</td>
</tr>
<tr>
<td>Host computer tape formats</td>
<td>4-49</td>
</tr>
<tr>
<td>Tape codes</td>
<td>4-49</td>
</tr>
</tbody>
</table>
### 5. Defining clusters

- **Cluster features**
- **Cluster processing overview**
- **What clusters do for the programmer and operator**
- **Where clusters are stored**
- **How applications use clusters**
  - Simple and OTEXT applications
  - Stockset applications
- **Mixing applications**
- **Defining clusters and stocksets with PDL and DJDE**
  - PDL commands
  - DJDEs
  - Points to note
  - Steps for creating clusters
  - Keeping stockset changes to a minimum
  - Using clusters with ordered stocks

### 6. Print format commands

- **Overview**
- **ABNORMAL command**
  - Abnormal command parameters
  - ERROR
  - IMISMATCH
  - ISUBSTITUTE
  - OTEXT
  - PAGES
  - REP
  - SECURITY
  - Recovery
  - Marker page
- **ACCT command**
  - Accounting data on report basis
  - Overall usage accounting
  - Installation accounting report
  - ACCT command parameters
  - DEPT
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>USER</strong></td>
<td>6-14</td>
</tr>
<tr>
<td>Points to note—ACCT command</td>
<td>6-15</td>
</tr>
<tr>
<td>Example—ACCT command usage</td>
<td>6-15</td>
</tr>
<tr>
<td><strong>BARCODE</strong></td>
<td>6-15</td>
</tr>
<tr>
<td>Points to note—BARCODE command</td>
<td>6-16</td>
</tr>
<tr>
<td><strong>ac:CME command</strong></td>
<td>6-17</td>
</tr>
<tr>
<td>Short form CME specifications</td>
<td>6-17</td>
</tr>
<tr>
<td>Cataloged CMEs</td>
<td>6-17</td>
</tr>
<tr>
<td>CME parameters</td>
<td>6-18</td>
</tr>
<tr>
<td>CONSTANT</td>
<td>6-18</td>
</tr>
<tr>
<td>FONTS</td>
<td>6-18</td>
</tr>
<tr>
<td>LINE</td>
<td>6-19</td>
</tr>
<tr>
<td>POSITION</td>
<td>6-19</td>
</tr>
<tr>
<td>Points to note—CME command</td>
<td>6-20</td>
</tr>
<tr>
<td>Examples—CME command usage</td>
<td>6-21</td>
</tr>
<tr>
<td><strong>EXPORT command</strong></td>
<td>6-21</td>
</tr>
<tr>
<td>EXPORT command parameters</td>
<td>6-22</td>
</tr>
<tr>
<td>SEPARATORS</td>
<td>6-22</td>
</tr>
<tr>
<td>SNUMBER</td>
<td>6-23</td>
</tr>
<tr>
<td>SPLIT</td>
<td>6-24</td>
</tr>
<tr>
<td>SRECOVER</td>
<td>6-24</td>
</tr>
<tr>
<td>TIMING</td>
<td>6-25</td>
</tr>
<tr>
<td>Points to note—EXPORT command</td>
<td>6-25</td>
</tr>
<tr>
<td>Examples—EXPORT command usage</td>
<td>6-26</td>
</tr>
<tr>
<td><strong>LINE command</strong></td>
<td>6-26</td>
</tr>
<tr>
<td>LINE command parameters</td>
<td>6-27</td>
</tr>
<tr>
<td>DATA</td>
<td>6-27</td>
</tr>
<tr>
<td>FCB</td>
<td>6-27</td>
</tr>
<tr>
<td>FONTINDEX</td>
<td>6-28</td>
</tr>
<tr>
<td>MARGIN</td>
<td>6-28</td>
</tr>
<tr>
<td>OVERPRINT</td>
<td>6-29</td>
</tr>
<tr>
<td>PCC</td>
<td>6-29</td>
</tr>
<tr>
<td>PCCTYPE</td>
<td>6-30</td>
</tr>
<tr>
<td>UCSB</td>
<td>6-30</td>
</tr>
<tr>
<td>VFU</td>
<td>6-31</td>
</tr>
<tr>
<td>Points to note—LINE command</td>
<td>6-31</td>
</tr>
<tr>
<td>Examples—LINE command usage</td>
<td>6-33</td>
</tr>
<tr>
<td><strong>MESSAGE command</strong></td>
<td>6-34</td>
</tr>
<tr>
<td>MESSAGE command parameters</td>
<td>6-34</td>
</tr>
<tr>
<td>ITEXT</td>
<td>6-35</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>OTEXT</td>
<td>6-35</td>
</tr>
<tr>
<td>BTEXT</td>
<td>6-36</td>
</tr>
<tr>
<td>Parameter rules</td>
<td>6-39</td>
</tr>
<tr>
<td>Points to note</td>
<td>6-40</td>
</tr>
<tr>
<td>Points to note—Audit Log</td>
<td>6-42</td>
</tr>
<tr>
<td>OUTPUT command parameters</td>
<td>6-45</td>
</tr>
<tr>
<td>BFORM</td>
<td>6-47</td>
</tr>
<tr>
<td>COLLATE</td>
<td>6-47</td>
</tr>
<tr>
<td>COPIES</td>
<td>6-48</td>
</tr>
<tr>
<td>COVER</td>
<td>6-48</td>
</tr>
<tr>
<td>CYCLEFORMS</td>
<td>6-49</td>
</tr>
<tr>
<td>DENSITY</td>
<td>6-50</td>
</tr>
<tr>
<td>DESTINATION</td>
<td>6-50</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>6-50</td>
</tr>
<tr>
<td>FACEUP</td>
<td>6-51</td>
</tr>
<tr>
<td>FEED</td>
<td>6-51</td>
</tr>
<tr>
<td>FORMAT</td>
<td>6-52</td>
</tr>
<tr>
<td>FORMS</td>
<td>6-52</td>
</tr>
<tr>
<td>GRAPHICS</td>
<td>6-53</td>
</tr>
<tr>
<td>INVERT</td>
<td>6-54</td>
</tr>
<tr>
<td>IMAGE</td>
<td>6-54</td>
</tr>
<tr>
<td>MODIFY</td>
<td>6-55</td>
</tr>
<tr>
<td>NTO1</td>
<td>6-55</td>
</tr>
<tr>
<td>NUMBER</td>
<td>6-56</td>
</tr>
<tr>
<td>OFFSET</td>
<td>6-57</td>
</tr>
<tr>
<td>OSTK</td>
<td>6-57</td>
</tr>
<tr>
<td>PAPERSIZE</td>
<td>6-58</td>
</tr>
<tr>
<td>PURGE</td>
<td>6-59</td>
</tr>
<tr>
<td>RESOLUTION</td>
<td>6-59</td>
</tr>
<tr>
<td>SF1FUNCTION</td>
<td>6-60</td>
</tr>
<tr>
<td>SF2FUNCTION</td>
<td>6-60</td>
</tr>
<tr>
<td>SHIFT</td>
<td>6-60</td>
</tr>
<tr>
<td>SIZING</td>
<td>6-61</td>
</tr>
<tr>
<td>STOCKS</td>
<td>6-62</td>
</tr>
<tr>
<td>SYSPPR</td>
<td>6-63</td>
</tr>
<tr>
<td>TMODE</td>
<td>6-63</td>
</tr>
<tr>
<td>TRANS</td>
<td>6-64</td>
</tr>
<tr>
<td>UNITS</td>
<td>6-64</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>6-65</td>
</tr>
</tbody>
</table>
## TABLE OF CONTENTS

### XMP
- Points to note—OUTPUT command: 6-66
- Examples—OUTPUT command usage: 6-75
- Multiple logical pages on physical page: 6-79

### PDE command
- BEGIN: 6-80
- FONTS: 6-80
- PMODE: 6-81
- Points to note—PDE command: 6-81
- Examples—PDE command usage: 6-82

### ROUTE commands
- Cataloged RTEXT files: 6-83
- ROUTE command parameters: 6-83
- RFORM: 6-84
- RTEXT: 6-84
- Example—ROUTE command usage: 6-85

### STOCKSET command
- STOCKSET command parameters: 6-86
- ASSIGN: 6-86
- INFEED: 6-86
- SYSPAGE: 6-87
- Point to note—STOCKSET command: 6-87
- Example—STOCKSET command usage: 6-87

### VFU command
- VFU command parameters: 6-89
- ASSIGN: 6-89
- BOF: 6-89
- TOF: 6-90
- Points to note—VFU command: 6-90
- Example—VFU command usage: 6-90

## 7. Using logical processing

### Overview
- Logical processing command format: 7-1
- Logical processing commands with TEST parameters: 7-2

### CRITERIA command
- CHANGE: 7-3
- CONSTANT: 7-4
- LINENUM: 7-4
- Test expressions: 7-4
  - Specifying one CRITERIA command: 7-5
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying two CRITERIA command</td>
<td>7-5</td>
</tr>
<tr>
<td>CONSTANT mode</td>
<td>7-6</td>
</tr>
<tr>
<td>CHANGE mode</td>
<td>7-6</td>
</tr>
<tr>
<td>LINENUM parameter</td>
<td>7-6</td>
</tr>
<tr>
<td>Combining CHANGE and CONSTANT modes</td>
<td>7-6</td>
</tr>
<tr>
<td>Points to note—CRITERIA command</td>
<td>7-7</td>
</tr>
<tr>
<td>Examples—CRITERIA command usage</td>
<td>7-7</td>
</tr>
<tr>
<td>String comparison concepts</td>
<td>7-8</td>
</tr>
<tr>
<td>String comparisons</td>
<td>7-8</td>
</tr>
<tr>
<td>Character types</td>
<td>7-8</td>
</tr>
<tr>
<td>Masked comparisons using default type assignments</td>
<td>7-9</td>
</tr>
<tr>
<td>Masked comparisons using non-default type assignments</td>
<td>7-9</td>
</tr>
<tr>
<td>BANNER command</td>
<td>7-10</td>
</tr>
<tr>
<td>HCOUNT</td>
<td>7-11</td>
</tr>
<tr>
<td>HIJOBNO</td>
<td>7-11</td>
</tr>
<tr>
<td>HRPTNA</td>
<td>7-12</td>
</tr>
<tr>
<td>TCOUNT</td>
<td>7-12</td>
</tr>
<tr>
<td>TEST</td>
<td>7-12</td>
</tr>
<tr>
<td>TYPE</td>
<td>7-13</td>
</tr>
<tr>
<td>Points to note—BANNER command</td>
<td>7-13</td>
</tr>
<tr>
<td>Examples—BANNER command usage</td>
<td>7-13</td>
</tr>
<tr>
<td>BSELECT and BDELETE commands</td>
<td>7-14</td>
</tr>
<tr>
<td>TEST—BSELECT and BDELETE commands</td>
<td>7-14</td>
</tr>
<tr>
<td>TEST</td>
<td>7-14</td>
</tr>
<tr>
<td>Points to note—BSELECT and BDELETE commands</td>
<td>7-15</td>
</tr>
<tr>
<td>Examples—BSELECT and BDELETE commands usage</td>
<td>7-15</td>
</tr>
<tr>
<td>RAUX command</td>
<td>7-16</td>
</tr>
<tr>
<td>TEST</td>
<td>7-16</td>
</tr>
<tr>
<td>Points to note—RAUX command</td>
<td>7-16</td>
</tr>
<tr>
<td>Example—RAUX command usage</td>
<td>7-17</td>
</tr>
<tr>
<td>RBAR command</td>
<td>7-17</td>
</tr>
<tr>
<td>RFEED command</td>
<td>7-17</td>
</tr>
<tr>
<td>TEST</td>
<td>7-17</td>
</tr>
<tr>
<td>Points to note—RFEED command</td>
<td>7-18</td>
</tr>
<tr>
<td>RSELECT and RDELETE commands</td>
<td>7-18</td>
</tr>
<tr>
<td>TEST</td>
<td>7-18</td>
</tr>
<tr>
<td>Points to note—RSELECT and RDELETE commands</td>
<td>7-19</td>
</tr>
<tr>
<td>Example—RDELETE command usage</td>
<td>7-19</td>
</tr>
<tr>
<td>ROFFSET command</td>
<td>7-19</td>
</tr>
<tr>
<td>PASSES</td>
<td>7-20</td>
</tr>
</tbody>
</table>
## TABLE OF CONTENTS

- **TEST** 7-20
  - Points to note—ROFFSET command 7-21
  - Example—ROFFSET command usage 7-21

**RPAGE command** 7-21
- RPAGE command parameters 7-22
- SIDE 7-22
- TEST 7-22
- WHEN 7-23
- Points to note—RPAGE command 7-23

**RSTACK command** 7-26
- RSTACK delimiter modes 7-26
- Delimiter display 7-27
- Delimiter on accounting page 7-27
- Status display 7-27
- Online RSTACK usage 7-27
- RSTACK command parameters 7-28
  - ACCTINFO 7-28
  - DELIMITER 7-28
  - HRPTNA 7-29
  - PRINT 7-29
  - TEST 7-30
  - Points to note—RSTACK command 7-30
  - Example—RSTACK command usage 7-30

**RSUSPEND and RRESUME commands** 7-31
- BEGIN 7-32
- TEST 7-32
- Points to note—RSUSPEND and RRESUME commands 7-32
- Example—RSUSPEND and RRESUME commands usage 7-33

**TABLE command** 7-34
- CONSTANT 7-35
- MASK 7-35
- Examples—TABLE command usage 7-35

### 8. Specifying dynamic job descriptor entries (DJD Es)

#### Overview 8-1
- Benefits of using DJDEs 8-1

#### Page- and record-oriented DJDEs 8-2
- Page-oriented 8-2
- Record-oriented 8-4
- IDEN command 8-6
- OFFSET 8-6
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPRINFO</td>
<td>8-6</td>
</tr>
<tr>
<td>PREFIX</td>
<td>8-7</td>
</tr>
<tr>
<td>SKIP</td>
<td>8-7</td>
</tr>
<tr>
<td>Point to note—IDEN command</td>
<td>8-7</td>
</tr>
<tr>
<td>DJDE record specification</td>
<td>8-8</td>
</tr>
<tr>
<td>Application of DJDEs</td>
<td>8-9</td>
</tr>
<tr>
<td>DJDE operator information pages</td>
<td>8-11</td>
</tr>
<tr>
<td>Job parameter modification restrictions</td>
<td>8-11</td>
</tr>
<tr>
<td>Duplex DJDE page printing</td>
<td>8-13</td>
</tr>
<tr>
<td>Effect on multiple logical pages</td>
<td>8-13</td>
</tr>
<tr>
<td>COPIES=processing</td>
<td>8-14</td>
</tr>
<tr>
<td>Online DJDE restrictions</td>
<td>8-14</td>
</tr>
<tr>
<td>DJDE processing optimization</td>
<td>8-14</td>
</tr>
<tr>
<td>DJDE parameter definitions</td>
<td>8-15</td>
</tr>
<tr>
<td>ALTER</td>
<td>8-15</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>8-15</td>
</tr>
<tr>
<td>BSEQ</td>
<td>8-16</td>
</tr>
<tr>
<td>BSIDE</td>
<td>8-17</td>
</tr>
<tr>
<td>BSKIP</td>
<td>8-17</td>
</tr>
<tr>
<td>Points to note—BARCODE command</td>
<td>8-18</td>
</tr>
<tr>
<td>BATCH</td>
<td>8-18</td>
</tr>
<tr>
<td>BEGIN</td>
<td>8-18</td>
</tr>
<tr>
<td>BFORM</td>
<td>8-19</td>
</tr>
<tr>
<td>BOF</td>
<td>8-19</td>
</tr>
<tr>
<td>BTEXT</td>
<td>8-20</td>
</tr>
<tr>
<td>Parameter rules</td>
<td>8-23</td>
</tr>
<tr>
<td>Points to note—BTEXT command</td>
<td>8-23</td>
</tr>
<tr>
<td>C text</td>
<td>8-26</td>
</tr>
<tr>
<td>CANCEL</td>
<td>8-26</td>
</tr>
<tr>
<td>COLLATE</td>
<td>8-27</td>
</tr>
<tr>
<td>COPIES</td>
<td>8-27</td>
</tr>
<tr>
<td>DATA</td>
<td>8-28</td>
</tr>
<tr>
<td>DEPT</td>
<td>8-28</td>
</tr>
<tr>
<td>DESTINATION</td>
<td>8-28</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>8-28</td>
</tr>
<tr>
<td>END</td>
<td>8-29</td>
</tr>
<tr>
<td>FEED</td>
<td>8-29</td>
</tr>
<tr>
<td>FILE</td>
<td>8-29</td>
</tr>
<tr>
<td>FONTINDEX</td>
<td>8-30</td>
</tr>
<tr>
<td>FONTS</td>
<td>8-30</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points to note—FONTINDEX and FONTS</td>
<td>8-31</td>
</tr>
<tr>
<td>FORMAT</td>
<td>8-31</td>
</tr>
<tr>
<td>FORMS</td>
<td>8-32</td>
</tr>
<tr>
<td>Points to note—FORMAT and FORMS</td>
<td>8-32</td>
</tr>
<tr>
<td>GRAPHICS</td>
<td>8-32</td>
</tr>
<tr>
<td>IMAGE</td>
<td>8-33</td>
</tr>
<tr>
<td>INVERT</td>
<td>8-34</td>
</tr>
<tr>
<td>ITEXT</td>
<td>8-35</td>
</tr>
<tr>
<td>JDE</td>
<td>8-35</td>
</tr>
<tr>
<td>JDL</td>
<td>8-35</td>
</tr>
<tr>
<td>MAP</td>
<td>8-35</td>
</tr>
<tr>
<td>MARGIN</td>
<td>8-36</td>
</tr>
<tr>
<td>MODIFY</td>
<td>8-36</td>
</tr>
<tr>
<td>NUMBER</td>
<td>8-36</td>
</tr>
<tr>
<td>OTEXT</td>
<td>8-37</td>
</tr>
<tr>
<td>OVERPRINT</td>
<td>8-37</td>
</tr>
<tr>
<td>PMODE</td>
<td>8-38</td>
</tr>
<tr>
<td>RFORM</td>
<td>8-38</td>
</tr>
<tr>
<td>RTEXT</td>
<td>8-39</td>
</tr>
<tr>
<td>SAVE</td>
<td>8-39</td>
</tr>
<tr>
<td>SEFMAP</td>
<td>8-39</td>
</tr>
<tr>
<td>SEPARATORS</td>
<td>8-40</td>
</tr>
<tr>
<td>SF1FUNCTION</td>
<td>8-41</td>
</tr>
<tr>
<td>SF2FUNCTION</td>
<td>8-41</td>
</tr>
<tr>
<td>SHIFT</td>
<td>8-41</td>
</tr>
<tr>
<td>SIDE</td>
<td>8-42</td>
</tr>
<tr>
<td>NUMBER</td>
<td>8-42</td>
</tr>
<tr>
<td>SPLIT</td>
<td>8-43</td>
</tr>
<tr>
<td>SRECOVER</td>
<td>8-43</td>
</tr>
<tr>
<td>STIMING</td>
<td>8-44</td>
</tr>
<tr>
<td>STOCKS</td>
<td>8-44</td>
</tr>
<tr>
<td>TMODE</td>
<td>8-44</td>
</tr>
<tr>
<td>TOF</td>
<td>8-45</td>
</tr>
<tr>
<td>TRANS</td>
<td>8-45</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>8-45</td>
</tr>
<tr>
<td>Points to note—DJDE parameters</td>
<td>8-45</td>
</tr>
<tr>
<td>Examples—SEFMAP usage</td>
<td>8-47</td>
</tr>
<tr>
<td>DJDE FILE parameter</td>
<td>8-47</td>
</tr>
<tr>
<td>Points to note</td>
<td>8-48</td>
</tr>
<tr>
<td>File effectiveness</td>
<td>8-48</td>
</tr>
</tbody>
</table>
### 9. Using graphics

**Graphics considerations**  
9-1  
**Input for graphics**  
9-1  
**Processing modes**  
9-2  
- Random mode  
9-2  
- Move mode  
9-2  
- Block mode  
9-2  
- Batch mode  
9-3  
**Tape formats**  
9-3  
- Noninterleaved  
9-3  
- Document interleaved  
9-3  
- Page interleaved  
9-4  
- Batch mode  
9-4  
**Online formats**  
9-4  
- Noninterleaved  
9-4  
- Document and page interleaved  
9-4  
- Batch mode  
9-5  
- Document interleaved graphic file transfers  
9-5  
- Management of image files  
9-5  
**PDL command and DJDE options for graphics**  
9-6  
**Performance**  
9-7  
- Random mode  
9-7  
- Online  
9-7  
- Document interleaved file creation  
9-7  
**Restrictions**  
9-7  
- Graphic feature restrictions  
9-7  

### Appendices

A. PDL command and DJDE summary  
A-1  
B. PDL command quick reference  
B-1  
C. Character code assignment tables  
C-1  
D. Offline specifications  
D-1  
E. Editor quick reference  
E-1  

### Glossary

GLOSSARY-1  

### Index

INDEX-1
About the reference set

This document is part of a reference set designed to help you receive maximum benefit from your 4635 laser printing system.

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

4635 Laser Printing System document set

4635 LPS Product Reference

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

4635 LPS Operator Guide

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

4635 LPS PDL/DJDE Reference

- Print Description Language components and processes
- Input processing functions
- Output processing functions
- PDL/DJDE command summary
- Page formatting guidelines
- Character code assignment tables
- PDL/DJDE programming information with step-by-step instructions
• Configuration options  
• Commands  
• OSS software installation, upgrade, and modification procedures

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

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

• PC UI procedures  
• Hierarchy of PC UI windows

• Provides a quick reference of commonly-used operator commands.

• OSS and other messages  
• Meaning and recovery procedures
INTRODUCTION

- Pre-installation site preparation
- Supplies and font ordering
- Documentation and training
- Post-installation requirements

Tape Formats Manual

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

LPS Standard Font Library Font User Guide

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

Helpful Facts About Paper

- Selection and guidelines
- Storage and handling
- Specifications for different printers
INTRODUCTION

About this manual

The 4635 Laser Printing System PDL/DJDE Reference describes the print description language commands and dynamic job descriptor entries used to control the printing of jobs on the Xerox 4635 LPS.

The 4635 Laser Printing System PDL/DJDE Reference is divided into the following chapters and appendices:

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

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

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

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

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

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

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

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

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

Appendix A: PDL commands and DJDE summary.
Appendix B: PDL command and DJDE syntax quick reference.
Appendix C: Character code assignment tables.
Appendix D: Offline specifications.
Appendix E: Editor command quick reference.
PDL syntax conventions used in this manual

inkref or dots Variable names or values are represented in italics.

a|b|c Choices are separated by vertical bars.

{a|b|c} Required choices are enclosed in braces.

[a|b|c] Optional choices are enclosed in brackets.

b Default options are underlined.

... (Ellipsis) indicates repetition of an element, for example:
(form-id₁[,form-id₂][,...])
1. Overview

**PDL features and functions**

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

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

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

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

LPS component types

Hardware

Hardware refers to all the physical components of the LPS. Examples are the tape drives, the PC User Interface, the high-capacity feeder-stackers, and the physical subsystems of the system controller and the printer. Refer to your Xerox 4635 LPS Product Guide and 4635 LPS Operator Guide for detailed descriptions of the hardware components of your LPS.

Software

Software refers to all coded instructions (programs) which are executed by the LPS. Some programs interface with LPS hardware, some with LPS firmware, and some with other software. Examples are the LPS operating system, the PDL compiler, and the output task.

LPS hardware components

Advanced Image Subsystem (AIS)

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

1. In the first step, referred to as “fill”, the video format is stored into a 64-megabit memory called a “page buffer.” The entire page is processed and stored into this memory resulting in a bitmapped image.

2. In the second step, referred to as “dump”, the page buffer contents are sent, one scan line at a time, to the laser.

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

Raster image processor (RIP)

The raster image processor (RIP) is an optional printed wire board which converts Interpress vector data into raster data. RIP makes it possible for the LPS to print complex groupings of data, fonts, forms, graphics, and vector graphics.
LPS software components

The LPS operating system software (OSS), housed in system controller, runs both background and foreground processing. The major programs or “tasks” run by the system controller are shown in their hierarchical relationships in the list below. The OSS tasks are:

- Operating system executive (OSEXEC) task
- File control program (FCP)
- Operator communication subsystem (OCS)
- Diagnostic tasks
- System utilities
- Editor task
- Font editor task
- Input processing task
- Ink catalog search (ICS)
- File transfer (XFR)
- Report processing (RPT)
- Memory control program (MCP)
- Terminal ancillary control program (TACP)
- Ancillary IOT monitor program (AIM)
- Dynamic job descriptor (DJD) task
- Interpress decomposer (IPD) task
- Output processing task
- Print description language (PDL) processor
- Forms description language (FDL) processor
- Host interface processor (HIP) task.

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

- Operating System Executive (OSEXEC)
- File Control Program (FCP)
- Operator Communication Subsystem (OCS)

- Utilities
  - Diagnostics
  - Editor
  - Font Editor
  - FDL Processor
  - PDL Processor

- Input
  - DJD
  - IPD
  - RPT

- Output
  - PDL Processor

- Host Interface Processor (HIP)
PDL related programs and tasks

The programs and tasks most important to PDL are these:

Operating system executive task (OSEXEC)

The operating system executive (OSEXEC) task is always running. It interfaces with the LPS hardware and logs hardware errors. In addition, it queues devices, manages resources, establishes priority for software tasks, and schedules processing.

File control program (FCP)

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

Operator communication subsystem (OCS)

The operator communication subsystem (OCS) task acts as an interface between the operator and software tasks. OCS receives input from the operator and displays messages to the operator. OCS also interfaces between system tasks.

Editor task

The editor task creates and modifies disk files. It uses a temporary work file and stores the contents of the work file permanently on disk when you save the file. It sends files to print and directs CMD files to execute.

Font editor task

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

Input processing task

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

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

Report task (RPT)

Works in connection with the Input task.
OVERVIEW

- Records the disk addresses of the font, form, and image files to be used for the job. The result is a page buffer and a page log for each page to be printed. The page buffer consists of the variable data and print instructions for the page. The page log consists of tracking information used in processing the page.

**Dynamic job descriptor (DJD) task**

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

**Output processing task**

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

**Print description language (PDL) processor**

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

**Forms description language (FDL) processor**

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

**Preparing for a print job**

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

- System administrator loads fonts, logos, and signatures to the printer from tape, floppy, or the host. Refer to the 4635 LPS Operations Reference for detailed information.
- System administrator determines the fonts, forms, and graphics system parameters needed by the job.
- Programmer creates the JSL and FSL source files.
- Programmer invokes the FDL compiler to compile the FSL files into form (FRM) files for the print job.
- Programmer invokes the PDL compiler to compile the source files into CME, TST, PDE, JDL, LIB, and STK object files for the print job.
- Operator resets the system parameters to match the fonts, forms, and graphic requirements of the job.
OVERVIEW

- LPS completes any foreground processing and waits in IDLE status for the operator to enter the START command with the JDE and JDL for the print job.

Job flow process

The steps required to process a print job are summarized below.

OCS processing
- The START command identifies the JDE/JDL created by the programmer for the job. The OCS task reads the START command and notifies the OSEXEC task to load output, input, DJD, and IPD tasks.
- The OCS task then writes a job queue entry (JQE) for the job in the job queue buffer. The JQE includes information from the start command, for example, the JDE and JDL name and the number of copies.

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

Output processing
- The OCS task sends a message to output indicating that output processing can start. This message includes the address in the PRFIL1.SYS file of the information for the first page to be printed.
  - The output task reads the starting page log buffer and builds an entry in its own page tracking buffer. The page tracking buffer contains the information output processing uses for page creation (text, data, form, and font addresses, counts, and flags for options such as simplex or duplex).
OVERVIEW

- The output task copies portions of the font, graphic, and logo files from disk to the Advanced Image Subsystem (AIS). The AIS is instructed to create the page image from these specifications and store it in its private page buffer.

- The output task instructs the AIS to send the page image to the printer. As the AIS sends the image, the output task uses the page tracking buffer to monitor and record delivery of the printed page to the bin.

- The output task identifies each page successfully printed and delivered. The input task can then reuse that page space in the page buffer to process an incoming page.

The output task also identifies each page unsuccessfully printed or delivered. The output task uses the page buffer and log information stored in the PRFIL1.SYS file to repeat the steps which recreate the page image, send it to the printer, and monitor its delivery to the bin. Then the output task releases the space for that page in the PRFIL1.SYS file.

- When the output task has verified the successful printing and delivery of all pages prepared for it by the input task, the output task signals the input task that output processing is waiting for more formatted page and page log buffers.

- When the input task reaches the end of file marker on the input data, the input task messages the OCS task that input processing of the print job is complete.

- The OCS task displays the **JOB XXXX HAS COMPLETED INPUT PHASE** message to the operator.

- When the output task finishes printing, it messages the OCS task that the job has completed printing.

- The OCS task displays the following message to the operator:
  
  **JOB XXX HAS COMPLETED PRINTING.**

Input data types

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

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

- Unformatted
- Formatted

Unformatted data

Unformatted data—that is, raw data from a computer file—is selected, formatted, and combined with form data (boxes, lines, headings, totals, and so forth) and graphics files to produce a printed report.
The raw data can be sent to the LPS offline from magnetic tape, online from a channel-attached host, or through an Ethernet network.

The forms definition commands used to select records and format the printed output can be entered directly from the keyboard display, or from a host- or PC-based forms definition software package.

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

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

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

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

- **Processing data.** Processing data is optional and it allows the operator to control the output of selected reports, or selected copies of a multiple copy report, for cover-to-cover print processing on any job. For example, you may wish to specify that an inventory report has a blue card stock cover, 49 pages of the report, and a blue card stock back cover. You may also decide that four copies without cost information are needed for distribution to clients. The three command sets described below provide output control:

  - **JDE.** Gives the operator control over the mechanics of a particular print job. JDE commands specify the feed and output trays, simplex or duplex printing, stapling, collating, and so forth.

  - **DJDE.** Enables you to modify the printing environment dynamically. These commands are inserted into the input data stream to modify the command characteristics of the existing job descriptor entry (JDE). DJDEs can take effect on a report-to-report, page-to-page, and record-to-record basis.

  - **CME.** Enables you to replace certain parts of a report with predefined static data on selected copies or to specify font changes within the variable data.

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

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

Formatted data is sent to the LPS from a host-based document composition software package, for example, XPPI, XDGI, or PC-based software through the front-end processor. These systems are often used for electronic publishing and can produce very sophisticated printed documents. Data from these sources come in a form that the Xerox 4635 LPS can already understand; there is no need for the operator to create forms definition files or the like, as it is with the unformatted data described above.
2. Print Description Language (PDL)

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

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

Purpose of PDL

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

This compilation process is depicted in figure 2-1.

Figure 2-1. PDL compilation

![Figure 2-1. PDL compilation](image)

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

- Host, using a host-based editing facility to create 80 byte EBCDIC records.
- LPS system controller. Refer to the Xerox 4635 LPS Operations Reference.
PDL command structure

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

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

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

Command line length

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

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

Command components

The components of a PDL command are:

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

These are shown in figure 2-2.

In addition, there are syntax rules you must use in order for the system to recognize and process your JSLs. These rules are described later in this section.
Figure 2-2. Example of PDL command components

<table>
<thead>
<tr>
<th>COMMAND STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFU005: VFU ASSGN = (1,5), TOF = 5, BOF = 66;</td>
</tr>
<tr>
<td>Identifier</td>
</tr>
<tr>
<td>Parameter keyword</td>
</tr>
<tr>
<td>Parameter option</td>
</tr>
<tr>
<td>Additional parameter keywords and their options</td>
</tr>
</tbody>
</table>

In this example, the identifier, parameter keyword, and parameter options are all part of the VFU command, which is represented by VFU, the required command keyword. All of these components may be collectively referred to as a command statement.

**Identifier**

If the command is to be referenced by another command, an identifier must precede the command keyword.

The PDL command below has a command identifier (VFU1), a command keyword (VFU), and three command parameters (ASSIGN, TOF, BOF):

```
VFU1: VFU ASSIGN=(1,1),TOF=1,BOF=55;
END;
```

A command identifier is a label that may consist of one to six alphanumeric characters (A-Z and 0-9). It must be followed by a colon (:). The identifier VFU1 in the command above could be coded with any number of blanks following the VFU1 characters, but can have no blanks within the identifier name.

**Note:** The VFU1 identifier is referenced in the LINE command (LINE VFU =VFU1) of tables 2-1 and 2-2. A command that requires an identifier must always be defined prior to any reference to it.

**Command keyword**

A command keyword is required. For example, CME is the command keyword in table 2-1 and VFU, TABLE, CRITERIA, CME, PDE, and so on are the command keywords in table 2-2. A command keyword is required for each PDL command statement. All command keywords are listed in appendix A.
Parameter

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

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

Table 2-1. Set of typical PDL commands

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Command keyword (required)</th>
<th>Parameter keyword</th>
<th>Parameter option</th>
<th>Additional parameter keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFU1:</td>
<td>VFU</td>
<td>ASSIGN=</td>
<td>(1,1),</td>
<td>TOP=1, BOF=55;</td>
</tr>
<tr>
<td>CME4:</td>
<td>CME</td>
<td>LINE =</td>
<td>(1,60),</td>
<td>POSITION=5, FONT=2;</td>
</tr>
</tbody>
</table>

Comments

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

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

/*comment
/*nested comment*/
*/

Syntax rules

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

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

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

\texttt{BFORM = (SMLFRM,,2)}

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

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

\texttt{END; END;}

\textbf{Example}

\texttt{LINE VFU=VFU1, DATA=(1,10), OVERPRINT=(PRINT,DISP);}

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

\section*{Right-part constants}

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

\textbf{Value constants}

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

\texttt{PDE BEGIN=(1.1, .37);}
\texttt{BLOCK LENGTH=1320;}
\texttt{RECORD LENGTH=132;}
\texttt{IMAGE=(1.30 CM, 0.85 IN);}
String constants

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

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

Keyword

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

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

Variable name

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

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

Hexadecimal

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

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

Character

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

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

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

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

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

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

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

is equivalent to

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

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

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

EBCDIC

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

For example:

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

is equivalent to the hexadecimal

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

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

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

Octal

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

```
BLOCK CONSTANT=O'07070707'
```
H2 and H6 constants generate H2000 BCD and H6000 BCD codes, respectively. Use of H2 and H6 is identical to use of E and A prefixes described above. For example:

```
BLOCK CONSTANT=H2'373737'
BLOCK CONSTANT=H6'373737'
```

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

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

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

is equivalent to:

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

Other examples of the use of a repeat count are:

```
T1: TABLE CONSTANT=(3)'027';
T2: TABLE CONSTANT=(4)'C1';
```

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

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

Job source library (JSL) structure

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

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

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

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

The list below outlines the command levels and some typical specifications that are included in these various levels.

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

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

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

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

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

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

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

System or JDL level

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

The SYSTEM command has the form:

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

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

For example:

```
SAMPL: SYSTEM;
```

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

Catalog level

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

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

<table>
<thead>
<tr>
<th><strong>cat-name</strong></th>
<th>is a 1 to 6 character alphanumeric identifier of which at least one character must be alphabetic. The cat-name is referenced by JDEs after the CATALOG command set has been defined.</th>
</tr>
</thead>
</table>

For example:

```
POWER:CATALOG;
```

In this command, POWER is the catalog level identifier to be used in the INCLUDE parameter of a JOB command. For example, to reference the catalog named POWER in a job, the job level command would be:

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

**Job or JDE level**

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

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

<table>
<thead>
<tr>
<th><strong>jde-name</strong></th>
<th>is a 1 to 6 character alphanumeric identifier. It specifies the name of the JDE being defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cat-name</strong></td>
<td>is a 1 to 6 character alphanumeric identifier of a previously defined catalog name.</td>
</tr>
</tbody>
</table>

**Examples-JOB command**

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

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

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

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

```
END; END;
```

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

In figure 2-4, note that `HOST=POWERVS` indicates the source and structure of input data. The HOST command indicates whether the JSL is for an offline or online job. Some HOST parameters can apply to both online and offline hosts. Refer to appendix A for a quick reference.
/* THIS JOB DESCRIPTOR LIBRARY CONTAINS JOB DESCRIPTOR ENTRIES FOR PROCESSING GRASP, POWER AND POWER VS JOB TAPES

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

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

VOLUME          HOST=POWERVS,PLABEL=YES,CODE=ASCII;
BLOCK   LENGTH=2048;
RECORD  LENGTH=136,STRUCTURE=VB,LTHFLD=2,
        ADJUST=0,FORMAT=BIN,PREAMBLE=3;
LINE            DATA=(1,132),PCCTYPE=IBM1403,PCC=(0,NOTRAN),
        OVERPRINT=(PRINT,NODISP),VFU=VFU001;
ACCT            USER=(BIN,TRAY);

/* CATALOG COMMAND SET FOR POWER VERSIONS */

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

/* CATALOG COMMAND SET FOR GRASP */

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;

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

1:JOB;
    VOLUME  HOST=POWERVS,CODE=PEBCDIC;
2:JOB;  INCLUDE=(CATPOW);
    VOLUME  HOST=POWER;
    RECORD LTHFLD=1,PREAMBLE=1,ADJUST=2;
3:JOB;  INCLUDE=(CATPOW);
    VOLUME=HOST=POWERVS,CODE=PEBCDIC;
4:JOB;  INCLUDE=(CATGRP);
    VOLUME  HOST=POWER;
END;
In figure 2-5, note that online systems do not define the BLOCK command.

**Figure 2-5. Sample online job descriptor library**

```plaintext
ONLINE: JDL;
VFU1:VFU
ASSIGN=(1,4),ASSIGN=(2,10),ASSIGN=(3,16),
ASSIGN=(4,22),ASSIGN=(5,28),ASSIGN=(6,34),
ASSIGN=(7,40),ASSIGN=(8,46),ASSIGN=(9,66),
ASSIGN=(10,52),ASSIGN=(11,58),ASSIGN=(12,64),
TOF=4,BOF=66;
VOLUME HOST=IBMONL,
OPTIMIZE=(NCC,NDC,NPR);
LINE PCCTYPE=IBM3211,VFU=VFU1,
UCSB=IGNORE,FCB=IGNORE;
ACCT USER=(BIN,TRAY);
IDEN PREFIX='DJDE',SKIP=7,OFFSET=2,OPRINFO=YES;
/* */
/* TABLES AND CRITERIA */
/* */
T1: TABLE MASK='?',CONSTANT='HE?DE? PAGE';
C1: CRITERIA CONSTANT=(1,11,EQ,T1),LINENUM=(1,10);
T2: TABLE CONSTANT='TRAILER PAGE';
C2: CRITERIA CONSTANT=(0,12,EQ,T2),LINENUM=(1,10);
T3: TABLE CONSTANT='EOJ';
C3: CRITERIA CONSTANT=(0,3,EQ,T3),LINENUM=(1,20);
T4: TABLE CONSTANT='// JOB';
C4: CRITERIA CONSTANT=(0,6,EQ,T4),LINENUM=(1,20);
T5: TABLE CONSTANT=(10)'*';
C5: CRITERIA CONSTANT=(0,10,EQ,T5),LINENUM=(50,10);
/* */
/* JOBS WITH NO BANNER PAGES */
DFLT: JOB;
/* */
/* JOBS WITH HEADER PAGES ONLY */
HDRP: JOB;
BANNER TEST=C1,HCOUNT=2,TCOUNT=0;
/* */
/* JOBS WITH TRAILER PAGES ONLY */
TRLP: JOB;
BANNER TEST=C2,HCOUNT=0,TCOUNT=3;
/* */
/* JOBS WITH BOTH HEADER AND TRAILER PAGES */
BOTH: JOB;
BANNER TEST=(C1 OR C2),HCOUNT=2,TCOUNT=3;
/* */
/* OTHER JOBS */
/* */
EOJ: JOB;
BANNER TEST=C3,TCOUNT=1;
JOB: JOB;
BANNER TEST=C4,HCOUNT=1;
POWER: JOB;
BANNER TEST=C5,HCOUNT=2;
/* */
END;
```
Creating separate files for grouping PDL commands

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

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

Refer to your 4635 LPS Operations Reference for detailed information on using system and editor command files.

Hierarchy of replacement

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

Hierarchy within a job descriptor library (JDL)

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

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

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

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

<table>
<thead>
<tr>
<th>Highest order</th>
<th>Lowest order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DJDE RECORDS*</td>
<td>OSS DEFAULTS</td>
</tr>
<tr>
<td>TAPE LABEL</td>
<td></td>
</tr>
<tr>
<td>START COMMAND*</td>
<td></td>
</tr>
<tr>
<td>JOB/JDE COMMAND LEVEL</td>
<td>Options introduced at higher level override those at lower level</td>
</tr>
<tr>
<td>CATALOG COMMAND LEVEL</td>
<td></td>
</tr>
<tr>
<td>SYSTEM/JDL COMMAND LEVEL</td>
<td></td>
</tr>
</tbody>
</table>

*One exception to this hierarchy is that the COPIES parameter in the START command overrides COPIES DJDE from the input data stream.

**Non-JDL hierarchy — START command**

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

**START command**

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

**Offline**


**Online**

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

**Restrictions**

- The START command parameters are positional and must be separated by commas. You must enter a comma to replace a parameter that is not specified. The options you specify in the START command override those specified in the job descriptor library.

- If a specified font or form file fails a validity check during input, the system aborts the job and displays the following messages:

  - OS8852: Invalid font file header
  - OS8855: Invalid form file header
• START with parameters does not execute when HIP print jobs are being processed.

• You cannot specify a tape device when you have specified DISC or FORM.

Examples of START command usage follow the parameter definitions below. START command options are as follows:

**Parameters**

- **jde**
  a 1 to 6 character identifier for the job descriptor entry to be used in processing the job. If the jde option is not specified in the START command, the user-specified default (DFLT) of the job descriptor library is used.

- **jdl**
  a 1 to 6 character identifier of the job descriptor library for the print job. It must be listed in the jdl file directory. If the jdl option is not specified, the default is used. The default jdl must have been created by the user with an identifier of DEFAULT in the SYSTEM command.

- **S | M**
  parameters which control the processing mode of a job. The two modes are: single report (S) and multiple report (M). The default mode is multiple report. Single report mode halts the system after the processing of each report to allow the operator to select the job set-up parameters for the next report by entering a new START command. Whenever the S mode is invoked in the START command, the system processes only one report at a time, and the accounting sheet will call the report “REPORT 1.” If the REPORTS option of the START command is invoked at the same time as the S mode, the REPORT option takes precedence over the S mode option. If multiple reports are listed in the REPORTS option, all the reports are processed in multiple report mode, except the last report, which is processed in single report mode. Multiple report mode allows all reports in all files to be processed continuously. Processing automatically sequences from report to report, file to file, volume to volume until all reports have been processed.

- **copies**
  prints a specified number of copies of a report. It is an override of the value specified in the job descriptor entry, and also overrides a DJDE command if present in the input data stream.

- **REPORTS:**
  allows you to specify the sequence and subset of reports to be processed. This option is available only when processing tape input. Only reports specified are printed. For r1,r2,..., the user specifies numeric values or ranges of
values, representing the print order of reports. A range is specified as n-m, where n and m are the first and last reports in the range to process, respectively. For example, entering REPORTS: 6,1-3,5,4 causes the sixth report to print first, followed by the first through third, followed by the fifth and then the fourth. If the job contains more than six reports, they are not be processed. The REPORTS option is also useful for printing one report of a multiple report tape. This saves the step of spacing over reports not needed. A maximum of 14 values is allowed. A range of values uses 2 of the 14 maximum values allowed.

TDn specifies the tape device being used.

DISC indicates you want to print a file stored on the system disk.

FORM allows you to specify a form used for the job. This option overrides any form specification in the job descriptor entry (FORMS command of OUTPUT statement and the RFORM command of the ROUTE statement). The specified form must be on disk in the FRM directory.

Example 1

START J12,H2SYS

This command starts a print job using the H2SYS job descriptor library and the job descriptor entry J12. It runs in multiple report mode (by default) and prints the number of copies as specified in the J12 job descriptor entry. The job descriptor library, H2SYS, must reside in the JDL directory. After the START command is initiated, several messages display to inform the operator of the print jobs in progress. In most instances, only one more command entry is required. An example of an offline interaction is shown in Figure 2-7 below.

Figure 2-7. Starting an offline print job

OS1000 READY FOR COMMANDS hh:mm:ss
START J12,H2SYS
OS1010 Starting job 00003
OS2010 Mount input tape; “CONTINUE I” when ready
CONTINUE I
OS0010 Resuming INPUT
OS0020 Resuming OUTPUT
OS1020 Job 00003 has completed input phase
OS1030 Job 00003 has completed printing
OS1000 READY FOR COMMANDS hh:mm:ss
Example 2

START J12,H2SYS,,5

This command is the same as in example 1 with the exception that five copies are requested. The value of 5 entered for copies overrides the value specified in the J12 job descriptor entry. Note that a comma replaces the unspecified mode option; therefore, the default mode, multiple report, takes effect.

Example 3

START

Since no options are specified, the START command defaults take effect. The default for the job descriptor library is DFAULT, which must exist in the JDL directory. The job descriptor entry used is DFLT, which must exist in the DFAULT JDL. The command START,DFAULT has the same effect.

Example 4

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

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

Example 5

START,ONLINE,,,FORM=GBAR

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

Final modification can be made to job descriptor parameters through DJDE parameters in the job input data stream. These DJDE parameters are discussed in chapter 8, “Specifying dynamic job descriptor entries (DJDEs).”

Points to note

• “INTERPRESS” is not a valid keyword
• FORM option is only valid online.

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

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

Example:

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

SAM2:SYSTEM;
VOLUME CODE=ASCII;
AA:CATALOG;
VOLUME CODE=EBDIC;
OUTPUT COPIES=100;
JOB1:JOB INCLUDE=(AA);
OUTPUT COPIES=50;
END;
3. Creating a job source library (JSL)

Overview

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

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

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

Please keep in mind that many of the contents of this chapter are samples and suggestions of what can be done with PDL to create your desired applications, that there is a multitude of other possibilities and options available, and that detailed information on each topic is provided in the other chapters of this manual, and in the Xerox 4635 LPS Forms Creation Guide.

What your JSL specified for LPS processing

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

You then code jobs in job source language (JSL). Your JSL files may call FRM files.
If your document uses color, the PDL applications you code must include the appropriate Xerox 4850 or 4890 HighLight Color LPS ink specifications. On your Xerox 4635 LPS, a color JSL is compiled without errors, but the color ink is ignored and the application prints in black, the system default.

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

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

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

**Decisions to make before creating your JSL**

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

**Input data**

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

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

Output specifications

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

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

Type of application to create

When planning the specifications you will indicate in your JSL, you must first decide the type of application you want by answering the following questions:
• Will you modify an existing application or create a new one?
• Will you use a form? If so, you need to either call out an existing form in your JSL or create a new form first and then call it out in the JSL.
• Will the application be landscape (horizontal) or portrait (vertical) orientation? (Orientation is discussed later in this chapter.)
• Will you use more than one color or type of stock in the print job?
Special features

You also want to determine the following:

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

Interactions between JSLs, catalogs, and jobs

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

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

Review of PDL components and syntax

Before beginning the formatting of your JSL, a quick review of PDL components may be helpful. These are discussed in more detail in chapter 2, “Print description language components and processes.”

Command levels

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

- A job descriptor library (JDL) command must precede all of the levels. It provides the name of the JDL, which is the compiled or “object” form of the JSL.
- ID, system, and catalog level commands can be referenced by commands following them. For this reason, these commands must be numbered or named.

**Command components**

These are the components of a PDL command:
- Command identifier
- Command keyword
- One or more parameters
- Comments, if desired

**Command identifiers**

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

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

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

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

**Command keywords**

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

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

**Command parameters**

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

- Each parameter consists of a left and a right part separated by an equal sign (=).
- Commas or blank spaces separate parameters.
- Parentheses are used to enclose multiple right parts of a parameter, for example:

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

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

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

Example 2
/* CHECK PAPER STOCK FOR EACH JOB*/
LETTER: JDE INCLUDE=SMPL1;
OUTPUT DUPLEX=YES,
COPIES=10,
FORMS=BKUP,
BFORM=NONE;
LABELS: JDE INCLUDE=SMPL1;
OUTPUT DUPLEX=NO,
GRAPHICS=YES;

PDL syntax

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

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

Hints and tips

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

- The only required elements in a JSL are the following:
  - A JDL name, which is the name of the file created by compiling the JSL.
  - Job names
  - END; command at the end of the JSL.
  System defaults could be used for all other specifications,
although typically each application has its own specific characteristics.

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

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

- If you are not sure which specifications to select, try running a job using the system defaults and then adjust the JSL to meet your requirements. You can modify an existing JSL in the same manner.

- Keep the hierarchy of replacement, described in chapter 2, “Print description language (PDL) components and processes,” in mind. It is much easier to specify generic or global characteristics at the system level, for example, than to call the same specifications out over and over again for each job or catalog.

- You do not need to use all command levels in a JSL. Many JSLs have only ID, system, and job-level commands.

- Keep in mind that the specifications to select in your JSL can be changed easily. By using the IDEN command (discussed in chapter 9, “Using graphics”) you can allow DJDEs to override PDL commands on a page-by-page or record-by-record basis. Also, certain operator commands can alter the print job in such areas as the number of copies to be printed, the sequencing of reports, and paper feed specifications.
Steps in creating a JSL

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

Using the Editor

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

Name the JDL identifier

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

XRXSPL: JDL;

Specifying VFUs

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

/* ID level commands*/

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

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

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

Setting up input parameters

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

VOLUME HOST=IBMOS, 
LABEL=STANDARD, 
CODE=EBCDIC, 
PLABEL=YES;

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

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

Specifying LINE command parameters

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

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

Specifying ACCT command parameters

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

ACCT USER=TRAY;

Specifying use of DJDEs

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

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

Refer to appendix A, “PDL/DJDE command summary,” for information on which PDL commands have DJDE counterparts.

Adding logical processing specifications

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

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

Specifying formats

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

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

Using copy modification entries

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

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

Defining paper requirements

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

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

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

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

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

Specifying output requirements

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

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

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

```plaintext
/* Job level commands*/

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

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

The SEFFNT command specifies font mapping.
The EXPORT command specifies how reports will be segmented.
The DESTINATION parameter of the OUTPUT command specifies where printed output is delivered.

Ending a JSL

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

```plaintext
END;
```

Finished JSL

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

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

/* System level commands*/
VOLUME HOST=IBMOS,
        LABEL=STANDARD,
        CODE=EBCDIC,
        PLABEL=YES;
BLOCK LENGTH=2660,
        PREAMBLE=4, LTHFLD=2,
        FORMAT=BIN;
RECORD LENGTH=136,
        PREAMBLE=4,
        STRUCTURE=VB,
        LTHFLD=2, OFFSET=0,
        FORMAT=BIN;
LINE DATA=(1,132),
        PCCTYPE=ANSI,
        PCC=(0,NOTRAN),
        VFU=VFU1;
ACCT USER=TRAY;
IDEN PREFIX='C9700', SKIP=7,
        OFFSET=1;
PDE1:  PDE BEGIN=(0.18 IN,0.66 IN),
        FONT=(LO112B,LO1BOA),
        PMODE=LANDSCAPE;
PDE2:  PDE BEGIN=(1.03,.51),
        BEGIN=(6.30,.51),
        FONT=P1O12A,
        PMODE=PORTRAIT;
CME1:  CME LINE=(9,-),POS=1,FONT=1;
CME2:  CME LINE=3,POS=59,
        CONSTANT='FIRST QUARTER';
SSML:  STOCKSET ASSIGN=('WHITE', CVR),
        ASSIGN=('YELLOW', BLL),
        ASSIGN=('GREEN', SUM),
        INIFEED=CVR,
        SYSPAGE=SUM;
MESSAGE OTEXT=('ALL FORMS DUPLEX ONLY!!!', 1, WAIT),
        ITEXT=('COPY 2 WILL NEED BLUE PAPER');

/* Job level commands*/
JOB1:  JDE;
OUTPUT DUPLEX=Yes,
        SHIFT=Yes,
        FORM=SMPLE,
        MODIFY=CME1,
        FORMAT=PDE1,
        COPIES=10,
CREATING A JOB SOURCE LIBRARY (JSL)

BFORM=SMLBK,
OFFSET=FIRST,
PURGE=NO;

JOB2:  JDE;
       LINE   VFU=VFU2UP;
       OUTPUT FORMAT=PDE2,
                  FORMS=SPL2,
                  COPIES=7;
       END;

Compiling the JSL

Compiling converts the information from a job source file (JSL) into an object file (JDL). This is the command to enter at the LPS PC User Interface to compile the JSL:

PDL [filename [.JSL] [,parameters] ]

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

The characteristics of each of these is described below:

file-name

Specifies the 1 to 6 character name of the JSL file to be input to the PDL task. If no file name is specified, source input for PDL is read from tape, and a JDL file is created on disk for each SYSTEM (or JDL) system command encountered.

file-name.JSL

the file-id which allows you to optionally include the file extension, or file-type, to the file name in the PDL command. If the file extension is used, it must be as shown as JSL. Any deviation aborts the command.
Parameters may be any combination of the following:

**NOPRINT**

Specifies that only source records that contain errors, the diagnostics that apply to those lines, and the PDL summary reports are printed during compilation. If there are no errors, there is no printout. The default is to print all of the above plus the PDL source records.

**NOSOURCE**

Specifies that no source files (JSLs) be created when the input is from tape. This option has no effect when input is not from tape.

**REPLACE**

Specifies that an existing JDL file may be replaced by a new output file of the same name. This is the default.

**NOREPLACE**

Specifies not to create a new JDL file if it has the same name as an existing file. REPLACE is the default.

**TRAY**

Sends the compiled listing of your code and diagnostics to the sample tray. BIN is the default, which delivers the printout to the output tray.

**NRWD**

Specifies that no rewind occurs after a tape file is processed. The default is to rewind.

**DISPLAY**

Displays all PDL messages. The default prints the messages only on the PDL listing.

**TDn**

Tape device (assigned at SYSGEN time) that contains the JSL.

Examples:

```
PDL XEROX, NOSOURCE, NOPRINT
PDL H2SYS, TRAY
```

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

---

**Printing the job**

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

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

The START command specifies the job descriptor library (JDL) and the job descriptor entry (JDE), also called the JOB for printing. Refer to the “Non-JDL hierarchy—START command” section in chapter 2, “Print description language components and processes,” in this manual or to your Xerox 4635 LPS Operator Guide or 4635 LPS Operations Reference, for additional information on the START command.
Page considerations

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

- Paper sizes
- Page orientations
- Registration shift and skew

Paper sizes

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

- Physical page
- System page
- Virtual page

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

System page

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

Physical page

The physical page is the size of the paper itself. You may select any page dimensions within a 8- by 10-inch minimum and a 14.33- by 17-inch maximum. If you have the 7 by 10 option kit installed, then the minimum size is 7 by 10.

Edgemarking

Edgemarking is the placement of marks along the edge of the page. These marks consist of graphic elements that bleed off the paper, tabs for section reference, or marks that denote changes made in redline drafts. Refer to your Xerox 4635 LPS Operations Reference for detailed information on edgemarking.

Non-imaged elements

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

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

• A ruled line near the edge of the system page must be positioned at least one half the line's thickness inside the system page to be printed. If positioned less than one half of the line's thickness, you will get an error message but the system will print the job. Refer to your Xerox 4635 LPS Operations Reference for additional information.

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

Page orientations

There are two types of page orientation:

• Landscape
• Portrait

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

Figure 3-1. Vertical and horizontal positions in landscape and portrait modes
Landscape and portrait orientation dimensions are shown in figures 3-2 and 3-3.

**Landscape orientation**

Figure 3-2. **FMT1**: equivalent impact printer format, 6 lines per inch
Figure 3-3. FMT6: equivalent impact printer format, 6 lines per inch
Registration shift and skew

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

Note: The following figures are the same specifications merely rotated to show portrait and landscape orientations. The shift and skew variances described here are within allowable specifications but, as this can affect the registration of variable data in preprinted forms and the placement of images close to the edge of the page, it is important to make allowance for this condition.
For best results, when designing preprinted forms, allow approximately .1 inch or 2.0 mm space on all sides of any boxes, or above and below any lines onto which variable data is to be printed, as illustrated in figure 3-6.
Figure 3-6. **Maintaining margins in preprinted boxes to allow for LPS registration and skew variations**

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

## Fonts

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

Figure 3-7. **Character spacing**

Because the length of a line printed with a proportional font is relatively unpredictable, fixed fonts are used for variable data on a report to avoid overprinting of forms by variable data. Proportional fonts are normally used for forms data, such as titles and headings. A business letter is an example of the use of proportional fonts for variable data. An example of the difference in line length is illustrated in figure 3-8.
Fonts are available in various families (for example, OCR and Titan), sizes, and faces medium and bold. Refer to the Xerox 4000 Family Laser Printing Systems Font User Guide for more information on fonts and for samples of the font families, sizes, and faces available for use with your Xerox 4635 LPS.

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

- Landscape
- Portrait
- Inverse landscape
- Inverse portrait

Refer to the Xerox 4000 Family Laser Printing Systems Font User Guide for specific font information, the Xerox 4635 LPS Forms Creation Guide for using fonts in a form, and the 4635 LPS Operations Reference for using font editor keyword commands (used to create source font files from existing licensed and non-licensed font files).

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

Font and graphic memory

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

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

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

4635 compatibility with the 4850, 4135 and 4050/4090/4650 LPS

Your Xerox 4635 LPS allows you to process Xerox 4850 Highlight Color LPS print jobs which contain ink references.

The Xerox 4635 LPS is also fully compatible with the 4135 LPS with respect to JSL commands.
The Xerox 4635 LPS prints all 4050, 4090, and 4650 LPS applications successfully.

The functions of the various laser printing systems are listed in table 3-1 and explained in the sections that follow.

**Table 3-1. Software version compatibility**

<table>
<thead>
<tr>
<th>JDL SOURCE</th>
<th>V3.5 4050 4090 4650</th>
<th>V3.6 Rel. 1.5 4135</th>
<th>V3.7 4850</th>
<th>V3.8 4050 4090 4650</th>
<th>V3.9 9790</th>
<th>V4.0 4890</th>
<th>V3A 4635</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downloaded</td>
<td>As Expected</td>
<td>As Expected</td>
<td>As Expected</td>
<td>As Expected</td>
<td>As Expected</td>
<td>As Expected</td>
<td>As Expected</td>
</tr>
<tr>
<td>Compiled on a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V3.5</td>
<td>1</td>
<td>As Expected</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>V3.6 Rel. 1.5</td>
<td>1</td>
<td>3</td>
<td>As Expected</td>
<td>3</td>
<td>1</td>
<td>As Expected</td>
<td>3</td>
</tr>
<tr>
<td>V3.7</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>As Expected</td>
<td>1</td>
<td>As Expected</td>
<td>3</td>
</tr>
<tr>
<td>V3.8</td>
<td>1</td>
<td>As Expected</td>
<td>1</td>
<td>1</td>
<td>As Expected</td>
<td>1</td>
<td>As Expected</td>
</tr>
<tr>
<td>V3.9</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>V4.0</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>V3A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>As Expected</td>
</tr>
</tbody>
</table>

1. Commands unique to this software release are not supported. Unpredictable results will occur.
2. Commands unique to this software release are not supported. However, an error message is displayed, unique commands are ignored, and job will print.
3. Job prints in black only. Two color graphics are not supported.
4. Papersize not supported by the printer. Will cause job to abort.
5. RFEED command is not supported.
6. LOGO command in JSL is not supported.
Xerox 4850 and 4890 HighLight Color LPS

Running 4850/4890 applications on your 4635 LPS

If you have a Xerox 4850 or 4890 version 4.0 HighLight Color LPS, your applications probably include the use of blue, red, or green highlight color. The JDL used to print these applications contain commands and parameters which specify the location and the color ink to be used. Your 4635 LPS will compile, process, and print these jobs, ignoring the ink specified and using the default black ink.

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

PDL Commands and parameters

- ABNORMAL command
  - IMISMATCH parameter
  - ISUBSTITUTE parameter
- CME command
  - INK parameter
- IDR command
  - ICATALOG parameter
  - ILIST parameter
  - PALETTE parameter
- LINE command
  - INKINDEX parameter
- OUTPUT command
  - BFORM parameter
  - CYCLEFORMS parameter
  - FORMS parameter
  - IDFAULT parameter
  - IDR parameter
  - IMAGE parameter
  - IRESULT parameter
  - NUMBER parameter
  - XMP parameter
- VOLUME command
  - INTERPRESS parameter.

DJDEs

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

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

**Downloading 4850/4890 LPS applications to your 4635 LPS**

You can also download color JDLs to your Xerox 4635 LPS. When compiled on your Xerox 4850/4890, these JDLs can then be run on either your 4635 or 4850/4890 LPS. Any ink specifications you have made in the JSL are printed in the default black ink on the 4635; however, they print in the desired inks on your 4850/4890. Logos are not supported on a 4890 or 4635 LPS. Refer to your Xerox 4850/4890 HighLight Color LPS PDL/DJDE Reference.

The only exception to this is the RFEED command, which is supported on the Xerox 4635 LPS, but not on your 4850.

**4850/4890 HighLight Color LPS forms**

The 4635 LPS makes a distinction between color and monochrome forms. It is important to note that forms do not have to contain color-unique forms source library (FSL) commands in order to be classified as color forms. These forms may be generated by either compiling the FSL using the 4850/4890 LPS forms description language (FDL) compiler, converting the monochrome form to color with the File Conversion Utility (FCU) resident on the 4850/4890 LPS, or downloading color forms created from host or third party vendor software packages.

In addition, the 4635 LPS makes a distinction between color and monochrome logos. Due to differences in file formats, the 4635 LPS restricts you from specifying a color form that references a monochrome logo. The opposite is also true. That is, a monochrome form may not reference a color logo. Unlike logos, any form may reference either color or monochrome images.

The 4635 LPS FDL compiler cannot compile an FSL containing color-unique commands. However, this does not imply that the 4635 LPS will not print them. If an FSL were precompiled by a 4850/4890 LPS FDL compiler and its file (FRM) downloaded to a 4635 LPS, the form will print.

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

**Points to note**

• Light tints with isolated pixels of color that print on the Xerox 4850/4890 may not print on the Xerox 4635 LPS.

• Color text printed over solid black background or black text printed over solid color backgrounds on the 4635 LPS may not be visible. Be especially careful in using this format because this situation does not generate displayed or printed messages.
Likewise, color text printed over gray backgrounds or black text printed over color shaded backgrounds on the 4635 printer may not be clearly visible. Be especially careful in using this format because this situation does not generate displayed or printed messages.

- 600 spots per inch (spi) tints and shades printed on the 4635 are finer and more uniform than 300 spi tints and shades printed on the 4850/4890 LPS.

### 4050/4090/4650 LPS

**Creating 4635 LPS applications on a 4050, 4090, or 4650 LPS**

As with any application, you can create 4635 JSLs on your 4050, 4090, or 4650 LPS and then compile them on your 4635 LPS for printing.

**Running 4050, 4090, and 4650 jobs on your 4635 LPS**

The Xerox 4635 LPS offers full backward compatibility with the Xerox 4050, 4090, and 4650 laser printing systems. In other words, all standard JDLs produced on these printers will run successfully on the 4635 LPS. However, it is advisable to recompile them.
Overview

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

The basic PDL commands available to control input processing are BLOCK, CODE, PCC, RECORD, SEFFNT, TCODE, and VOLUME, which are defined in this chapter. Chapter 7, “Using logical processing commands,” defines commands enabling you to specify logical functions that may be performed on either a record, block, or page basis.

Table 4-1 summarizes the input PDL commands.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK</td>
<td>Input data block characteristics</td>
</tr>
<tr>
<td>CODE</td>
<td>Input code translation table</td>
</tr>
<tr>
<td>PCC</td>
<td>Defines printer carriage control code table</td>
</tr>
<tr>
<td>RECORD</td>
<td>Input data record characteristics</td>
</tr>
<tr>
<td>SEFFNT</td>
<td>Defines font mapping for short-edge feed jobs</td>
</tr>
<tr>
<td>TCODE</td>
<td>Marked comparison type assignments</td>
</tr>
<tr>
<td>VOLUME</td>
<td>Input medium characteristics</td>
</tr>
</tbody>
</table>
Table 4-2. Summary of BLOCK command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJUST</td>
<td>Block length adjustment value</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CONSTANT</td>
<td>Block termination code</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Length field recording mode</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Maximum block size</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LMULT</td>
<td>Multiplication factor to determine block length</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LTHFLD</td>
<td>Length of field containing the block length</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>OFFSET</td>
<td>Location of the block length field</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>POSTAMBLE</td>
<td>Length of extraneous data at end of block</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>PREAMBLE</td>
<td>Length of operating system portion of block</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>ZERO</td>
<td>End of block criteria test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

The following describes the syntax of BLOCK command parameters and explanations of the parameter options.

**ADJUST**

Syntax: `BLOCK ADJUST=value`

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

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

Default: Default is 0
CONSTANT

Syntax  
**BLOCK CONSTANT** =  \( sc \)

Function  
Specifies that the block delimiter constant \( sc \) and all data following it are ignored until the end of the block is reached.

Option(s)  
\( sc \) - A string (hexadecimal, octal, or character) constant as described in chapter 3. The length of the constant may be from one to four bytes.

Default  
No default

FORMAT

Syntax  
**BLOCK FORMAT** =  \( type \)

Function  
Specifies the recording mode of the block length field.

Option(s)  
\( type \) - Available type parameters are: BIN (binary), DEC (decimal), PACK (packed with no sign), PKSG (packed with sign).

Default  
Default is BIN

LENGTH

Syntax  
**BLOCK LENGTH** =  \( value \)

Function  
Specifies the longest physical block being processed.

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

Default  
Default is 1330
**LMULT**

Syntax: `BLOCK LMULT= value`  

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

Option(s): `value` - Multiplied by the value in the length field (refer to LENGTH parameter) to compute the number of bytes in the block. A value is an integer in the range of 1 (the default) to 15.

Default: Default is 1

**LTHFLD**

Syntax: `BLOCK LTHFLD= value`  

Function: Specifies the length of the field containing the block length.

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

Default: Default is 0

**OFFSET**

Syntax: `BLOCK OFFSET= value`  

Function: Specifies the location of the block length field.

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

Default: Default is 0
### POSTAMBLE

**Syntax**

`BLOCK POSTAMBLE= value`

**Function**

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

**Option(s)**

`value` - An integer in the range of 0 (the default) to block length.

**Default**

Default is 0

### PREAMBLE

**Syntax**

`BLOCK PREAMBLE= value`

**Function**

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

**Option(s)**

`value` - An integer in the range of 0 (the default) to block length.

**Default**

Default is 0

### ZERO

**Syntax**

`BLOCK ZERO = {YES | NO}`

**Function**

Specifies the end of block indicator.

**Option(s)**

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

**Default**

Default is NO

### Points to note— BLOC K command

- The LENGTH parameter may be overridden by ANSI, IBM OS/Standard, or Honeywell 2000 COBOL labels that specify block length.
- The values for LTHFLD, OFFSET, FORMAT, and PREAMBLE may be overridden if RECORD STRUCTURE is changed as the
result of ANSI, IBM OS/Standard, or Honeywell 2000 COBOL label processing.

- The length on a 4-by-3 packed format tape or Honeywell 600 is the number of 6-bit bytes or characters in the tape block.

- The length of the block delimiter constant should not be coded as the BLOCK POSTAMBLE. Both lengths are subtracted from the end of the block.

- The search for the block delimiter constant starts after the block preamble and proceeds forward to the first appearance of the constant.

- The maximum block size that may be processed by the input task is 24,576 bytes. The input task is able to allocate at least one input buffer for offline tape jobs when the tapes are written in maximum size blocks.

- If a block length is specified which is less than this minimum block length found on the tape, input processing allocates input buffers, which are sized to the minimum tape block length. It is wise to specify the maximum block length in the JSL, so that input buffers are large enough to handle the largest block. However, this may slow performance. If the actual tape block length is smaller than the JSL block length, then no error message is reported; otherwise, an error is displayed.

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

### Online versus offline JDLs

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

#### Example—BLOCK command usage

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

```
BLOCK LENGTH=351,OFFSET=2,ADJUST=0,LTHFLD=2,PREAMBLE=8,FORMAT=BIN;
```
Figure 4-1 shows sample values for BLOCK command parameters.

Figure 4-1. Sample BLOCK command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>User-defined code assignments</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Standard character translation table</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

CODE command parameters

The syntax of CODE command parameters and definitions of the parameter options are shown, as follows.

ASSIGN

Syntax

\[
\text{CODE ASSIGN} = \{ \text{input, output}(\text{input, output1[,output2][,...]}) \}
\]

Function

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

Option(s)

Input - Defines the input code
**SPECIFYING INPUT PARAMETERS**

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

**Default** No default

**DEFAULT**

**Syntax**  
`CODE DEFAULT= code-type`

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

**Option(s)**  
`code-type` - One of the following: EBCDIC, ASCII, BCD, PEBCDIC, H2BCD, H6BCD, IBMBCD, value. A value is a one-byte hexadecimal, octal, or alpha constant. If a value is specified, all inputs are coded to the specified value.

The DEFAULT parameter must be coded prior to any ASSIGN parameter for the assignment exceptions to come into effect. A DEFAULT parameter following any defined corresponding ASSIGN input or output parameter options causes this correspondence to be replaced by the DEFAULT parameter.

**Default** Default is EBCDIC

**Point to note — CODE command**

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

**Examples — CODE command usage**

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

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

**Example 2**  
`CODE DEFAULT=EBCDIC,ASSIGN=(X'5A', (X'40',X'40',X'40'));`

In the latter case, consecutive input codes, such as X'5B' and X'5C', need not be specified to accomplish code modification.
PCC command

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

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

The parameters available for the PCC command are summarized in table 4-4.

Table 4-4. Summary of ac:PCC command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADVTAPE</td>
<td>Printer action for two successive channel skips</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>User carriage control code exceptions</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Standard printer carriage control table</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>INITIAL</td>
<td>Initial reference point for first carriage control</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>MASK</td>
<td>Mask of bits from carriage control byte</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

PCC command parameters

The following information provides PCC command parameter syntax and defines the parameter options.

ADVTAPE

Syntax

ac:PCC ADVTAPE={YES NO}

Function

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

Option(s)

YES - Specifies that multiple skips are honored.

NO - Specifies that multiple skips result in only one skip action being taken. Note that SK1P (skip-to-channel-1-and-print) followed by a second SK1P results in a page transition since printing occurred on the first page (even if blanks were printed).
ASSIGN

Syntax  \texttt{ac:PCC ASSIGN = \{(byte, ccln) (byte, (ccln1, [ccln2] [,...])\}}

Function Specifies printer carriage control exceptions to a default table (refer to DEFAULT parameter).

Option(s) \texttt{byte} - The printer carriage control byte being defined. Its value is in the range 0 to 255 (X00 to XFF).
\texttt{ccln} - Specifies the action that should be taken when the printer carriage control byte defined in \texttt{byte} is encountered. A \texttt{ccln} may be any of the following definitions:

1. \texttt{TOF} - When \texttt{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.

2. \texttt{IGN} - When \texttt{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.

3. \texttt{OVR}, space-print parameters - This option has the following components:
   - \texttt{OVR} - When \texttt{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:
     - Each of the three fields is optional; however, you must specify at least one field. The range for the variables \texttt{m} and \texttt{n} is 0 to 15.
     - \texttt{field\_1} \texttt{field\_2} \texttt{field\_3}

\texttt{field\_1} Specifies the carriage action to take before printing. An example is the ANSI carriage control.
\texttt{SPm} Space \texttt{m} lines before printing.
\texttt{SKn} Skip-to-channel-\texttt{n} before printing.

\texttt{field\_2} Specifies whether printing occurs for this record.
\texttt{P} Print the output data at the line number computed after \texttt{field\_1} is processed.
No printing occurs for this record (default).

Specifies the carriage action to take after printing. An example is the IBM 1403 carriage control.

SPm  Space m lines after printing.
SKn  Skip-to-channel-n after printing.

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 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: Default is OVR

**DEFAULT**

Syntax:  
ac::PCC DEFAULT= \{ ccln \} \{ pcctype \}

Function: Allows selection of a set of printer carriage control codes. A table may be selected and then specific control codes may be modified with the ASSIGN parameters.

Option(s):
  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.

  pcctype - Parameters and the tables that can be modified are: ANSI, B2500, B2700, B3500, B3700, B4700, B6700, H2000, H6000, IBM 1401, IBM 1403, U570, XEROX, NONE.

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

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

Syntax

ac:PCC INITIAL= \{ BOF \ TOF \}

Function
Specification of the initial reference point (TOF or BOF) from which a report performs its first carriage control function.

Option(s)
Specification of the initial reference point (TOF or BOF) from which a report performs its first carriage control function.

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

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

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

Default
Default is TOF

MASK

Syntax

ac:PC MASK= value

Option
value - Specifies an 8-bit value to be ANDed with the printer carriage control byte being processed. ANDing occurs after translation, if any (refer to “Points to note—PCC command,” item 4). The result of this process is to mask off bits from the carriage control byte code which are not relevant to the operation being specified.

Default
Default is X'FF'

Points to note—PCC command

- Multiple user-defined PCC tables are allowed, but only one may be used without a command identifier. The corresponding PCCTYPE parameter on the LINE command references each table through a command identifier. The keyword USER can be used to reference any user-defined PCC table for which no command identifier is coded.
- The user may end a PCC command with a semicolon and start another PCC command to continue specification of the carriage control codes. Multiple PCC commands may be used within a single PCC table definition as long as there are no intervening non-PCC commands.
- Consecutive byte values need not be specified. Thus, the commands
ASSIGN=(X'60',SP1),ASSIGN=(X'61',SP2),ASSIGN=(X'62',SP3);
can be coded in the single command
ASSIGN=(X'60',SP1,SP2,SP3));

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

Example—PCC command usage

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

Added to the standard IBM 1403 carriage control code table are:
X'40' (space 1 line and print), X'F1' (skipto-channel-1-and print),
and X'F8' (print-and-skip to channel 8).

RECORD command

Table 4-5. Summary of RECORD command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJUST</td>
<td>Record length adjust value</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CONSTANT</td>
<td>Record termination code</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Length field recording mode</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Maximum logical record length</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>LMULT</td>
<td>Multiplication factor to determine record length</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LTHFLD</td>
<td>Length of field containing record length</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>OFFSET</td>
<td>Location of record length field</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>POSTAMBLE</td>
<td>Length of extraneous data at end of record</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>STRUCTURE</td>
<td>Input record structure</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

RECORD command parameters

Table 4-5 provides RECORD command parameter syntax and
defines the parameter options.
ADJUST

Syntax  
```
RECORD ADJUST= value
```

Function  Specifies an adjustment value added to or subtracted from the contents of the record length field to determine the true record length.

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

Default  Default is 0.

CONSTANT

Syntax  
```
RECORD CONSTANT= sc
```

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

Option(s)  
```
sc - Specifies a string (hexadecimal, octal, or alpha) constant as described in chapter 3, “Creating a job source library (JSL).” The length of the constant may be from 1 to 4 bytes.
```

Default  No default

FORMAT

Syntax  
```
RECORD FORMAT= type
```

Function  Specifies the format of the record length field.

Option(s)  
```
type - Available type parameters are: BIN (binary), DEC (decimal), PACK (packed with no sign), or PKSG (packed with sign).
```

Default  Default is BIN

LENGTH

Syntax  
```
RECORD LENGTH= value
```
Function: Specifies the length of the longest logical record. A value specifies the length, in bytes, of the longest logical record.

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

Default: Default is 133 offline
Default is 150 online

LMULT

Syntax: RECORD LMULT = value

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

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

Default: Default is 1

LTHFLD

Syntax: RECORD LTHFLD = size

Function: Specifies the length of the field containing the record length.

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

Default: Default is 0

OFFSET

Syntax: RECORD OFFSET = value

Function: Specifies the location of the record length field.
SPECIFYING INPUT PARAMETERS

**Option(s)**

<table>
<thead>
<tr>
<th>Option(s)</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Specifies the record length field offset. This offset is the byte offset from the first byte of the record to the record length field. A value must be an integer in the range 0 (the default) to LENGTH-LTHFLD-1.</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>Default is 0</td>
</tr>
</tbody>
</table>

**POSTAMBLE**

**Syntax**

RECORD POSTAMBLE= length

**Function**

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

**Option(s)**

<table>
<thead>
<tr>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Specifies the length in bytes. The length must be an integer in the range 0 (the default) to record length.</td>
</tr>
<tr>
<td>Default</td>
</tr>
</tbody>
</table>

**PREAMBLE**

**Syntax**

RECORD PREAMBLE= length

**Function**

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

**Option(s)**

<table>
<thead>
<tr>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>- The byte offset from the first byte of the record to the first byte of the user's portion of the record. The length must be an integer in the range 0 (the default) to record length.</td>
</tr>
<tr>
<td>Default</td>
</tr>
</tbody>
</table>

**STRUCTURE**

**Syntax**

RECORD STRUCTURE= structure-type

**Function**

Specifies the general record structure of the input data.

**Option(s)**

<table>
<thead>
<tr>
<th>structure-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>- May be any of the following: 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.</td>
</tr>
<tr>
<td>Default</td>
</tr>
</tbody>
</table>
Points to note—RECORD command

- **LENGTH**
  May be overridden by ANSI, IBM O/S/Standard, or Honeywell 2000 COBOL labels that specify record length.

- **LTHFLD, OFFSET, FORMAT, and PREAMBLE**
  The values for these may be overridden if STRUCTURE is changed as the result of ANSI, IBM O/S/Standard, or Honeywell 2000 COBOL label processing.

- **CONSTANT and STRUCTURE**
  CONSTANT may be enabled as the result of STRUCTURE being changed to U in label processing. However, no definition is assumed for the constant string. The default must be zero, or it must be defined in the job descriptor entry.
  The LENGTH on a 4 by 3 packed format tape is the number of 6-bit bytes or characters in the record.

- **LINE command, DATA parameter, and RECORD command LENGTH parameter**
  When the DATA parameter of the LINE command is changed to a value greater than the default value, the LENGTH parameter of the RECORD command must also be changed accordingly. (Refer to the “LINE command” section of chapter 6, “Specifying output parameters,” for additional information.)
  For online processing, print position indexing must not be used if LENGTH greater than 150 is specified.

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

Example—RECORD command usage

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

```
RECORD LENGTH=133, OFFSET=2, LTHFLD=2, PREAMBLE=4, ADJUST=4, FORMAT=BIN;
```
Figure 4-2 shows sample values for RECORD command parameters.

<table>
<thead>
<tr>
<th>RECORD LENGTH</th>
<th>PCC</th>
<th>PRINT DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>X'01'</td>
<td></td>
</tr>
</tbody>
</table>

**Sample RECORD command**

![Diagram of RECORD command parameters]
Figure 4-3. Record format types

**Fixed Length Record Format**

- Fixed Length Record Format:
  - C: Data
  - One-byte optional vertical format control character
  - Block
  - Unblocked Records: BP Record 1 Gap BP Record 2 Gap BP Record 3
  - Blocked Records: BP Record 1 Record 2 Record 3 Gap BP Record 4 Record 5 Record 6

All records within a file are of the same length. Fixed length blocks may be truncated, but only at the end of a record. Truncated blocks must be an even multiple of the record length plus the block preamble and postamble (if any).

**Variable Length Record Format**

- Variable Length Record Format:
  - Length Indicator: C DATA
  - One byte optional vertical format control character
  - Format "BIN"
  - 2 bytes, length in binary
  - 2 bytes, zeros
  - Format "DEC"
  - 4 bytes, length in decimal characters
  - Block
  - Unblocked Records: BP Record 1 Gap BP Record 2 Gap BP Record 3
  - Blocked Records: BP Record 1 Record 2 Record 3 Gap BP Record 4 Record 5 Record 6

The block preamble may contain a block length field of the same format as the record length field. Blocks may be truncated, but only at the end of a record.

**Undefined Length Record Format**

- Undefined Length Record Format:
  - C: Data
  - One-byte optional vertical format control character
  - Block
  - Unblocked Records: BP Record 1 RD BP Record 2 RD BP Record 3 RD
  - Blocked Records: BP Record 1 RD Record 2 RD Record 3 RD

BP = Optional Block Preamble. The block may optionally contain a postamble.
RD = Record Delimiter. The record delimiter is a string of from 1 to 4 characters (optional for unblocked format).
Gap = Interblock Tape Gap
SEFFNT command

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

The uses of the SEFFNT command parameters are defined in table 4-6.

Table 4-6. Summary of SEFFNT command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>A previously created font mapping</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SEFMAP</td>
<td>Font mapping where the second specified font is the SEF mapping for the first font</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Mapping files

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

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

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

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

The SEFFNT command has different effects depending on the context:
- Bypassing font mapping
- Mapping contained in the JDL
- Standalone mapping files
- JDL creates a mapping file.

SEFFNT command parameters

These contexts and the SEFFNT mapping parameters are described, as follows:

[label:] SEFFNT SEFMAP

Syntax

\[
\text{[label:]SEFFNT SEFMAP} = ((\text{font}_1, \text{font}_2)[,(\text{font}_m, \text{font}_n)])
\]

Function

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

Option(s)

\( (\text{font}_1, \text{font}_2) \) - Font pair, where \( \text{font}_2 \) is the SEF mapping for \( \text{font}_1 \).

\( (\text{font}_m, \text{font}_n) \) - Font pair, where \( \text{font}_n \) is the SEF mapping for \( \text{font}_m \).

Considerations

In example 1 in the “Examples-SEFFNT command usage” section, the file name FNTMAP is created.

Default

No default

SEFMAP

Syntax

\[
\text{SEFMAP} = \left\{ \begin{array}{l}
((\text{font}_1, \text{font}_2)[,(\text{font}_m, \text{font}_n)]) \\
\text{NONE}
\end{array} \right. 
\]

Function

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

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

(font₁, font₂) - Font pair, where font₂ is the SEF mapping for font₁.

(fontₘ, fontₙ) - Font pair, where fontₙ is the SEF mapping for fontₘ.

**NONE** - Disables font mapping and short-edge feeding (SEF); stops previous SEFFNT.

**Default**

No default

---

**MAP**

**Syntax**

MAP= file-name

**Option(s)**

file-name - References a previously created font mapping file from a [label:] SEFFNT command (assumes the label already exists).

**Default**

No default

---

**Points to note—SEFFNT command**

- When the system recognizes a SEF job, it scans the PDE font list and substitutes them with font mappings defined in the JSL. If you specify that font mappings are contained in a file, the input task checks to see if this file exists. If the mapping file or the font file does not exist, job processing continues using the font specified in the PDE command. If no font mapping exists, you receive a **CON I** or **ABO I** message on the screen and have the option to continue processing using the font specified in the PDE or abort the job.

- The system does not assume any mappings for fonts. You must always provide the mapping. Once the font mapping is found, range checking is based on the new orientation. The range checking determines only if the BEGIN values are within the bounds of the page—no check is made to determine whether the line will run off the edge of the page.

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

- Input will use the first font in the PDE list (the list that has been substituted for the new font mappings) for the following system pages:
  - OPRINFO pages (DJDE)
  - Label

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

For landscape printing, however, SEF uses a head-to-head orientation, whereas LEF landscape uses head-to-toe orientation.

• If you want to use 11 by 17 inch paper or any size that requires short-edge feed (SEF), be aware that form source libraries (FSLs), forms (in FRM files), logos (in LGO files), and images (in IMG files) are not automatically scaled or rotated. (Refer to your 4635 LPS Operations Reference for detailed information on scaling.)

• \((\text{font}_1, \text{font}_2)\) - means that the fact that \(\text{font}_1\) is mapped to \(\text{font}_2\) and \(\text{font}_2\) is mapped to \(\text{font}_3\) does not imply that \(\text{font}_1\) is mapped to \(\text{font}_3\).

• \((\text{font}_m, \text{font}_n)\) - means that the fact that \(\text{font}_1\) is mapped to \(\text{font}_2\) and \(\text{font}_2\) is mapped to \(\text{font}_3\) does not imply that \(\text{font}_1\) is mapped to \(\text{font}_3\).

Examples—SEFFNT command usage

**Example 1**

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

FNTMAP.LIB is created on system disk.

**Example 2**

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

**Example 3**

JOB1: JOB;  
OUTPUT FORMAT=PDE1;  

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

**Example 4**

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

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

**Example 5**

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

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

**Example 6**

JOB4: JOB;  
SEFFNT SEFMAP=(L01BOB,P0812A);
Job 4 overrides the system level SEFFNT mapping with a job level definition.

**TCODE command**

The TCODE command defines a set of input-character-to-type assignments to be used for the masked string comparisons in a report. It must be used whenever it is not possible to use one of the standard sets of default type assignments without modification. The uses of the TCODE command parameters are defined in table 4-7.

Table 4-7. **Summary of TCODE command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>Initial set of character type assignments for use with masked comparisons</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TASSIGN</td>
<td>Association of one or more specified characters with one or more specified typecodes</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TRESET</td>
<td>Disassociation of one or more characters, or all characters, in a set from one or more typecodes</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

The TCODE command should be coded as follows:

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

**TCODE command parameters**

TCODE parameter syntax and option definitions are shown in as follows.

**DEFAULT**

**Syntax**

```
ac:TCODE DEFAULT= tcode-type
```

**Function**

Specifies an initial set of character type assignments for use with masked comparisons. Default assignments for any standard character set may be specified by coding the appropriate keyword.
Option(s)  

tcode-type - Available tcode-type keyword parameters are: ASCII, BCD, EBCDIC, PEBCDIC, H2BCD, H6BCD, IBMBCD, and value. Initial typecodes are set to allow the following: all numeric characters (0-9) are type 1 characters, and all uppercase and lowercase alphabetic characters (A-Z, a-z) are type 2 characters. All other characters, if any, are untyped characters.

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

Default  

No default

TASSIGN

Syntax  

ac:TCODE TASSIGN = 

\[
\begin{cases} 
\text{(typespec, 'inputspec')} \\
\text{('inputspec1', ['inputspec2'] [...])} 
\end{cases}
\]

Function  

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

Option(s)  

typespec - A typespec may be coded as a single integer, n. When coded as a single integer, the value may be in the range of 0 to 7. When coded as a series of integers, the values may be within the range of 1 through 7.

inputspec - An inputspec may be coded as a single character, char, as a series of characters, (char[,...]), or as a range of characters, (char1,-,char4). The syntax char1,-,char2 means the set of input codes corresponding to characters bounded by and including char1 and char2 (refer to appendix D for relative order of characters).

When a typespec is coded as a 0, all characters specified by an inputspec are disassociated from all typecode assignments. Thus, the TASSIGN parameter may untype a specified set of characters. When a typespec is coded with a single value (ranging from 1 through 7), all specified characters are associated with that typecode. When a typespec is coded as a series of values (all ranging from 1 through 7), the specified characters are each associated with all of the specified types. A typespec may also be specified with keywords: NUMERIC in lieu of 1 or ALPHA in lieu of 2.

Default  

No default
SPECIFYING INPUT PARAMETERS

TRESET

Syntax

\[
\text{ac:TCODE TRESET=}\left\{\text{\begin{align*}
typespec,\text{'}\text{inputspec}\text{'} \\
('\text{inputspec1}',[\text{inputspec2}',\ldots]) \\
\text{ALL}
\end{align*}}\right\}
\]

Function

Disassociate one or more characters (or all characters) in a set from one or more types.

Option(s)

- **typespec** - The syntax of TRESET is identical to that of the TASSIGN parameter except a typespec of 0 has no effect.
- **inputspec** - When an inputspec is coded with the keyword ALL, all characters in the code table are disassociated from the specified types. The TRESET parameter may be specified more than once in the TCODE command.
- **ALL** - Specifies that all characters in the code table are disassociated from the specified types.

Default

No default

Points to note—TCODE command

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

The examples below illustrate how the TCODE command may be coded to specify the following types of character sets:

<table>
<thead>
<tr>
<th>Type</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Defines numeric characters (0-9) by default.</td>
</tr>
</tbody>
</table>
2 Defines alphabetic characters (A-Z and a-z) by default.

3 Defines uppercase alphabetic characters (A-Z) by TCODE command.

4 Defines lowercase alphabetic characters (a-z) by TCODE command.

None Characters which are neither numeric nor alphabetic.

Examples—TCODE command usage

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

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

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

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

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

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

The # symbol functions as a case switching operator.

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

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

The following provides VOLUME command parameter syntax and defines the parameter options.

**BMULT**

**Syntax**

VOLUME BMULT= value

**Function**

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

**Option(s)**

value
**CODE**

**Syntax**

```
VOLUME CODE= options
```

**Function**

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

**Option(s)**

- **keyword**
  
  Specifies a specific, standard, system-defined code translation table. The available options are: ASCII, BCD, EBCDIC, H2BCD, H6BCD, IBM BCD, 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 inches.

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

**Default**

Default is EBCDIC

**EOV**

**Syntax**

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

**Function**

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

**NO PAUSE**
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.

**NO EOF**
Specifies normal end-of-volume processing.

**Default**
(NO PAUSE, NO EOF)

---

**HOST**

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

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

**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, ICL2900, NCR, OCTDUMP, OLDUMP, O SWTR, 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**

**Default**

Default is IBMOS

**Syntax**

```
VOLUME INTERPRESS= options
```

**Function**

Provides additional control over the Interpress environment.

**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 | NOCHECKSUM]
[BREAKPAGE | NOBREAKPAGE]
[INTEGRAL | FRACTIONAL]
[PERFORMANCE | COMPLIANCE], [RIP | NO RIP]
```

The options have the following components:

**YES**

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

**CHECKSUM | NOCHECKSUM**

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

**BREAKPAGE | NOBREAKPAGE**

- **BREAKPAGE**
  Requests a trailing break page is printing instructions are present in the job.
- **NOBREAKPAGE**
  Specifies a break page only if the system detects errors.

**INTEGRAL | FRACTIONAL**

- **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**

- **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 priority bitmap graphics are imaged on the RIP.

RIP | NO RIP
--- | ---
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

Syntax

```
VOLUME LABEL= label-type
```

Function
Specifies the type of label on the input tape.

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

Syntax

```
VOLUME LCODE= keyword
```

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

Options

```
keyword
```

Specifies a specific, standard, system-defined code translation table. The available options are: ASCII, BCD, EBCDIC, H2BCD, H6BCD, IBM BCD, 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

Syntax

\[ \text{VOLUME LPACK=} \quad \text{options} \]

Function

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

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 required unpacking.

YES

Indicates that the labels are packed.

Default

Default is NO

MAXLAB

Syntax

\[ \text{VOLUME MAXLAB=} \quad \text{value} \]

Function

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

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.

Considerations

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

Default

Default is 81

MINLAB

Syntax

\[ \text{VOLUME MINLAB=} \quad \text{value} \]

Function

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

**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 and must not exceed the BLOCK LENGTH.

**Default**

Default is 80

**OPTIMIZE**

**Syntax**

`VOLUME OPTIMIZE= options`

**Function**

Enhances the performance of the data stream.

**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

**Syntax**

```
VOLUME OSCHN = value
```

**Function**

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

**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 O S Writer banner page is considered found. The default is 9.

**Default**

Default is 9

### OSHDP

**Syntax**

```
VOLUME OSHDP = value
```

**Function**

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

**Options**

```
value
```

Specifies the number of header pages which precede the report. The default is 0.

**Default**

Default is 0

### OSTLP

**Syntax**

```
VOLUME OSTLP = value
```

**Function**

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

**Options**

```
value
```

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

**Default**

Default is 0
PLABEL

Syntax

VOLUME PLABEL= options

Function
Specifies whether the tape labels are printed.

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.

Default
Default is NO

RMULT

Syntax

VOLUME RMULT= value

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

Options

value
An integer from 1 to 15.

RSAT

Syntax

VOLUME RSAT= options

Function
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.

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.

Default  Default is REMOUNT

**TCODE**

Syntax  
```
VOLUME TCODE= options
```

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

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**

Syntax  
```
VOLUME UNPACK = options
```

Function  Specifies the unpacking method of the input data when required.

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.

Default  Default is NONE
Points to note— **VOLUME command**

- The character code assignment tables in appendix C summarize the label specifications that are valid for each host type. If a selected HOST type is inconsistent with a selected LABEL type, an error message is output by PDL, and a valid label is automatically substituted. The LABEL parameter of the VOLUME command should not be coded in cases where it is not appropriate for a particular host type.

- A JDL is supplied with the printing system software to dump the contents of a tape. Six standard tape formats may be dumped with this JDL (H6000 BCD, EBCDIC, ASCII, H2000 BCD, IBM BCD, UNIVAC FIELDATA). Check the listing of the DUMP.JSL file to determine the appropriate JDE for the tape to be dumped. Other tape formats may be dumped by specifying the host-type (DUMP or OCTDUMP) and appropriate parameters, such as CODE and UNPACK, in a user-coded JDL.

- The MAXLAB and MINLAB on a 4 by 3 packed format tape are the number of 6-bit bytes or characters in the tape blocks that are treated as labels.

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

- When running a CODE=NONE job with DUPLEX=YES and SHIFT=YES, MARGIN must be specified in m inches. For example:

  m=--V2-BEGIN-ALI + .01  
  m          Positive decimal number of the form  
  nnn.dd (up to two digits to the right of  
          the decimal point)  
  V2       Even (back) side shift value in inches  
  BEGIN     Scan line in inches as specified in the PDE  
            (hpos in portrait, vpos in landscape)  
  ALI  Anticipated scan alignment (estimated  
       inches equivalent, 3 scans=.01 in.)

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

**Examples— VOLUME command usage**

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

```
VOLUME HOST=UNIVAC, LABEL=STANDARD, UNPACK=T4X3,  
      CODE=ASCII, LCODE=ASCII;
VOLUME HOST=IBMOS;
VOLUME HOST=IBMONL, LABEL=NONE, INTERPRESS=YES, CODE=EBCDIC;
VOLUME HOST=B6700, LABEL=ANSI, BMULT=6, RMULT=6, PLABEL=YES;
```
Input sources

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

Online printing systems

This section describes various features of a Xerox laser printing system that is channel attached to a host computer and explains how you can access these features.

The LPS may have other interfaces as well: remote dialup communication interfaces, an Ethernet communication interface, and SNA/SDLC input through the host interface processor (HIP).

Channel-attached LPS

A laser printing system equipped with the online interface unit (OLI) may be attached to a byte multiplexer, block multiplexer, or selector channel of an IBM 360 or 370, system 3000 or 4300 series, and 9370 models, or of an IBM system 390 ES/9000. The channel-attached LPS operates in one-byte, six-byte, eight-byte, or burst mode. No programming changes to the IBM operating system software are required, provided that an IBM 3211 printer and 3811 printer Control Unit is already supported on the host system.

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

Online 3211/4245 mode

Basic LPS input processing functions are augmented by the following functions for online 3211 or 4245 mode operations:

- Handling of all interactions with the online interface (OLI) hardware.
- Building of translate tables based on the present contents of the universal character set buffer (UCSB) and the current FOLD or UNFOLD command in effect.
• Processing of forms control buffers (FCBs) received from the host to redefine channel-to-line number assignments and to set margins based on the print position indexing byte.

**Online-specific commands**

**Creating a JDE or JDL**

An online job descriptor library (JDL) and its job descriptor entries (JDEs) are created by the user according to the same rules and syntax as other entries. The one PDL command that must be specified for normal online processing is VOLUME HOST=IBM ONL. The HOST parameter of the VOLUME command performs two functions:

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

The defaults which are automatically selected when VOLUME HOST=IBM ONL is specified are as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD</td>
<td>LENGTH = 150</td>
</tr>
<tr>
<td>LINE</td>
<td>PCCTYPE = IBM 3211 or IBM 4245, DATA = (0,150)</td>
</tr>
</tbody>
</table>

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

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

**DJDE processing**

Printing is controlled through parameters from the user-defined JDE file which may be dynamically modified by dynamic job descriptor entries. DJDEs are user created and are processed by the LPS as part of the print data from the host machine. Chapter 8, “Specifying dynamic job descriptor entries (DJDEs),” defines their format and the PDL statements necessary to let the system know they are part of the input data.

Print-and-skip carriage control commands which are associated with online DJDE records are treated as skip-immediate carriage control commands. If the LPS is not at the channel specified, a skip occurs; but if it is already there, no skip occurs unless the last command received with a data record was a print-without-spacing command. Print-and-space carriage control commands associated with DJDE records are ignored.
Nonprint files (for example, font files) can be downloaded from the host using the FILE DJDE.

**Online optimization**

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

**Copy-sensitive copy modification entries (CME)**

Copy modification entries (CMEs) which are copy sensitive (that is, a copy range has been specified on the MODIFY parameter of the OUTPUT command) are ignored online since the data is not automatically available multiple times. To produce copy-sensitive output, the user transmits data the number of times the data needs to be modified. Thus, if six copies of a report are desired and one CME applies to copies 1 through 4, and another CME applies to copies 5 and 6, the report should be transmitted from the host system twice. The first transmission contains a copy count of 4 with CME1 specified, and the second transmission contains a copy count of 2 with CME2 specified. As an alternative, the application may be successfully printed with only one transmission by using copy-sensitive forms. Refer to the FORMS parameter of the OUTPUT command in chapter 6, “Specifying output parameters.”

**Report separation**

Report separation is defined in terms of banner page detection or other user-defined processing criteria. Logically separated reports are physically offset from one another in the output bins. The two logical processing commands which can be used to control online report separation are BANNER and RSTACK. Refer to chapter 7, “Using logical processing commands.”

**Universal character set buffers (UCSBs)**

The IBM 3211/4245 universal character set buffer (UCSB) feature is supported by the LPS. The UCSB is used by the LPS software as a basis for generation of folded and unfolded translate tables. In a folded translate table, uppercase characters are substituted for only those lowercase characters which are defined as unprintable in the UCSB. Note that any lowercase characters defined as printable in the UCSB are still printed as lowercase if available in the font. In an unfolded translate table, blanks are substituted for characters defined as unprintable in the UCSB. The translate table is rebuilt at the start of each report, when a selected JDE/JDL is processed, when a new UCSB is transmitted from the host, and when a new FOLD or UNFOLD command is received.

The selection of a folded or unfolded translate table is based on the FOLD and UNFOLD 3211/4245 commands. If FOLD has been most recently received from the host, a folded translate table is built; if UNFOLD is the current command, an unfolded table is built.

Online users may restrict UCSB information by accepting or suppressing the normal processing of host-transmitted UCSB data. This can be accomplished by the UCSB=IGNORE parameter.
SPECIFYING INPUT PARAMETERS

parameter of the LINE command. If UCSB=IGNORE, lowercase characters print normally if they are present in the font and as blanks if they are not. The translate table is rebuilt only at the start of a report.

The UCSB parameter can be changed from IGNOR E to PROCESS in a selected JDE, but the translate table is not rebuilt until a UCSB load, FOLD or UNFOLD is transmitted from the host.

For most online reports, specification of UCSB=IGNORE is appropriate. The LPS font capability normally negates the need for these translations which were originally defined to handle print train characteristics. If, however, a report using lowercase characters is printed using a font without lowercase, the user may:

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

The first step allows the UCSB to process. The second step causes a folded translate table to be built so that uppercase characters are substituted for lowercase. At the start of the next report, UCSB=IGNORE comes back into effect and the translate table is rebuilt based on the CODE command in the JDE file.

UCSB processing

• When a UCSB is transmitted from the host, it is saved on a PFILR$.$SYS file. When system generation is executed, the system looks for the UCSB saved in the file. If the system does not find the UCSB, it reinitializes the file to define all characters as printable.

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

• UCSB=IGNORE should be specified if a CODE other than EBCDIC is specified. If UCSB=PROCESS, a UCSB load causes the CODE default to revert to EBCDIC until end of report or until a selected JDE/JDL is processed.

Forms control buffer (FCB)

The 3211/4245 forms control buffer (FCB) defines channel positions and forms length. Online users may restrict the forms control buffer information by accepting or suppressing the normal processing of host-transmitted FCB input. This can be accomplished by the FCB=IGNORE parameter of the LINE command.

Vertical format control processing

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

• When a START command is entered, the VFU table and its associated BOF in the JDE come into effect.

• If no VFU is specified in the JDE, the last FCB sent from the host is retrieved from the disk and becomes active.
• If no BOF is specified in the JDE, the length of the last FCB sent from the host is retrieved from the disk and becomes BOF. Note that TOF is not affected by the FCB.
• If FCB=PROCESS and an FCB is sent from the host, it overrides the previous FCB or VFU. Its length replaces the previous BOF. Furthermore, its contents and length are saved on the disk.
• When a DJDE is used to change a channel assignment, any previous assignments to the channel being changed are suspended until the end of report. At the end of report, DJDE changes are discarded and the last active FCB or VFU comes back into effect.
• When sysgen is executed, the FCB saved on disk is reinitialized to the length of 66 lines with the following channel assignments:

<table>
<thead>
<tr>
<th>Channel</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>4</td>
<td>10</td>
<td>16</td>
<td>22</td>
<td>28</td>
<td>34</td>
<td>40</td>
<td>46</td>
<td>52</td>
<td>58</td>
<td>64</td>
<td>66</td>
</tr>
</tbody>
</table>

The default corresponds to the STD2 default supplied by IBM.

• If a print-and-skip-to-channel command is sent for an undefined channel, the LPS executes a print-and-space-one command.
• If FCB=PROCESS, an FCB load causes positioning to TOF. If the system is already at TOF, page eject is suppressed. If the line number prior to the FCB load is greater than TOF, a page eject occurs.
• The print position indexing (PPI) byte of the FCB is not suppressed by FCB=IGNORE. If an FCB with a PPI byte is received, indexing is invoked; if an FCB without a PPI byte is received, indexing is turned off (even if FCB=IGNORE). Use of the PPI byte to shift the starting print position may cause problems in recognition of both DJDEs and banner pages.

**Online record length**

The default record length supported by the Xerox laser printing systems is 150 bytes, the same default as the 3211/4245 printer. However, the you can choose to use larger record lengths with the following restrictions:

• For print record lengths from 151 bytes to 214 bytes, do not use the print position indexing feature that is normally available on the IBM 3211/4245. This feature is invoked only with FCBs; if FCB=IGNORE is invoked, this restriction is not significant.
• For print record lengths from 215 bytes to 2140 bytes, the user must not use the OPTIMIZE parameter of the VOLUME command which provides for online buffering for improved performance.

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

- If the data record transmitted from the host exceeds the RECORD LENGTH specified, the record is truncated to the specified record length and no warning is provided.

  Note: This action is consistent with the IBM 3211/4245 printer.

- The online dump JDL specifies a RECORD LENGTH of 150 bytes. If you are planning to use a longer record length, modify the JDL to dump the data accurately.

- The LENGTH parameter in the RECORD command cannot be changed using a DJDE since this would require dynamic rebuffering. Therefore, you should consider modifications to the default online JDL to simplify operations in the online environment.

- The OPTIMIZE parameter of the VOLUME command can be changed through the DJDE JDE= subject to the restrictions listed in OPTIMIZE.
Figure 4-4. Sample online JDL file

```jdl
ONLINE:  JDL;
VFU1: VFU
ASSIGN=(1,4),ASSIGN=(2,10),ASSIGN=(3,16),
ASSIGN=(4,22),ASSIGN=(5,28),ASSIGN=(6,34),
ASSIGN=(7,40),ASSIGN=(8,46),ASSIGN=(9,66),
ASSIGN=(10,52),ASSIGN=(11,58),ASSIGN=(12,64),
TOF=1,BOF=66;
VOLUME   HOST=IBMONL,
OPTIMIZE=(NCC,NDC,NPR);
LINE     PCCTYPE=IBM3211,VFU=VFU1,
UCSB=IGNORE,FCB=IGNORE;
ACCT     USER=(BIN,TRAY);
IDEN     PREFIX='DJDE',SKIP=7,OFFSET=2,OPRINFO=YES;
/*
* TABLES AND CRITERIA
*/
T1: TABLE  MASK='?',CONSTANT='HE?DE? PAGE';
C1: CRITERIA CONSTANT=(1,11,EQ,T1),LINENUM=(1,10);
T2: TABLE CONSTANT='TRAILER PAGE';
C2: CRITERIA CONSTANT=(0,12,EQ,T2),LINENUM=(1,10);
T3: TABLE CONSTANT='EOJ';
C3: CRITERIA CONSTANT=(0,3,EQ,T3),
    LINENUM=(1,20);
T4: TABLE CONSTANT='// JOB';
C4: CRITERIA CONSTANT=(0,6,EQ,T4),
    LINENUM=(1,20);
T5: TABLE CONSTANT=(10)'*';
C5: CRITERIA CONSTANT=(0,10,EQ,T5),
    LINENUM=(50,10);
/*
* JOBS WITH NO BANNER PAGES
*/
DFLT: JOB;
/*
* JOBS WITH HEADER PAGES ONLY
*/
HDRP: JOB;
BANNER TEST=C1,HCOUNT=2,TCOUNT=0;
/*
* JOBS WITH TRAILER PAGES ONLY
*/
TRLP: JOB;
BANNER TEST=C2,HCOUNT=0,TCOUNT=3;
/*
* JOBS WITH BOTH HEADER AND TRAILER PAGES
*/
BOTH: JOB;
BANNER TEST=(C1 OR C2),HCOUNT=2,TCOUNT=3;
/*
* OTHER JOBS
*/
EOJ: JOB;
BANNER TEST=C3,TCOUNT=1;
JOB: JOB;
BANNER TEST=C4,HCOUNT=1;
POWER: JOB;
BANNER TEST=C5,HCOUNT=2;
/*
END;
*/
```

XEROX 4635 LASER PRINTING SYSTEM PDL/DJDE REFERENCE 4-45
Online recovery

In the event of a system failure, data that is formatted for print and stored on disk can be imaged upon system restart if job recovery is requested. Data which was being received and processed at the time of the rollover is lost and requires retransmission.

The amount of data which requires retransmission depends on two factors:

- The options specified in the VOLUME command OPTIMIZE parameter and DJDEs that indicate the use of interleaved graphics. If NPRECOVER is specified, up to nine data pages are buffered in memory before being written to the disk.
- When end of report is detected, all pages in memory are written to the disk. Therefore, if NPRECOVER is specified, up to nine data pages would require retransmission, but not typically from more than one report. If NPRECOVER is not specified, up to two pages would require retransmission. If interleaved graphics are being transmitted at the time of system failure, the amount of data requiring retransmission depends on the number and size of the graphics.

Note: The number of pages (n) is determined by the number of skip-to-channel-1 commands encountered since the last set of pages were written to the disk.

Online dump

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

Starting and ending dump sessions

The dump session is started with a START command when the LPS is online and in the idle state. The online dump JDE/JDL used must specify VOLUME HOST=OLDUMP and CODE=EBCDIC.

The dump session is terminated by either the operator ENDJOB command or the offline operator command. Other online operator commands during a dump session perform the standard functions.

Dump format

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

The HEADER segment consists of six fields, shown in table 4-9.
Table 4-9. **HEADER segment fields**

<table>
<thead>
<tr>
<th>Segment</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST</td>
<td>English translation of the host command in hexadecimal.</td>
</tr>
<tr>
<td>COMMAND</td>
<td>One byte of status in hexadecimal transmitted to the host.</td>
</tr>
<tr>
<td>END STATUS</td>
<td>Incremental count in decimal of the host commands received during a dump session. The maximum number generated is 999,999.</td>
</tr>
<tr>
<td>SEQ#</td>
<td>Length in decimal of the data transmitted.</td>
</tr>
<tr>
<td>DATE and TIME</td>
<td>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.</td>
</tr>
</tbody>
</table>

The DATA segment consists of the input data in hexadecimal transmitted to or from the host with EBCDIC translation based upon the CODE command. Input data that cannot be translated is left as blanks. Each line of data is preceded by a byte count in hexadecimal.

**Points to note**

- DJDEs, banner, and other logical processing functions are not honored when running in the dump mode, so an operator ENDDJOB command must be entered to end the dump session.
- NO-OP command from the host is not dumped since it is intercepted and processed by the OLI hardware and is transparent to the LPS software.
- Hardware errors are reported in the system error log, but are not dumped due to Host, OLI, or driver interface constraints.
- Commands processed when input is stopped, for example, when paper trays become empty, are not dumped.
- Print records longer than the specified RECORD LENGTH value are truncated without warning. If records longer than 150 bytes are to be sent, the RECORD LENGTH parameter in the online dump JDE/JDL should be modified to the longer length.
- Entering the operator SAMPLE command (with parameters) is not allowed while the system is online, whether or not a job is running. Exceptions to this are: pressing the <SAMPLE> key or entering the operator SAMPLE command with no parameters.
- Operator “HIP START ...” command is accepted, but HIP is unable to submit a print job while the system is online.

**Downloading files from the host to the LPS**

Files may need to be downloaded from the host for various reasons. In a production printing environment, file storage space may be limited by the requirement of a large print file. At some installations, files such as form files which undergo periodic...
revision may be compiled and kept on the host computer and transmitted to the printer when required by a job. System patch tapes may be required. Other LPS font and logo tapes may need to be downloaded.

Three methods of downloading files are available:

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

The HOSTCOPY utility requires the system to be placed offline and idle, then rebooted from disk to enter the operator HOSTCOPY command from the system menu. It does not allow files to be transferred while normal printing activity is taking place.

The DJDE FILE command transfers files while normal printing activity is taking place; however, the file to be transferred may need to be preconditoned to prevent trailing-blank truncation by the host spooler.

A third method of file transfer, using the LPS and host file transfer interface, may be used to transfer user files and also allows an online user to obtain the status of the reports submitted for printing on the LPS. A remote connection between the host and LPS is used for file transfer.

Valid download file types

Files may be downloaded in two formats: card-image files and LPS-labeled files. Certain destination file types are not accepted for either LPS-labeled or card-image files. These are file types OSD, SAF, SYS, $Y$, and TSK. In addition, card-image format files are restricted to destination file types CMD, FSL, JSL, MSC, PCH, TMP, and TPF.

DJDE FILE command

This DJDE provides a capability of loading card-image or LPS-labeled files to the LPS disk while a printing job is in progress. The print job can be invoked solely for the purpose of downloading files; in other situations, the files can be interleaved with variable text data so that they are available when referenced by input or output processing. In either case, a DJDE packet must immediately precede the data for a file (or batch of files) which is copied to the LPS disk. The DJDE packet must include a FILE parameter that is used to signal the start of the file data.

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

Note: The DJDE FILE command does not allow a file replacement or deletion transaction to occur if the file to be replaced or deleted is secured and the logon level is not 5.
Offline mode

Tape structure concepts

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

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

Host computer tape formats

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

Tape codes

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

Packed data formats

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

Whenever an unpacking method is included in the JDL, the system unpacks the characters before data processing. Each 6-bit character is extracted, and two high-order zeros are appended. Normally, after data is unpacked, it must be translated. The character code set is defined in the CODE parameter of the VOLUME command. For a 4-by-3 unpacking method, the data is...
SPECIFYING INPUT PARAMETERS

generally encoded in BCD, and one of the three standard BCD CODE parameters (H2BCD, H6BCD, and IBMBCD) can be used.

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

Record formats

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

Record structure

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

Multivolume processing

All multivolume reports that force input to make multiple passes over the data are handled in the following manner. For each copy requested, a complete pass is made over the group of volumes that make up the current report. The input processor is forced to make multiple passes over the input data for multicopy reports that specify copy-sensitive CMEs, and (optionally) for multicopy reports that exceed the size of the print file. CMEs are discussed in the “Copy modification entries” section in chapter 6, “Specifying output parameters.” The choices available for handling print file saturation are discussed under the VOLUME command RSAT parameter in this chapter.

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

Users with multiple volume jobs and multiple laser printing systems may wish to print each volume on a separate system. This requires that each volume be processed independently. Refer to the EOV parameter in table 4-15 for a discussion of this processing.

In the event of a system rollover while processing a multivolume report, input (as a portion of the recovery process) displays the following message:

ENTER 'CON I' WHEN VOLUME n IS MOUNTED
before performing a forward tape space to the recovery point. 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.
The 4635 LPS provides two standard paper feeder trays (tray 1 and tray 2). Four additional trays (trays 3, 4, 5, and 6) are a mixture of feeder-stacker modules that increase output bin and feeder capacity. A bulk input feeder (tray 7) provides one more feeder. Multiple feeder-stacker modules provide flexibility by allowing additional stock loading and increased production due to less required operator intervention.

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

### Cluster features

Clusters are a systematic method provided for feeding specific paper stock to print jobs. The following are major cluster features:

- One or more feeder trays are identified as a single logical group called a “cluster.”
- Each cluster feeds one stock type to the LPS which automatically selects between the cluster trays as they become empty or as page size requirements change.
- Cluster can have stock in different sizes.
- One or more trays in the cluster may be “preferred.” Preferred trays are trays recommended by the application programmer or by the site supervisor for the operator to load the stock.
- Trays actually loaded with the stock are the “current” trays.
- Operator can override the preferred tray recommendation.
- Data stream can specify the cluster used to feed a page.
- Operator can override the data stream instructions by selecting a specific cluster or tray.

### Cluster processing overview

Briefly, the programmer creates applications which tell the system controller on which stock jobs are to be printed. The operator loads the stock and informs the system controller which stock is in which tray. The system controller then processes the job in two steps:

- Identifying the stocks requested by the application
DEFINING CLUSTERS

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

What clusters do for the programmer and operator

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

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

Then you store the defined clusters in the CLUSTR.LIB file through the cluster (CLP or CLU) commands described in your 4635 LPS Operations Reference. CLUSTR.LIB, the printer’s cluster database, stores up to 159 clusters. The clusters can be called by all jobs run on the system. Since jobs often require various stocks, you also define cluster groups as “stocksets.” The stocksets can also be called by all jobs run on the system.

The operator uses clusters to decrease how often you load stocks. The operator verifies job and cluster status and overrides tray specifications for clusters through the cluster (CLU) commands described in your 4635 LPS Operations Reference. The cluster definitions inform you which stocks to load in which trays for queued jobs.

When an operator loads trays without entering CLU commands, the system assumes that the trays have been refilled and no cluster changes have occurred.

Where clusters are stored

The CLUSTR.LIB file is the cluster database for the LPS. Each cluster definition is a record in the file. This file is created when a print job is run and no CLUSTR.LIB exists.

CLUSTR.LIB is a user file which you can copy from one system to another, even if the first LPS has trays which the second LPS does not have. (The trays not in the second system are deleted from the cluster definitions which reference them.)

The default CLUSTR.LIB created contains definitions for the MAIN, AUX, and AUTO clusters. These clusters can be used as they are or modified before use, but they cannot be deleted:

- MAIN is predefined as tray 1
- AUX is predefined as tray 2
- AUTO is predefined as trays 1, 2, 3, 4, 5, 6

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

- Do not, however, use FEED=MAIN and FEED=AUX in conjunction with stocksets.
- You must use the FEED=MAIN operator command at the start of a job if the data stream is controlling trays, cluster information, and so on.
- You cannot edit CLUSTR.LIB. It is automatically updated by the CLU and CLP commands which enter clusters in the database or modify cluster definitions.

How applications use clusters

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

- Some applications do not change stock types (simple applications).
- Some applications change stock types but do not use stockset commands to do so (OTEXT applications).
- Some change stock types using the STOCKSET command (stockset applications).

The appropriate cluster usage for these three types of applications is discussed below.

Simple and OTEXT applications

These applications do not provide for the use of different stocks. If the application is online, then it uses the default stock for the site. If the application is offline, and the operator knows from the START command which stock to load, the operator loads the stock. Simple applications are fed stock from the MAIN and AUX clusters. MAIN and AUX are defined by the programmer to improve printer performance for the jobs run by the site, ensuring that paper can be loaded during printing.

Note: If your site runs simple, OTEXT, and stockset jobs, then it is useful to maintain a cluster for each stock loaded using cluster and stockset definitions and changing MAIN and AUX to logicals for each cluster as required by the jobs.

The OTEXT parameter of the MESSAGE command is described in detail in chapter 6, “Specifying output parameters.”

Stockset applications

Each page has a particular stockset and a particular cluster associated with it. The cluster is always verified because it determines from which trays a page may feed. The stockset is verified only if the current page calls a different stockset than the previous page called.
Stockset changes require the following cluster checks:

- **Missing**: Is the cluster defined in the CLUSTER.LIB?
- **No trays**: Does the cluster have current trays?
- **Tray overlap**: Do any of the clusters have the same current trays?
- **Sizing error**: Are all trays the same size? If not, is autosize set? Refer to your Xerox 4635 LPS Operator Guide for more information.

A cluster without these errors has an “in use” status, that is, it has trays and is in the stockset last called by the job. Any one of these errors in any clusters in the stockset gives “in error” status to the stockset.

As jobs print, the output task uses the application’s specified cluster to choose a tray for each page. Occasionally it provides a cluster status display. If an operator enters FEED commands that override the data, for example, the FEED=DJDE, then the printer cycles down and displays:

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

### Mixing applications

When a site runs simple and OTEXT applications with stockset applications, the LPS executes two additional stockset checks to help ensure the correct paper is loaded. They are as follows:

- If a report has no stockset information, then a stockset change is assumed.
- A stockset change is assumed at each job boundary, regardless of the stocksets used.

To run a simple or OTEXT application correctly after a stockset application while avoiding printer cycle downs, ensure that the current trays are always in the MAIN and AUX clusters.

A stockset application run after a simple or OTEXT application always causes a printer cycle down, allowing you to load the correct trays (if necessary) before continuing.

### Defining clusters and stocksets with PDL and DJDE

The PDL commands and DJDEs listed here relate to clusters. For more detailed information on each command, refer to chapter 6, “Specifying output parameters,” and chapter 8, “Specifying dynamic job descriptor entries (DJDEs).”
**PDL commands**

The **STOCKSET** and **OUTPUT** commands specify cluster and stocks in your JSLs. Their syntax is shown in this section.

```plaintext
ac: STOCKSET
ASSIGN = {stock-descriptor
(stock-descriptor1[,stock-descriptor2][,...])

INIFEED = {stock-name
stock-reference

SYSPAGE = {stock-name
stock-reference
MAIN
AUX

OUTPUT
FEED = {stock-reference
stock-name
MAIN
AUX
OPR

STOCKS = stockset-name
```

**DJDEs**

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

```plaintext
FEED = {stock-reference
stock-name
MAIN
AUX
OPR

STOCKS = stockset-name
```

**Points to note**

- Assign each cluster a unique logical name which begins with a letter and consists of 1 to 6 alphanumeric characters. Make certain that the cluster name is not a keyword, parameter, or option used by the LPS. Group the clusters used by each application into a stockset.
- Stocksets convert cluster references to cluster names. (Cluster references are recommended but are not required.)
The cluster name appears first in quotes, followed by its cluster reference. The cluster is identified to the operator by cluster name. Try to define stocksets with a maximum of one cluster for each feeder tray on the LPS since the printer often cycles down when a report calls a stockset with more clusters than the number of feeder trays on the system. You may define stocksets within JDLs or catalog stocksets separately in an STK file. Like PDE files, STK files are accessible to all JDL files. If you define the stockset in an STK file, you must recompile only the STK file to effect a stock change. If you define the stockset in the JSL files which use it, you must recompile all the JSL files to effect a stock change.

- Avoid using stocksets with the cover and RAUX options. Cover pages and pages which meet the RAUX criteria are fed from the AUX cluster, overriding the FEED= option (or any other option) specified for these pages.

Steps for creating clusters

The steps you should take to create a cluster are listed below. Figure 5-1 provides a visual representation of these steps.

Figure 5-1  Steps for creating clusters

Follow these steps:
1. Determine the stocks the application will use.
2. Create a JSL file and compile it to produce an STK file: Define a logical mnemonic name (a cluster reference) for each stock (for instance, COVER in the example below). Associate each cluster reference with a cluster name easily.
recognize by the operator as a particular stock (for instance, CV102 in the example below).

```plaintext
BILLS: STOCKSET  ASSIGN=('CV102', COVER),
     ASSIGN=('BL231', BILPAG),
     ASSIGN=('SM233', SUMMRY);
```

3. Use the cluster reference rather than the cluster name in .JSL files and DJDEs. This way you can make cluster stock changes by modifying the cluster name in the STOCKSET command and recompiling it. This way, you avoid modifying the data stream or the application. The cluster reference in JDL files and PDE files thus remains valid.

4. Use the FEED= DJDE in the JSL file to specify the cluster to be used. For example:

```plaintext
...
...data for cover letter...
DJDE FEED=BILPAG,END;
...data for bill...
DJDE FEED=SUMMRY,END;
...data for summary sheet
(and in the JSL)
OUTPUT STOCKS=BILLS, FEED=COVER;
```

**Note:** The specified cluster must be identified in the active stockset.

If clusters are used in the print job, the OUTPUT command requires the STOCKS= parameter; the FEED= parameter is optional. The INIFEED= parameter of the STOCKSET command takes effect when no FEED = parameter is coded. If the STOCKSET command has no INIFEED= parameter, the first cluster specified in the STOCKSET command is used. The following example produces the same results as the previous example.

```plaintext
BILLS: STOCKSET  ASSIGN=('CV102', COVER),
     ASSIGN=('BL231', BILPAG),
     ASSIGN=('SM233', SUMMRY),
     INIFEED=COVER;
OUTPUT STOCKS=BILLS;
```

**Note:** The STOCKS=DJDE makes the STOCKSET command INIFEED= parameter take effect, thereby canceling the previous FEED = parameter.

The STOCKSET command has an optional SYSPAGE= parameter. It feeds the system pages (for example, PLABEL, OPRINFO, rollover marker pages, accounting pages) from the specified cluster; otherwise these pages are printed from the MAIN cluster. The following example illustrates its use:

```plaintext
BILLS: STOCKSET  ASSIGN=('CV102', COVER),
     ASSIGN=('BL231', BILPAG),
     ASSIGN=('SM233', SUMMRY),
     ASSIGN=('RED00', RED),
     INIFEED=COVER,
     SYSPAGE=RED;
OUTPUT STOCKS=BILLS;
```
Keeping stockset changes to a minimum

Each formatted page is associated with the active stockset and the active FEED= option stock identified by the INIFEED parameter of the STOCKSET command, the FEED parameter of the OUTPUT command, or the FEED DJDE. A stockset change causes the LPS output task to make additional checks when a page is to be printed. These checks can result in a printer cycle down if any cluster in the new stockset presents a tray overlap or sizing problem. Thus you should keep stockset changes within applications to a minimum. For example:

DJDE STOCKS=BILLS1, END;
...data...
DJDE STOCKS=BILLS1, END;
...data...
DJDE STOCKS=BILLS2, END;
...data...

This data stream has three STOCKSET= specifications, but only one stockset change. Report boundaries are not treated as stockset changes unless use of the stockset specified in a JDE or JDL makes a page print on another stockset. For example:

DJDE STOCKS=BILLS1, END;
...data...
REPORT BOUNDARY
...data...
DJDE STOCKS=BILLS1, END;
...data...
DJDE STOCKS=BILLS2, END;
...data...

The data following the report boundary is printed using the JDE or JDL of the START command. If it does not specify OUTPUT STOCKS=BILLS1, the stockset changes.

**Note:** If the stockset changes but the second stockset consists of the same clusters as the first stockset, no cycle down occurs since, for printing purposes, the stockset has not changed.

Using clusters with ordered stocks

Ordered stocks consist of any repeating sequence of paper stock types stacked in a tray. These clusters function in the same way other clusters function.
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.

### Overview

Table 6-1 summarizes these PDL commands. The function, syntax, and usage of the parameters available with these commands are described in individual sections of this chapter.

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNORMAL</td>
<td>Specifies operations security and page offset on jam recovery.</td>
</tr>
<tr>
<td>ACCT</td>
<td>Defines user and system accounting requirements.</td>
</tr>
<tr>
<td>BARCODE</td>
<td>Allows processing of barcode data streams.</td>
</tr>
<tr>
<td>CME</td>
<td>Allows certain parts of report output to be replaced on copies with predefined static data or to specify font changes within variable data.</td>
</tr>
<tr>
<td>EXPORT</td>
<td>Directs printing of segments and segment recovery.</td>
</tr>
<tr>
<td>LINE</td>
<td>Controls margin, overprinting, carriage control, and so forth.</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>Sends message to operator during input or output processing.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Controls printing mode, forms, paper size and type, offsetting and output destination.</td>
</tr>
<tr>
<td>PDE</td>
<td>Defines page format (page orientation, fonts).</td>
</tr>
<tr>
<td>ROUTE</td>
<td>Sends message and form for a routing page.</td>
</tr>
<tr>
<td>STOCKSET</td>
<td>Defines a set of stocks used in a report.</td>
</tr>
<tr>
<td>VFU</td>
<td>Defines vertical format control.</td>
</tr>
</tbody>
</table>
The ABNORMAL command allows you to restrict certain operator functions and define system responses to error conditions. Table 6-2 summarizes ABNORMAL command parameter functions.

### Table 6-2. Summary of ABNORMAL command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>System response to abnormal conditions detected while processing or compiling DJDEs</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>IMISMATCH</td>
<td>The action to take when a specified ink requires a primary that is not currently loaded.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SUBSTITUTE</td>
<td>Specifies whether operator-initiated ink substitution is allowed.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>OTEXT</td>
<td>Optional WAIT message following recovery.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PAGES</td>
<td>Specifies the maximum number of successive simplex pages on which recovery to the page boundary is to be attempted.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>REP</td>
<td>Page offset and/or slip-sheet after recovery.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SECURITY</td>
<td>Restricted use of certain operator commands.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

### ABNORMAL command parameters

The following provides the syntax and function of ABNORMAL command parameters and defines the parameter options.

### ERROR

**Syntax**

```
ABNORMAL ERROR=
```

**Function**

Specifies the required system response to abnormal conditions detected in input while processing or compiling DJDEs.

**Option(s)**

- **CONTINUE** - Displays the normal system message defining the abnormal condition and continues processing the report without applying the offending DJDE parameter and without offering the operator the parameter of continuing or aborting the job.
- **ABORT** - Displays the message for the abnormal condition and aborts only the report it is processing (not the entire job) without offering the operator the parameter of continuing or aborting the job.
- **STOP** - The default for this parameter and indicates that the system stop processing the report and wait for the operator to respond with a CONTINUE or ABORT command.
**Considerations**

The `CONTINUE` and `ABORT` parameters are referred to as non-stop modes. The `STOP` parameter is the stop mode. Any abnormal condition detected, such as syntax error in DJDE specifications, missing file referred to by a DJDE, RTEXT error, or GRAPHIC error in either mode results in an appropriate error message being displayed. In addition, during the stop mode, the displayed error message is followed by a DJDE error message:

```
02006 ENTER CONTINUE I OR ABORT I
```

If abnormal conditions are detected while processing or compiling DJDEs, the DJDE parameter set containing the offending entry is set to print on the OPRINFO page and an error message is displayed. Errors detected while processing RTEXT data are not reported to the user. The graphics processing errors are reported separately on the graphics exception page printed at the end of the report.

**Default**

Default is STOP

---

**IMISMATCH**

**Syntax**

```
ABNORMAL IMISMATCH = { STOP | ABORT | CONTINUE }
```

**Function**

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.

**Option(s)**

STOP

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

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:

OS6213 × 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 the 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.

Default
Default is STOP
ISUBSTITUTE

Syntax

ABNORMAL ISUBSTITUTE= \{ ANY \ |
\ |
NONE \}

Function

Specifies whether operator-initiated ink substitution is allowed.

ANY

Any operator-initiated substitution is allowed. This is the default.

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.

Default

Default is ANY

O TEXT

Syntax

ABNORMAL OTEXT= \{ WAIT \ |
\ |
NOWAIT \}

Function

Specifies that the system is to stop and display a WAIT message

when the recovery marker page is printed. This delay allows the

operator to verify the output report in the near vicinity of the

marker page. The default is NOWAIT, which means that the

message is still displayed upon printing of the marker page, but

printing does not stop.

Example:

ABNORMAL OTEXT=WAIT,
ERROR=ABORT;

Default

Default is NOWAIT

PAGES

Function

Specifies the maximum number of successive simplex pages on

which recovery to the page boundary is to be attempted. For
Syntax ABNORMAL PAGES = \{ n \}

duplex printing. \( n \) is the number of sheets or front-side, back-side pairs of pages to be attempted. The default is zero.

Default: Default is 0

**REP**

Syntax ABNORMAL REP = \{ NO | YES \}

Function: Specifies that the first data page delivered to the bin following a printer jam recovery or a system crash may be offset from the rest of the report.

- NO specifies that the page will not be offset.
- YES specifies that the page will be offset.
- AUX specifies that a blank sheet, picked from the auxiliary tray, is to be placed as a slip-sheet before any data pages printed after a recovery from a jam, crash, or potential output integrity problem.

If the YES, AUX option is specified, the AUX sheet is the sheet that is offset (and not the first data sheet).

**SECURITY**

Syntax ABNORMAL SECURITY = \{ YES | NO \}

Function: Specifies that certain operator functions are restricted. YES specifies the following system level parameters (as described in the 4635 LPS Operations Reference: SAMPLE (sample print), MOVE BLOCKS (block spacing of tape input), and SPACE PAGES (skipping over output pages). NO specifies that none of the above restrictions are in effect.

If SECURITY is invoked in a selected JDE, such as a DJDE JDE, the SAMPLE and SPACE PAGES restrictions do not take effect until after the first page of the report has been imaged (or spaced past).

Default: Default is NO

**Recovery**

The automatic recovery procedures of the LPS are based on the availability of valid checkpoint data independently saved by both the output processor and the input processor. As a result, most
recoveries are performed with no operator assistance. In certain cases, the system may detect that the checkpoint data is invalid, and operator assistance is requested, that is, the operator or user is requested to mount a preceding tape volume (offline) or return to point of origin or last valid checkpoint (online).

**Marker page**

The system produces a marker page (refer to figure 6-1) as a function of the input processor and inserts it into the print job at the point of crash. It is printed after the output processor has printed data that was already processed by input before the crash occurred.

**Note:** Actual size is 8.5 by 11 inches, and the squares at the top and bottom of the page are black.
Figure 6-1  Sample marker page

***********************************************************************
*RECOVERY MARKER PAGE RECOVERY MARKER*
*E  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*
*  *
*M  JOB NUMBER:  2565  M*
*A  A*
*R  REPORT NUMBER:  1  R*
*K  K*
*E  E*
*RECOVERY MARKER PAGE RECOVERY MARKER*
***********************************************************************
ACCT command

The system maintains report processing information in two files which are accessible to the user. Both files can be saved on tape or transmitted to the host through the host file transfer facility for further processing. One method accumulates overall usage statistics on a department or JDL basis. Refer to your Xerox 4635 LPS Operations Reference for further information on the accounting file printout, initialization, and structure. In the second method, the status file accumulates report information, in the order that it is processed, on an individual report basis. Refer to the system level parameters JOBS and SFS and the “Report status information” section in your Xerox 4635 LPS Operations Reference for further information on the status file display, initialization, and structure.

Accounting information is automatically accumulated by the system on a report basis and also for overall system usage. The ACCT command enables the user to include a printout of the accounting summary with each report printed. Table 6-3 summarizes ACCT command parameter usage. The system level REPORT USER parameter provides an installation with a printout of overall system usage statistics. The following sections define how accounting information may be accessed and interpreted.

Table 6-3. Summary of ACCT command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT</td>
<td>Accumulation of user accounting statistics</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>USER</td>
<td>Output destination of accounting page with a report</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

An example of an offline accounting page is illustrated in figure 6-2.

Accounting data on report basis

The ACCT command enables you to request that an accounting page which contains a summary be output at the end of the printing for each report. This summary consists of a single page of information containing job setup information and counts of processing events.

Overall usage accounting

The system automatically accumulates accounting data in a file on disk for each report printed. An installation has the parameter of having the system accumulate this information under a specified name, such as department code, account, or accumulate it under a system default name.

The DEPT parameter of the ACCT command allows an installation to define a name under which the accounting data is accumulated. The system ACCOUNT command (refer to the Xerox 4635 LPS Operations Reference) must also be used with the DEPT parameter to create a name in the accounting file.
which corresponds to the name specified in the DEPT specification. It is necessary to enter this name only once with the ACCOUNT parameter, but it must be done before a print job is run using the DEPT name. If the name is not entered with the ACCOUNT parameter, the accounting data for this report is accumulated in the NODEPT entry in the accounting log. However, the DEPARTMENT entry on the accounting page for this report lists the requested department name followed by an asterisk to denote that no specific accounting by this name was performed.

If the DEPT parameter is not specified, data is accumulated under a name made up of the identifier on the JDL SYSTEM (or JDL) parameter with the characters :JDL appended to it, assuming that this same name exists in the accounting file. Thus, if a JDL identifier is H2SYS, and no DEPT was specified, the default accounting name on the accounting page is H2SYS:JDL. Data is accumulated under this default name only if this name exists in the accounting file. If the name is not in the accounting file, the accounting information goes to NODEPT. ACCOUNT parameter use is described in the “Accounting file maintenance” section of your Xerox 4635 LPS Operations Reference.

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

When the system level parameter ACCOUNT ADD, OPERATIONS is entered at the keyboard display, the name OPERATIONS is added to the current list of names in the system accounting file. The PDL 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 entered under this name or after the accounting file has been cleared.

### Installation accounting report

The REPORT USER system level parameter is used to print a copy of system accumulated accounting statistics. It prints out accounting data for each name entered into the accounting file by the ACCOUNT parameter. Use of the REPORT USER parameter is discussed in the chapter on tracking system activity in the Xerox 4635 LPS Operations Reference. A sample of the output is shown in figure 6-2.

DEPARTMENT/JDL entries as shown in figure 6-3 are generated by the OSS (NODEPT and TASKS$:JDL) and entered by you with the ACCOUNT parameter. NODEPT is the name under which accounting statistics are accumulated for print jobs without name entries in the accounting file. TASKS$:JDL is a name under which accounting is accumulated for printing done by system tasks such
as FDL, PDL, and the Editor. OPERATIONS is a user-supplied DEPARTMENT/JDL entry (in figure 6-3).

Other entries on this report, such as PAGES and LINES, are defined the same as those on the end-of-report accounting page.
Figure 6-2  Sample accounting page printed with job

```
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX

DATE:         14 OCT 94   AT   19:35:32
DEPARMENT:    M0036:JDL*
JOB ID:       248         REPORT NO. 1
FILE ID:
Input Processing Time:    00:00:07
Output Processing Time:   00:00:13
REPORT COMPLETION CODES:  0        0
Pages to Bin:             1
Pages to Tray:            1
Paper Path Holes:         0
Lines Printed:            66
Online Idle:              0
Blocks Read:              0
Blocks Skipped:           0
RECORDS READ:            79
DJDE RECORDS READ:        1
Maximum Copy Count:       1
OVERPRINTS:              0
COLLATE:                 YES
SF/MF:                   MULTI
SIMPLEX/DUPLEX:          SIMPLEX
JDE,JDL USED:            1,M0036
ACCTINFO:
INITIAL FONT LIST:        L00TPB

INITIAL FORM LIST:        HCF129

INITIAL CME LIST:         -NONE

```

XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX LPS XEROX
```
### Figure 6-3  
**Customer usage accounting report**

<table>
<thead>
<tr>
<th>Date</th>
<th>User Accounting Report</th>
<th>Page 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Department/JDL</strong></td>
<td><strong>Printing Summary</strong></td>
<td><strong>Time Summary</strong> -HR, MIN-</td>
</tr>
<tr>
<td>NODEPT</td>
<td>PAGES = 21</td>
<td>INPUT = 0,1</td>
</tr>
<tr>
<td></td>
<td>LINES = 2200</td>
<td>OUTPUT = 0,2</td>
</tr>
<tr>
<td></td>
<td>PAGES TO TRAY = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Processing Summary</strong></td>
<td><strong>Tape Summary</strong></td>
</tr>
<tr>
<td></td>
<td>JOBS = 2</td>
<td>MOUNTS = 0</td>
</tr>
<tr>
<td></td>
<td>FILES = 2</td>
<td>BLOCKS READ = 220</td>
</tr>
<tr>
<td></td>
<td>REPORTS = 2</td>
<td>BLOCKS SKIPPED = 0</td>
</tr>
<tr>
<td>Operations</td>
<td>PAGES = 105</td>
<td>INPUT = 0,5</td>
</tr>
<tr>
<td></td>
<td>LINES = 1700</td>
<td>OUTPUT = 0,7</td>
</tr>
<tr>
<td></td>
<td>PAGES TO TRAY = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Processing Summary</strong></td>
<td><strong>Tape Summary</strong></td>
</tr>
<tr>
<td></td>
<td>JOBS = 1</td>
<td>MOUNTS = 1</td>
</tr>
<tr>
<td></td>
<td>FILES = 7</td>
<td>BLOCKS READ = 1100</td>
</tr>
<tr>
<td></td>
<td>REPORTS = 7</td>
<td>BLOCKS SKIPPED = 0</td>
</tr>
<tr>
<td>Tasks:JDL</td>
<td>PAGES = 3</td>
<td>INPUT = 0,0</td>
</tr>
<tr>
<td></td>
<td>LINES = 100</td>
<td>OUTPUT = 0,0</td>
</tr>
<tr>
<td></td>
<td>PAGES TO TRAY = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Processing Summary</strong></td>
<td><strong>Tape Summary</strong></td>
</tr>
<tr>
<td></td>
<td>JOBS = 3</td>
<td>MOUNTS = 0</td>
</tr>
<tr>
<td></td>
<td>FILES = 3</td>
<td>BLOCKS READ = 0</td>
</tr>
<tr>
<td></td>
<td>REPORTS = 3</td>
<td>BLOCKS SKIPPED = 0</td>
</tr>
</tbody>
</table>
ACCT command parameters

The following provides ACCT command parameter syntax and defines the parameter options.

DEPT

Syntax

ACCT DEPT= \( sc \)

Function
Defines the name under which accounting statistics should be accumulated for this report job.

Option(s)

\( sc \) - A string constant of up to 31 characters representing a department code or name under which accounting information is maintained. These 31 characters and including the character ‘:’ are allowed for the department code. Refer to the “Overall usage accounting” section for further information on DEPT parameter usage. The \( sc \) name should already be created in the accounting file prior to processing the job.

Example: ACCT USER=BIN, DEPT='MARKETING';

Default
Default is JDL name

USER

Syntax

ACCT USER= \{ BIN, TRAY, BOTH, NONE \}

Function
Specifies destination of the one-page accounting summary for each report. This page is always printed in the mode of the report. Available parameters are: BIN, TRAY, BOTH, and NONE.

BIN
Directs an accounting page to the output bin.

TRAY
Directs an accounting page to the sample tray.

BOTH
Directs an accounting page to the output bin and the sample tray.

NONE
Specifies that no accounting page is delivered.

Note: When running in duplex, line counts for the back sides of the last five (or fewer) pages of each report are not reflected in the system accumulated accounting statistics. SELECT E specifies that all accounting pages are forced to tray.
Default

Default is BIN

Points to note—ACCT command

- If the report completion code on the accounting page is non-zero, an accounting page is delivered to the tray even if USER=NONE is coded. This is done to notify the user of a possible report integrity problem.

However, if the report is duplex and the first integrity problem occurs within the last eight back sides, the report completion code on the forced accounting page is zero.

- Jobs aborted by input as a result of job recovery produce an accounting page with as much accounting information as can be salvaged. The accounting page is delivered to both the bin and the sample tray.

- When ACCT USER=BOTH is specified, the two accounting pages show different values for “Output Processing Time.” The sheet sent to the stacker will show a time about one second longer than the sheet sent to the sample tray. All other values should match.

- Hexadecimal strings in a DEPT parameter are interpreted as standard EBCDIC and translated to Extended ASCII.

- A completion code of 0 indicates that the report may have completed successfully, except when a report integrity problem occurs within the last eight duplex pages.

Example—ACCT command usage

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

After each report is printed, an accounting page is delivered to the output bin and sample print tray. Accounting information for each job (run under this ACCT command) is accumulated in the system accounting log under the department name BUDGET. This assumes the name BUDGET has been entered into the list of accounting names (through the system level ACCOUNT parameter).

BARCODE

Syntax

BARCODE { [BSKIP=READ|SKIP] 
[BSIDE=ODD|EVEN|BOTH] 
[BSEQ=RESET|NO RESET|NO SEQ] }

Function

The BARCODE command allows processing of bar code datastreams with the ability to cross-check the three components of the system, namely, the PC UI, the system controller (ESS), and the printer (IOT). It is in cross-checking that the 4635 LPS is able to ensure that any bar code sheets printed have been accounted for by each component of the software. This handshake activity is synchronized using presence and sequence number information.
PRINT FORMAT COMMANDS

READ Indicates that all presented sides of all sheets will be scanned for bar code information, regardless of the presence information for that sheet. Bar codes encountered when none are expected generate an error condition.

SKIP Denotes that bar code information is processed only for pages with presence information indicating that bar codes are expected. In other words, bar codes encountered when none are expected are ignored. This is the default.

ODD Indicates that the system will assign sequence and presence information assuming that only odd sides are to be considered. This is the default.

EVEN Indicates that only even sides will be considered when assigning sequence and presence information. If this is coded for a simplex job, the job will contain no bar code sequence or presence information.

BOTH Indicates that odd sides if simplex sheets and both sides of duplex sheets will be considered in sequence and presence processing.

RESET Indicates that the sequence number should be reset to a value of one at the beginning of each report. This is the default.

NORESET Indicates that the sequence number will be continued across report boundaries.

NOSEQ Indicates that for pages matching the odd-even-both specification, the presence bit will be set, and a sequence number of zero will be specified. This indicates that the IOT should simply report the value of any bar codes encountered on these pages. No error is generated when no bar code is read on a page where one is expected.

Points to note—BARCODE command

- Sequence numbers can range from 0 to 65535, but 0 and 65535 are reserved for system use (refer to the BAR ALL command in the 4635 LPS Operator Guide and BARCODE NOSEQ commands in this chapter).
- At the start of a job, the sequence number is always reset to 1. It can also be reset when a user-specified ABORT I or an INPUT processing forced abort was initiated.
- The READ parameter is not intended to physically enable (or disable) the barcode reader. It only directs the course of action taken when a barcode is read when none is expected.
- JSLs using the BARCODE statement must be recompiled.

Use BSEQ to either reset the sequence number at the start of a report (RESET), not reset the sequence number (NORESET), or recognize the presence of bar codes without specifying sequence numbers (NOSEQ). The 4635 LPS makes the determination to perform RESET/NORESET/NOSEQ processing at the start of every report. The start of report for BSEQ
processing includes any DJDE packets specified prior to any data in the report. Any BSEQ DJDEs specified thereafter are ignored, including any JDE/JDL DJDEs.

### ac:CME command

The copy modification feature (also referred to as spot-carbon) allows certain parts of report output to be replaced on selected copies with predefined static data or to specify font changes within the variable data.

CMEs define a rectangular space on the printed page within which printed data is replaced with a substitution string or in which a font change occurs. More than one CME may be applied to a job. CMES may be coded as part of the JDL or created as separate files so that they may be referenced by one or more JDLs. This latter feature is described more fully in the “Cataloged CMEs” section of this chapter.

An identifier (of the type ac) is specified when defining the CME and references it with the MODIFY parameter of the OUTPUT command. The CME must precede the OUTPUT command.

CME parameters are summarized in table 6-4.

**Table 6-4. Summary of ac:CME parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTANT</td>
<td>Copy modification character string to be printed</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>FONT</td>
<td>Selection of font for printing input or CME data</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>LINE</td>
<td>Line range of copy modification</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>POSITION</td>
<td>Initial character position of copy modification</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

### Short form CME specifications

CME specifications may be given in short form to minimize the coding length. In short form, only the first character of a keyword need be given, equal signs are omitted, and commas are not inserted except where necessary to avoid ambiguity.

### Cataloged CMEs

CMEs need not be part of a user's JDL. They may be created as separate disk files and used as if they were part of the JDL that references them. This is done by creating a JSL file containing only CMES and using the PDL processor to compile it. For each CME, PDL creates an object file on disk, cataloged in the CME directory. The CME identifier is used as the object file name. When a CME source file is modified, it must be recompiled with PDL, but it is not necessary to recompile the associated JDLs.

When the CME is referenced by the OUTPUT command MODIFY parameter in a JDL and the CME does not exist within the JDL,
the system searches the directory for the file type xxxx.CME. It is then loaded into memory for use in processing the report.

The DJDE MODIFY parameter can also be used dynamically to associate a cataloged CME file with report processing. CMEs must be cataloged as separate disk files if they are referenced by a DJDE MODIFY command.

CME parameters

The following provides CME parameter syntax and defines the parameter options.

CONSTANT

**Syntax**

```
ac:CONSTANT= sc
```

**Option(s)**

- `sc` - Represents the string constant to be printed (refer to chapter 2 for definition of string constants). The width of the copy modification rectangle is determined by the number of characters specified by `sc`. More than one `sc` is allowed.

**Considerations**

There is no default for the CONSTANT parameter of the CME. It must be specified unless all that is required is a font change in the variable data. Within a text string, the character `#` may be used as a lowercase toggle. When a text string is encountered, it is assumed that characters are to be inserted into the print line as they appear in the text string (in uppercase normally). If a `#` is encountered, the lowercase mode is invoked, and all letters after `#` are considered lowercase until another `#` is encountered. The sequence `##` is used to indicate that the character `#` is to be inserted and is not to be treated as toggling lowercase mode.

**Default**

No default

FONTS

**Syntax**

```
CME FONTS= value
```

**Function**

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.

**Option(s)**

- `value` - May 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 a value is the current font (specified with the last FONTS 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 (POSITION) but no insertion text (CONSTANT) are specified, the
font change specified applies to input variable data at the position specified.

Default: No default

**LINE**

**Syntax**

CME LINE= \( \begin{pmatrix} n \\ (n,m) \\ (n,-) \end{pmatrix} \)

**Function**

Specifies the line range of the CME.

**Option(s)**

- \( n \) - The initial line number of the copy modification rectangle.
- \( m \) - The number of lines to repeat the information. If not specified, the information applies only to the starting line.
- A dash character indicates that information is to apply to all lines on a page beginning with the line indicated by \( n \).

**Considerations**

The line numbers specified in the LINE parameter must always be in ascending order. In other words, you cannot specify one line number then list another one of lower value in the same command. For example, the following command is incorrect and will not be processed:

CME1: CME LINE=(2,3),POS=15, CONSTANT=(6) 'HELLO',FONTINDEX=3;
LINE=1,POS=15;

As you can see, the first line parameter specifies starting at line 2, while the second LINE parameter specifies line 1, which contradicts the ascending order.

Default: No default

**POSITION**

**Syntax**

CME POSITION= \( n \)

**Function**

Specifies the initial character position of the CME in the print line.

**Option(s)**

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

Default: Default is 1
Points to note—CME command

- The CME LINE and POSITION parameters are order dependent in that LINE must always precede POSITION.

- Data lines
  Are not properly aligned if proportional fonts are being used in landscape mode and a switch is made to a larger font in the middle of the line. This restriction is a result of the functioning of the printer imaging hardware.

- Multiple copies
  If a multicopy is aborted before all the pages are set up in the page log, one copy gets printed if the CMEs are copy-sensitive. Otherwise, all copies of the setup pages are printed.

- OVERPRINT, FON TINDEX, and CMEs
  - CMEs are not applied if OVERPRINT=MERGE and FONTINDEX are both specified.
  - CME parameters are applied to all lines within the range regardless of whether or not variable data appears on those lines.

- Multiple lines and columns
  Multiple lines may be specified and multiple columns may be specified for each line. Multiple line specifications must be given in ascending (top to bottom of page) order. Multiple column specifications for a line range must be given in ascending (left to right) order. There also may be multiple text specifications following a column specification. These are combined to form a single text string. Font specifications may be specified at any point. The last font specified remains in effect until another font is specified. An example of a CME with these characteristics is as follows:

  XYZ: CME LINE=(1,10),POS=40,FONT=2,POS=80,FONT=3,
   LINE=(11,20),POS=1,FONT=3,POS=40,FONT=1,POS=80,FONT=2,
   LINE=(31,50),POS=1,FONT=2,POS=40,FONT=3,POS=80,FONT=1;

- Form and variable data
  Under certain circumstances, a form and its associated variable data do not align properly, even though it appears that both use the same line spacing. Variable data has a line-spacing value that is computed as dots per line, and rounding is done on that value. In FDL, the rounding takes place only when it is needed to resolve to a dot address and, therefore, might involve more than one line at a time. Refer to the “Rounding measurements” section in the 4635 LPS Forms Creation Guide for further information.

- CME string constant
  Whether a font switch is applied to a CME string constant depends upon the order in which FONT and CONSTANT are specified in the CME. If FONT is specified before CONSTANT, the font switch applies to the string constant. If FONT is specified after CONSTANT, the font switch occurs at the line position immediately after the string constant.
PRINT FORMAT COMMANDS

WHERE A CME STRING CONSTANT IS PRINTED ON A LINE DEPENDS UPON WHERE THE POSITION AND CONSTANT PARAMETERS ARE SPECIFIED IN THE CME. IF POSITION IS SPECIFIED BEFORE CONSTANT, THE STRING CONSTANT IS PRINTED AT THE SPECIFIED POSITION. IF POSITION IS SPECIFIED AFTER CONSTANT, THE STRING CONSTANT IS PRINTED AT THE CURRENT POSITION (POSITION 1 IF NO OTHER POSITION OR CONSTANT PARAMETER HAS BEEN SPECIFIED), AND A SUBSEQUENT CONSTANT OR FONT PARAMETER WILL TAKE EFFECT AT THE SPECIFIED POSITION.

• RSX11 host type

When the host type is RSX11, CMEs are often applied twice per line because of the way the carriage return/line feed is processed. When mixing font switches and string constants, the string constants may appear twice and be overprinted with two different fonts. The recommended approach for RSX11 host types is to specify the string constant font before the string constant itself, such as:

```
FONT=1, CONSTANT='string', FONT=2;
```

Examples—CME command usage

**Example 1**

The following is an example of copy-sensitive CMEs.

```
CME1: CME LINE=3, POS=59, CONSTANT='FIRST QUARTER';
CME2: CME LINE=(37,3), POS=81, CONSTANT=('**');
CME3: CME LINE=(1,60), POS=5, FONT=2, POS=12, FONT=1;
```

```
OUTPUT COPIES=3, MODIFY=(CME1,1,1), MODIFY=(CME2,2,1),
MODIFY=(CME3,3,1);
```

For the first copy of the report, line 3 is modified by the text 'FIRST QUARTER'. For the second copy, the character '**' is inserted in lines 37, 38, and 39 for 6 character positions (starting in column 81). For copy 3, fonts are changed for lines 1 to 60 at character positions 5 and 12. These fonts must be defined in the FONTS parameter of the PDE command.

Copy-sensitive CMEs are not supported for online processing or when COLLATE=NO is specified in the OUTPUT command.

**Example 2**

**CME parameters specified in standard form:**

```
CME12: CME LINE=47, POSITION=1, FONT=5, CONSTANT=(5)'**',
       LINE=48, POSITION=1, FONT=1;
       LINE=49, POSITION=10, CONSTANT='ABCD' I2;
```

**CME parameters specified in short form:**

```
CME12: CME L47,P1,F5,(5)'**',L48,P1,F1,L49,P10,C'ABCD';
       or
       CME12: CME L47P1F5(5)'**'L48P1F1L49P10'ABCD';
```

EXPO RT command

A "segment" is a set of consecutively delivered sheets whose management as a set or segment has been specified by PDL or DJDE command.
The EXPORT statement enables specification in PDL of the new Segment Management capabilities, including the following.

- Division of reports into segments
- Generation of separator sheets
- Use of segment numbers on separator sheets
- Use of segment recovery in place of page recovery.

Although the EXPORT command is primarily intended to be used with the bypass transport option, it functions with any allowable destination. If the OUTPUT command DESTINATION parameter specifies BIN, segments are offset relative to each other when delivered, and offsetting resulting from other commands is not processed. Table 6-5 summarizes the use of the EXPORT parameters.

**EXPORT command parameters**

The following describes the syntax of the EXPORT command parameters and defines the parameter function.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEPARATORS</td>
<td>Retain an internal copy of the first data page and use it to generate segment separator sheets.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SNUMBER</td>
<td>Print the segment number on generated segment separator sheets.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SPLIT</td>
<td>The conditions for ending segments and turning segment management on.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SRECOVER</td>
<td>Recovery mode.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>STIMING</td>
<td>This command is listed for compatibility only. It is not supported on the 4635 LPS.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

**SEPARATORS**

**Syntax**

```
EXPORT SEPARATORS= {FIRST|LAST|(LAST,D)|BOTH|(BOTH,D)|NONE}
```

**Function**

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

**Option(s)**

- **FIRST** - Specifies that a printer generated separator sheet be the first sheet of the second and subsequent segments.

  **Note:** The first segment begins with the first data page, so an additional sheet is not generated.
LAST - Specifies that at least one copy of the saved data page should be generated (on a separate sheet of paper) as a hardcopy segment separator and counted as the final sheet in the segment.

Note: In an online environment, a true banner trailer page, if encountered, will be used to image any LAST separators of the final segment in place of the saved data page.

BOTH - Specifies both FIRST and LAST, as above.

NONE - Specifies that no segment separator sheets be generated.

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

Example: SEPARATORS=(BOTH,D);

Default Default is NONE

SNUMBER

Syntax \texttt{EXPORT\ SNUMBER=} \texttt{(lnum, cnum[,findex])}

Function Specifies that the printer merge the current segment sequence number (beginning with 001) into any separator sheets produced. The format of the printed segment sequence number for all separators except a LAST separator in the last segment is:

\texttt{PART} \texttt{nnn} (8 characters). \texttt{nnn} is a three digit number with leading zeros.

The format for the LAST separator in the last segment is: \texttt{PART} \texttt{nnn of nnn} (15 characters).

Option(s) \texttt{lnum} - Specifies the integer line number on which the segment sequence number string is to start.

\texttt{cnum} - Specifies the integer column number on which the segment sequence number string is to end.

\texttt{findex} - Specifies the font index of the font to be used for printing the segment sequence number string. If not specified, the first font in the page’s font list is used.

Example: \texttt{SNUMBER=(1,75)};

Considerations Note that \texttt{lnum} and \texttt{cnum} are interpreted using the metrics of the font specified by \texttt{findex} (the same as with page numbering).

Refer to the “Points to note—EXPORT command” section in this chapter for additional information.

Default No default
SPLIT

Syntax

\[ \text{EXPORT SPLIT=} \ (\text{min},\text{max}) \]

Function

Specifies the use of segment management, and is required in order for the other EXPORT parameters to work.

Option(s)

\((\text{min},\text{max})\) - Specifies the minimum and maximum numbers of sheets, including any separator sheets, allowed in a segment. The minimum available number is 1 and the maximum number is 65,535.

If a segment end has not occurred by the time max sheets are generated, then the printer declares end-of-segment. If end-of-segment occurs before min sheets are generated, then the printer generates padding of either blank or multiple LAST sheets (if LAST separators have been specified).

Example: \( \text{SPLIT}=(2,7); \)

Refer to the “Points to note—EXPORT command” section in this chapter for additional information.

Default

No default

SRECOVER

Syntax

\[ \text{EXPORT SRECOVER=} \ \{ \text{PAGE, SEGMENT, ASK, DEVICE}\} \]

Function

Specifies the mode of jam, or error, recovery.

Option(s)

\text{PAGE} - Specifies standard recovery. The printer resumes with the first sheet not successfully delivered.

\text{SEGMENT} - Specifies reprinting of the entire currently printing segment. Since it is possible for sheets of multiple segments to be in the paper path at the same time, the segment most advanced in the paper path without being successfully delivered is the starting recovery position.

\text{ASK} - Specifies that the operator be requested through the user interface to select PAGE or SEGMENT recovery.

\text{DEVICE} - Places recovery mode selection responsibility with a Device Finishing Architecture (DFA)-compliant finishing device through the LPS bypass transport option.

If the DEVICE recovery option is specified and output is directed to a destination other than the bypass transport, the PAGE recovery option prevails.

Example: \( \text{SRECOVER=}\text{SEGMENT}; \)
STIMING

Syntax  \[ \text{EXPORT STIMING} = ([\text{INTERVAL}, \text{sec}][, \text{DELAY}, \text{sec}]) \]

Function The STIMING parameter is provided for possible compatibility with other Xerox software supporting third party finishing devices. This parameter is ignored by the Xerox 4635 LPS.

Points to note—EXPO RT command

- **SNUMBER**
  If you specify a line and column value for SNUMBER that is off the page, the segment number will not print. Furthermore, if the system encounters a valid SNUMBER dynamic job descriptor entry (DJDE) under this condition, unpredictable results may occur.

- **SPLIT = OFF**
  Segment management processing is turned off when the LPS encounters a SPLIT=OFF dynamic job descriptor entry (DJDE). In addition, all segment management parameters are affected in the following ways:
  - SRECOVER defaults to PAGE recovery
  - SEPARATORS are forced to NONE
  - Segment numbering restarts to PART 001 if segment management processing is turned back on.
  
  You must use the DJDE SPLIT = (min,max) to reinstate segment management processing after it has been disabled by the DJDE SPLIT=OFF.

- **SPLIT = NOW**
  If segment processing is not in effect (that is, it either was not specified in the JDL or the DJDE SPLIT=OFF was invoked), the system ignores the DJDE SPLIT=NOW.

- **Ending a segment**
  A segment may be ended under any of these conditions:
  - The maximum allowable value is reached. This value is taken from the specified value in the SPLIT command.
  - The end-of-report condition is met, including the logical end-of-report as specified in the RSTACK command
  - An end-of-data condition is met
  - SEPARATORS is specified in a DJDE
  - The system encounters a SPLIT=NOW dynamic job descriptor entry
The system encounters a SPLIT=OFF dynamic job descriptor entry.

- **Using a new separator page**
  You may use a new separator page under these conditions:
  - A SEPARATORS dynamic job descriptor entry (DJDE) is specified.
  - An end-of-report condition is met, including the logical end-of-report as specified in the RSTACK command.

- **Resetting the segment number**
  You may set a new segment number after the current one is terminated. This is accomplished when the system encounters a series of two DJDEs: SPLIT=OFF followed by SPLIT=(min,max).

In an online environment, banner pages are routed to the current job output destination. For example, if a job is being sent to a third party finishing device, the banner page will be sent there, also. The system does not redirect banner pages to the sample tray or output bin unless you specify that they be routed to a tray other than the job destination. Refer to the “BANNER command” section of chapter 7, “Using logical processing commands,” for more information on specifying banner pages.

**Examples— EXPO RT command usage**

The following are sample EXPO RT commands:

- EXP RT SPLIT=(2,7), SNUMBER=(1,75), SEPARATORS=NONE;
- EXP RT SPLIT=(2,7), SNUMBER=(1,75), SEPARATORS=FIRST, SRECOVER=SEGMENT;

**LINE command**

The LINE command provides parameters which allow the user to define the characteristics of the user portion of the input data record and how it is to be printed. These parameters are summarized in table 6-6.
Table 6-6. **Summary of LINE command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Input record printable data portion</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FCB</td>
<td>Host transmitted FCB handling</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>FONTINDEX</td>
<td>Position of font switch index</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MARGIN</td>
<td>Left margin on a physical page</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>OVERPRINT</td>
<td>Manner in which overprint lines are handled</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PCC</td>
<td>Location of the printer carriage control field</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>PCCTYPE</td>
<td>Printer carriage control table</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>UCSB</td>
<td>Host transmitted UCSB handling</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>VFU</td>
<td>Vertical format table to be used</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

**LINE command parameters**

The following provides LINE command parameter syntax and defines the parameter options.

**DATA**

**Syntax**

\[ \text{LINE} \text{ DATA} = (\text{pdo}, \text{length}) \]

**Function**

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

**Option(s)**

- **pdo** - (Print data offset) is the number of bytes between the start of the user portion of the logical record and the first character of the record to be printed.
- **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.

**Default**

- Default is (1,132) offline
- Default is (0,150) online

**FCB**

**Syntax**

\[ \text{LINE} \text{ FCB} = \{ \text{IGNORE, PROCESS} \} \]

**Option(s)**

- **IGNORE** - Suppresses the processing of the host transmitted forms control buffer (FCB).
**PRINT FORMAT COMMANDS**

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

**Default**  Default is PROCESS

### FONTINDEX

**Syntax**

```
LINE FONTINDEX =
{ offset
  (offset[,init-val][,bit-opt])
  NONE
}
```

**Function**  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.

**Option(s)**

- **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 parameter must also be used. ZERO specifies that an index value of 0 is associated with the first font in the font list, an index value of 1 is associated with the second font in the font list, and so forth. The default is ONE.
- **bit-opt** - A numeral having a value in the range of 1 through 7, which specifies the number of low-order bits within the font index byte, which, in turn, specifies an index value into the font list of the current PDE. The default value is 4.

If init-val is ONE or is not specified, the value of the font index byte in the data record is a number in the range 1 to n, where n is the number of fonts specified in the PDE parameter (1 to 128). If ZERO is specified, the font index is a number in the range 0 to n, where n is the number of fonts specified in the PDE parameter minus one (0 to 127). The font index value must be present in every record. If the FONTINDEX in the data record is greater than the number of fonts specified, the first font in the list is used.

**NONE** - Specifies that there is no font index.

Refer to “Points to note” following this table for discussion of FONTINDEX usage with overprint lines and maximum number of fonts, and the examples which follow.

**Default**  Default is NONE

### MARGIN

**Syntax**

```
LINE MARGIN =
{ value
  (value,value-type)
}
```

**Default**  Default is NONE
**PRINT FORMAT COMMANDS**

**PRINT FORMAT COMMANDS**

**Function**
Specifies the left margin on a physical page.

**Option(s)**
- **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.
- **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.

Refer to the “Points to note—VOLUME command” in chapter 4 for additional information.

**Default**
Default is (1,POS)

**OVERPRINT**

**Syntax**
```
LINE OVERPRINT= (over-opt, disp)
```

**Function**
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.)

**Option(s)**
- **over-opt** - Options are: PRINT, IGNORE, MERGE, PRINT2. If IGNORE is coded, all overprint lines are ignored. PRINT specifies that all overprint lines are to be printed as they would be on an impact printer; that is, the second line is printed over the top of the first with no regard to the previous data, including character spacing, which may vary between the two lines of data. PRINT2 specifies that up to two consecutive lines are printed per line—one line and one overprint. Other overprints for the line are ignored. MERGE is the same as PRINT except when used with FONTINDEX or CME processing (refer to the “Points to note” section and example 2 of the “Examples—LINE command usage” section following this table).
- **disp** - Options are: DISP and NODISP. These are Xerox 1200 Computer Printing System parameters, which are preserved here for compatibility purposes. Neither is functional. The number of overprint lines is always printed on the accounting page.

Example:
```
LINE OVERPRINT=(MERGE,NODISP),VFU=V1,
DATA=(1,132),
PCCTYPE=ANSI,PCC=(0,NOTRAN);
```

**Default**
Default is (PRINT,NODISP)

**PCC**

**Syntax**
```
LINE PCC= (offset, trans-type)
```
**PRINT FORMAT COMMANDS**

**PRINT FORMAT COMMANDS**

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

**Option(s)**
- **offset** - Specifies the byte offset of the user portion of the record within each logical record to the printer carriage control (PCC) field. 0 is the default.
- **trans-type** - Options NOTRAN/TRAN specify whether or not the printer carriage control byte is to undergo code translation. TRAN indicates that the byte is to be translated into LPS standard EBCDIC before being applied, using the translation defined in the CODE parameter of the VOLUME command. NOTRAN (the default) prohibits translation.

**Default**
Default is (0, NOTRAN)

**PCCTYPE**

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

**Function**
Specification of a carriage control set used in printing a job.

**Option(s)**
- **pcc-name** - Options are: ANSI, B2500, B2700, B3500, B3700, B4700, B6700, H6000, H2000, IBM 4245. Creation of a user-defined PCC table referenced by either an identifier id or the keyword USER is defined using the PCC command. Refer to the “PCC command” section in this chapter. For offline systems the default is ANSI. For online systems the default is IBM3211. Note that there is no functional difference between IBM3211 and IBM4245 for PCC types. For HOST=ICL2900 the PCCTYPE is forced to NONE.

The INITIAL parameter (refer to the “PCC command” section in this chapter) for any selected PCCTYPE (except ANSI or HOST=RSX11, USER, and id) is TO F. 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 PCC parameter in this chapter) for any selected PCCTYPE (except IBM1403, IBM3211, USER, and id) is YES. For IBM1403 and IBM3211/IBM4245, ADVTAPE is set to NO. For USER or id the ADVTAPE is set by the user in the PCC parameter. Refer to the LPS Tape Formats Manual for more information on PCCTYPE tables.

**Default**
Default is ANSI offline
Default is IBM3211/IBM4245 online

**UCSB**

**Syntax**
```
LINE UCSB=( IGNORE PROCESS
```

6-30 XEROX 4635 LASER PRINTING SYSTEM PDL/DJDE REFERENCE
Option(s)  PRO CESS - Accepts U CSB for PRO CESS, use of a FO LD ED or UN FO LD ED translate table is controlled from the host system. For UN FO LD, characters not defined as printable in the U CSB are converted to blanks.

IGNORE - Suppresses the host transmitted universal character set buffer (U CSB). If IGNORE is coded, the CODE parameter of the VOLUME command defines the character translation. For online systems only.

Default  Default is PRO CESS

VFU

Syntax  LINE VFU = 

Option(s)  vfu-id - The command identifier of the VFU table, which must precede this reference to it. (Refer to the "Defining print line positions" section.) The VFU table defines print line positions corresponding to skip-to-channel parameters for the job to be processed.

NONE - Indicates that any skip-to-channel parameter is to be replaced by a carriage control of print and space 1.

Default  Default is NONE

Points to note—LINE command

FONTINDEX

Although the maximum bit-opt value may be 7 (allowing a maximum font index value of 127), the maximum number of fonts is constrained by the size of available input and output dynamic memory, font memory size, and the CME or PDE FONTS parameter. Also, internal table structures limit the number of fonts which can be invoked on a single page to 94 fonts.

• FONTINDEX and OVERPRINT
  — FONTINDEX, used in conjunction with overprinting, is handled as follows: For OVERPRINT=PRINT, the LPS overprints records analogously to an impact printer if the fonts are the same size. If the fonts differ, records are overprinted without regard to character spacing. For OVERPRINT=MERGE, the LPS replaces characters in the previous record which are blank. Character spacing values are adjusted; thus proportionally spaced or different size fonts may be used, and the LPS performs the character placement.
  — Characters in the previous record which are blank are replaced only when they are represented by an EBCDIC X40 or an ASCII X20. Binary character representations other than an EBCDIC X40 or an ASCII X20, which produce a blank character when printed with a particular font, is not replaced except in the following case.
During input processing, when a binary character representation is greater than the highest character in the specified font, a blank (EBCDIC 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.

• **Multiple fonts**

  — When multiple fonts of various sizes are used on a single page, the line spacing of any particular line is determined by the largest font in use in the previous line. The current line can also be adjusted downward by the difference in height between the first and the largest font in the current line. The line spacing of the first line in a report is determined by the first font in the PDE, provided the FONTINDEX parameter has not been specified. When FONTINDEX is specified, the line spacing of the first line of a report is determined by the font indicated by the font index value in the first line of the report. If there is no font index value in the record and FONTINDEX is specified, the line spacing for the first line of the report is determined by the first font in the PDE. When the FONTINDEX or CME parameters are specified, the line spacing of the first line of subsequent pages is determined by the line spacing of the font called out by the last font index of the previous page into the current PDL font list. If a PDE DJDE has occurred prior to the page transition, the font used for the line spacing comes from this list.

  — The height of the last font used determines the initial base line of that first print line.

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

• **Font index bytes**

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

  — If the offset to the font index value exceeds the logical record size, the first font in the font list is used, except when HOST=UNIVAC is specified. If HOST=UNIVAC is specified, the byte located at the specified font index offset is processed. The font index byte must be located at the beginning of the record if Univac Fielddata and ASCII records are intermixed within the file.

• **Print line data**

  When changing the print line data to a value greater than the default value (150), the LENGTH parameter of the RECORD command must also be changed accordingly. (Refer to the
“RECORD command” section of chapter 4, “Specifying input parameters.)

- **PCCTYPE**

The PCCTYPE=IBM3211/IBM4245 is not intended to be used when processing offline jobs. Checks that the LPS online interface performs, such as for bad or NO-OP parameters, are not done in the offline mode. There is no check for bad or NO-OP parameters, and they are treated as undefined carriage control parameters in the offline mode and default to the carriage control parameter of print-and-space-1-line.

**Examples—LINE command usage**

**Example 1**

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

The LINE command above defines the characteristics of the following record structure shown in figure 6-4.

**Example 2**

```
LINE PCCTYPE=NONE, DATA=(1,60), VFU=V1;
```

**Example 3**

```
LINE PCCTYPE=NONE, DATA=(1,57), VFU=V1;
```

The print-data-offset parameter of the DATA parameter (the number of bytes between the start of the user portion of the record and the first character of the record to be printed) is 1 byte. A sample is shown in figure 6-5. The print-length parameter (the number of characters in the longest print line in the record) is defined as 132 bytes. The carriage control is defined as IBM1403, and the position of the carriage control character is the first byte within the user portion of the record. The position of the font index (an index value into a font list that indicates which font is to be selected) is byte 133 (relative to 0).
**MESSAGE command**

The MESSAGE and ROUTE commands permit the user to inform the operator of special conditions. The MESSAGE command displays user-defined text to the operator.

**Example—MESSAGE command usage**

The following MESSAGE command informs the operator that blue paper is required for copy 2 of a 4 copy report. Printing is suspended at the appropriate points so that the operator can load the paper.

```
OUTPUT COPIES = 4;
MESSAGE ITEXT = ('COPY 2 WILL NEED BLUE PAPER'),
               OTEXT = ('LOAD BLUE PAPER', 2, WAIT),
               OTEXT = ('LOAD WHITE PAPER', 3, WAIT);
```

The parameters available for the MESSAGE command are summarized in table 6-7.

**Table 6-7. Summary of MESSAGE command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEXT</td>
<td>Message to the operator during input processing</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>OTEXT</td>
<td>Message to the operator during job printing</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>BTEXT</td>
<td>Specifies audit log data to the UI</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
MESSAGE command parameters

The following provides MESSAGE command parameter syntax and defines the parameter options.

**Note:** All "MESSAGE" parameters must be specified in a single command. If multiple message commands are specified, the parameters in the final command will be applied.

**ITEXT**

**Syntax**  
MESSAGE ITEXT=\textit{options}

**Function**  
Specifies a text message to be output to the operator during input processing.

**Option(s)**  
\textbf{NONE}  
Specifies that no text message is output to the operator during input processing.

- \textbf{sc} - Specifies a text message of up to 80 characters.
- \textbf{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.
- \textbf{WAIT} - Specifies that after the text is displayed, printing is suspended until the operator has responded with a \texttt{CONTINUE} parameter.

**Default**  
Default is \texttt{NONE}

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

**OTEXT**

**Syntax**  
MESSAGE OTEXT=\textit{options}

**Function**  
Specifies a text message to be output to the operator during job printing.

**Option(s)**  
\textbf{NONE}  
Specifies that no text message is output to the operator during job printing.

- \textbf{sc}  
  Specifies a text message of up to 80 characters (maximum of 400 characters per report).
- \textbf{passnum}  
  Specifies the pass (copy ply) to which the text applies. Multiple \texttt{sc} messages, one per \texttt{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.

**END**

Specifies that the text is displayed after the last copy of the
report is printed.

**Default**

Default is NONE

**Considerations**

Use the OTEXT message to inform the operator how to align the
bar code reader for a given job.

1. Code an OTEXT message (with a “WAIT” statement in the
data stream) that indicates the location and orientation that
the barcode reader needs to be set. The WAIT statement in
the OTEXT message stops output until the operator
completes the alignment task and presses the CONTINUE
key.

2. Lift the X8 top printer cover.

3. Ensure that the bar code reader matches the orientation
specified in the OTEXT message. If necessary, turn the bar
code reader 90 degrees in either direction.

4. Turn the green shaft knob to locate the bar code reader in
the position specified in the OTEXT command.

5. Ensure that none of the fingers cross the path between the
bar code reader and the paper path. If necessary, lift the
finger handle on the lower front bracket and move the
fingers forward or backward.

6. Close the X8 top cover.

7. Press the CONTINUE key.

Example OTEXT message:

“Open the X8 cover and position the bar code reader to
7.00” in the landscape orientation now. When you are
finished, close the cover and press the CONTINUE key.”

**BTEXT**

The BTEXT command supports the PC UI report audit logging
feature which creates an audit log on the UI hard disk. The audit
log is initiated and terminated by including enabling data in the
job stream. The PD/L/DJDE command, BTEXT, is implemented
primarily to support barcode audit logging. BTEXT report
definition parameters are, however, useful without the installation
of a barcode reader. Page definition parameters, on the other
hand, are meaningless without barcode text or a barcode reader.
Syntax

```
MESSAGE BTEXT=’[RNA=xxx] [,CJN=sss] [,DPT=xxx]
```

**Function**

Specifies bar code audit parameters.

**Option(s)**

- **RNA**: Report name (maximum of 16 characters). This keyword is required in the first BTEXT of each report to be audited. It may only occur once per report. For clarity in relating the audit reports to SFS reports, this should be the same value used for job accounting. This parameter must be present to audit a job or report.

- **CJN**: Customer job number (maximum of 6 digits). For clarity in relating the audit report to SFS reports, this should be the same value used for job accounting.

- **DPT**: Department name (maximum of 31 characters). For clarity in relating the audit reports to SFS reports, this should be the same value used for job accounting.

- **LDT**: Log stale date indicator. Once this date has passed, the audit log may be deleted or overwritten without warning to the operator. Options include:
  - **(U)**nique: specified that report is to be run only once, ever. Operator intervention at PC UI is required to re-run the report or delete the Audit Log.
  - **(W)**eekly
  - **(B)**i-**W**eekly
  - **(S)**emi-**M**onthly
  - **(M)**onthly
  - **(Q)**uarterly
  - **(*)** (Asterisk): Operator is not notified of re-runs but the report is renamed with a unique version as if the operator had been notified and accepted the re-run. Reports re-run with this indicator stay in the Audit Log directory until deleted by the operator using a PC UI window provided.

- **0..365**: specifies the number of days where zero days specified means the report can be run as frequently as desired, and will not be checked for accidental re-runs.

- **yy-mm-dd**: yy: year
  - mm: month
  - dd: day

- **RSQ**: Report sequence checking enablement. **(Y)**es indicates the audit report will flag out of sequence sheets delivered. This option must be set to **(N)**o if out of sequence sheets of a report are delivered. For out-of-
order deliveries, the job must be run in NO SEQ or BAR ALI mode so that the IOT will not catch and purge the mis-sequenced pages.

**DDE**  
Duplicate detection enablement. (Y)es indicates the audit report will flag duplicate sheets delivered. This option must be set to (N)o if multiple copies of a report or pages within the report are printed. For out-of-order deliveries, the job must be run in NO SEQ or BAR ALI mode so that the IOT will not catch and purge the mis-sequenced pages.

**DAR**  
The normal audit report option. DAR parameters are a combination of a summary/detail option and an exception data reporting option. For example, SUM-WMGT specifies a summary report with waste management data shown chronologically.

### Summary/detail options:

- **SUM** (Sumary) displays the audit report headings, the total of sheets delivered to each destination, including scrap, expected sheets if specified, and reconciles reports amount if specified.

- **DET** (Detail) displays the audit report headings, detail information about each sheet expected and delivered, the total of sheets delivered to each destination, including scrap, expected sheets if specified, and reconciles reports amount if specified.

### Exception data reporting options:

- **WMGT** specifies waste management data shown chronologically.

- **OPLOG** specifies system events, operator commands, and responses that may relate to the production of the report shown chronologically. Examples would be a job abort, ESS boot, etc.

- **BOTH** specifies both WMGT and OPLOG.

**Note:** These options are specified as compound keywords, for example ‘SUM-WMGT’. They are never specified alone; ‘SUM’ is illegal.

**RRA** Report reconciliation amount (the total monetary value of the report expected). Maximum of 11 digits plus a decimal point may be specified. Commas are not allowed. This option is required in order to print an audit report reconciliation on amount.

**NSE** Number of sheets expected for the report. Up to five digits, for a maximum value of 99999, may be specified. This option is required in order to print an audit report reconciliation on number of sheets consumed. The maximum value of 99999 takes into account non-barcode pages in the report.
PRINT FORMAT COMMANDS

PRD  Process date. If specified, it prints in the audit report header (yy-mm-dd).

Page detail parameters:

Page detail parameters are used in a BTEXT command for every page for which detailed auditing is desired. These parameters are meaningless without bar code text printed on the page, or bar code reader.

SEQ  Sequence number. It may be up to five digits, for a maximum value of 65534. This is required in order to identify non-delivered, out of sequence, or duplicate pages. The value of this parameter must match the bar code text to be printed on the page.

Note: This parameter is required if the PRA or TXT parameters are present.

PRA  Page reconciliation amount. This is the total value of the page, up to 11 digits plus a decimal point may be specified. Commas are not allowed. This option is required in order to print an audit report reconciliation on amount.

Note: This parameter is dependent upon the SEQ parameter.

TXT  Identifying text. It may be up to 64 characters. If present, prints as detail information for each page expected or delivered as reported on the audit report.

Note: This parameter is dependent upon the SEQ parameter.

Default value:

NONE  Specifies no BTEXT. If BTEXT is not specified, defaults to NONE.

Default  Default is NONE

Parameter rules

• All parameters have the form
  Keyword=value

• values must be specified as shown in the BTEXT command syntax. A value may be a keyword as shown in the BTEXT command syntax or text described as follows:
  xxx  Alphanumeric text, including spaces. Case sensitivity is retained
  nnn  Numeric text, including one optional decimal point
  sss  Numeric text; no decimal points
  yy-mm-dd  Numeric text describing year, month, and day. Each pair of characters is separated by a dash. Eight (8) characters in all are required.

• Parameters are separated by a comma

• values cannot have commas embedded
• White space within a parameter is ignored if it immediately precedes or follows the “=” sign.
• White space following a value is ignored
• White space preceding or following a comma is ignored.
• A comma following the last parameter is ignored
• Parameter keyword text is case insensitive.

Points to note

• If RNA (report name) is not specified, no audit logging will be performed.
• Barcode text that prints on the pages is placed in the job stream by the application to enable barcode features.
• Although the BTEXT command can be placed in the JSL, it will usually be placed in a DJDE as most of the parameters will be unique to a job. When placed in the JSL, it is specified under the PDL MESSAGE command.

Example:  
```
MESSAGE BTEXT='RNA=CK0409';
```

• Unlike the PDL OUTPUT command, where all parameters are treated independently from one another, BTEXT parameters are treated as one entity. This implies that parameters specified at the job level do not override those specified at the system level on an individual basis. However, as a whole, the job level BTEXT does override the system level BTEXT. For example, if the system level contains the RNA and DPT parameters, but the job level only contains the RNA parameter, INPUT Processing does not use the DPT from the system level and the RNA from the job level. Instead, INPUT Processing only uses the RNA (with no DPT) from the job level and sends that information to the PC UI.

Caution

A word of caution must be mentioned as to when BTEXT information is sent to the PC UI. For every report, INPUT Processing reads the JDE/JDL specified in the START command at report set-up time. In addition, it also performs the BTEXT JDE to JDL overrides. Although BTEXT is processed at this time, the information is not immediately sent to the PC UI until after the first record of the report is read and processed. If the first record is a DJDE BTEXT, the DJDE BTEXT overrides the JDE/JDL BTEXT. Again, no combining is done. If the JSL has RNA and DPT parameters, and the DJDE BTEXT has just an RNA parameter, only the RNA parameter from the DJDE BTEXT is used for auditing.

Once received by the PC UI, a set of PC UI BTEXT parameter override rules must be observed. At the start of every report, the PC UI enables receiving BTEXT with a new RNA parameter. When this parameter is received, the PC UI ignores any further report detail parameters within the same report. This implies that overriding occurs with report detail parameters received after the start of the report and before the RNA parameter. After the RNA parameter is received, if report and page detail parameters are intermixed within the same BTEXT command for future pages, only the page detail parameters are processed. If there are multiple values for a BTEXT parameter within the same packet sent to the PC UI,
only the last value is used. For example, if there are multiple
DPT parameters specified in a DJDE packet which also
contain the RNA parameter, the last DPT parameter is used.

- When a DJDE BTEXT is used, it must precede all data from
the page to ensure it will be forwarded to the PC UI before
the page is processed. The system controller (ESS) software
does not check for this. It is up to the user to code the
application correctly.

- Multiple DJDE BTEXT commands on a page must appear in
the same DJDE packet because only one page-oriented DJDE
packet can take effect per page.

- The BTEXT report parameter(s) must precede the first BTEXT
page detail parameter. The ESS software does not check for
this. It is up to the user to code the application correctly.

- Reports with multiple copies are not expected to be audited
in the normal course of operations, as in the case of check
printing; this implies multiple copies of checks.

- To accommodate PDL record length constraints, BTEXT
commands that do not fit in one PDL record can span
multiple records by grouping parameters under separate
BTEXT commands.

Example

```plaintext
MESSAGE BTEXT='RNA=CK0409,CJN=HR0525',
BTEXT='RRA=153249.37,NSE=2490',
BTEXT='LDT=03-07-94,RSQ=Y';
```

Note, however, that BTEXT parameters cannot span PDL
records. Every BTEXT parameter must start and end in the
same PDL record.

- To accommodate short record lengths in the datastream, the
information to be communicated in a BTEXT command can
be placed into two or more consecutive DJDE records within
a DJDE packet.

Example

```plaintext
$DJDE BTEXT='RNA=CK0409,CJN=HR0525,';
$DJDE RRA=153249.37,NSE=2490',END;
```

- The maximum length of all BTEXT strings within a DJDE
packet or in the JSL MESSAGE command is determined by
the following:

\[
508 - \text{N} \quad \text{where N is the number of BTEXT commands in}
\text{the JSL MESSAGE command or DJDE packet.}
\]

Due to the string length limitations, it is imperative that
BTEXT strings not be padded with unnecessary blank
characters.

Example 1:

```plaintext
$DJDE BTEXT='RNA=CK0409,CJN=HR0525'';
$DJDE BTEXT='RRA=153249.37,NSE=2490',END;
```

In Example 1, the maximum length allowed for the BTEXT string
is: \(508 - 2 = 506\) (N is 2 as there are two BTEXT commands).

Example 2:

```plaintext
MESSAGE BTEXT='RNA=CK0409,CJN=HR0525',
BTEXT='RRA=153249.37,NSE=2490',
BTEXT='LDT=03-07-94,RSQ=Y';
```

In Example 2, the maximum length allowed for the BTEXT string
is: \(508 - 3 = 505\) (N is 3 as there are three BTEXT commands).
• The maximum length of each DJDE BTEXT string is 255.
• Values for required BTEXT parameters not supplied in a DJDE or JSL will be supplied defaults by the PC UI if defaults exist for the parameter.
• BTEXT commands are treated as text strings by the PDL/DJDE compiler and forwarded to the PC UI to process. Therefore, it is possible that JSLs containing BTEXT will compile successfully even though errors exist in the BTEXT command string. It is up to the PC UI to process the BTEXT commands and flag any errors found. The same goes for BTEXT DJDEs.
• If BTEXT is specified and you are not running in PC UI mode, INPUT issues a warning message and allows the operator to continue or abort. If the operator chooses to continue, the BTEXT commands are ignored. Otherwise, the job is aborted.
• Although the BTEXT parameters can all be specified under one command, it is recommended that you group the report parameters and the page detail parameters into separate BTEXT commands.

Example 1: Sample minimal data to audit a report with monetary and sheet reconciliation.

```$DJDE BTEXT='RNA=CK0409,RRA=74698.32,NSE=4985',;
 $DJDE BTEXT='SEQ=1,PRA=153.65,TXT=901 R. Adams',END;```

Example 2: DJDE to print a report with sheet reconciliation only.

```$DJDE BTEXT='RNA=Personnel Reviews, CJN=HR052,NSE=249',;
 $DJDE BTEXT='SEQ=1,TXT=326769 Harold R. Adams',END;```

Example 3: Sample DJDE to audit a report with sheet reconciliation only, and without page details.

```$DJDE BTEXT='RNA=Admin 31,NSE=4230',END;```

Example 4: Sample JSL for a test run. The same report name will be used each time, and may be run as frequently as desired.

```MESSAGE BTEXT='RNA=Test01,LDT=0';```

Points to note—Audit Log

The 4635 Auditor features have great flexibility and allows you much freedom of use. The following guidelines offer a quick path for best results. Refer to the Xerox 4635 Operations Reference for additional information regarding the use of Audit Log.

Maximizing with minimal effort

1. Waste management and operational information can be very useful and is easy to obtain. The simplest way is to modify the JSL to name the report with the BTEXT RNA=Report Name parameter. Nothing else needs to be specified. However, when these jobs are run, the number of pages per report should be large. This results in just one or a few reports per job which allows the audit report to highlight waste management data. If a job consists of many reports, all those reports must be reviewed for waste management.
Use the Summary, Waste Management and Operation Detail options in the Audit Log Display (or Print) window to obtain a summary report with each waste management event and all logged operations.

2. Reconciliation, sequence error, and duplicate error auditing must be performed with job stream BTEXT. For simplest operation, place all BTEXT into the job stream instead of splitting BTEXT between JSL and the job stream. Select a report size that will facilitate auditing, for example, 5000 pages. Thus a job of 100,000 checks will be audited as twenty reports, and each audit report is available shortly after the last page of the report is printed.

3. When displaying the list of audit logs, use the error filters in the Audit Log window to only list reports with reconciliation, sequence, or duplicate errors. These can be specified singly or in combination with check boxes in the Audit Log window.

   Even faster is to list audited jobs with these filters. Then use the filters in the Audit Log Display (or Print) window to select which reports will print or display. For summary or detail, only the reports with the selected errors will be processed. Use the summary option and detail records with the selected errors will print or display.

   The current limit for log lists is about 1200. If this error occurs, try again with either a ‘Jobs’ list or filters. The current speed of log list display is about 18 seconds per 100 entries. Hitting the ‘List’ target before the list is displayed causes the list to immediately clear and another list to generate.

4. Use operator options to stop the printer on reconciliation or sequence errors. The Audit Log Administration window has a target labeled ‘Defaults.’ Click on this to open the Audit Log Defaults window. Check boxes stay in effect from session to session.

   When the job stops due to a reconciliation error, the operator may display the audit log with an option in the window that is displayed. This is not possible for sequence errors, however.

5. Jobs that consist of many small reports are not good candidates for waste management auditing as too many reports must be manually reviewed to see the waste management information.

   Other Audit Logging points to note

1. Delete logs regularly. For maximum control, audit logs are never automatically deleted. The following exceptions exist:

   a) A rerun of a report that had no stale date specified (LTD=0). The current log replaces the old log.

   b) A rerun of a report where the stale date has expired. The current log replaces the old log.

   Where stale dates are not specified, but rerun revisions are desired, the LDT= parameter provides revision numbering. These logs need timely administration to delete them when they lose usefulness.
2. Use the rerun report revision numbering feature judiciously. The Auditor assigns a revision number to reports if directed by the LDT= parameter. There is a limit, however, of 999 revisions per report. When this limit is reached, the revision is assigned 1. If revision 1 currently exists, that log will be replaced.

A job that has a single report name (perhaps specified in the JSL) and many reports will create many revisions. If this job has more than 500 reports, the job cannot be run twice without exceeding the rerun count limit of 999 unless the operator first deletes all logs for the first job. If the job has more than 1000 reports, it is impossible to audit all reports. The 1001th and successive reports will not log but will give an error indicating that the rerun count is exceeded.

3. Avoid jobs that have many short reports. As noted above, it is easier to examine a few reports rather than many small ones. The Audit provides filtering tools for both situations, but the storage, display and printing of larger reports is far more efficient for larger than small reports. If such jobs exist, analyze them to see if they can be consolidated into fewer reports. If small reports are necessary for other reasons, the job may not be a good candidate for auditing.

4. Do not use LDT=0 in your JSL unless the job has only one report. More than one report results in the earlier reports being overlaid by later reports. As a result, only one report will be audited, the last one, and it may contain unpredictable data from earlier reports in the job.

5. A JSL may contain a BTEXT statement and the job stream may contain BTEXT. If both JSL and job stream contain BTEXT, understand the following rules:

5.1 If the job stream BTEXT precedes the first data record, it replaces any JSL BTEXT. No combining is done. For example, if the JSL has RNA and DPT parameters, and the job stream has just an RNA parameter, only the RNA parameter from the job stream is used for auditing.

5.2 If job stream data precedes the job stream BTEXT, job stream BTEXT may be ignored for auditing, because Input has not yet seen the job stream BTEXT. When this job stream BTEXT is encountered, it is sent to the PC UI, but ignored unless the JSL BTEXT did NOT contain an RNA parameter and the job stream BTEXT DOES contain an RNA parameter.

5.3 If both JSL and job stream BTEXT are to take effect, JSL BTEXT must not have an RNA parameter, and job stream BTEXT must follow the first data record and contain the RNA parameter. Note that the RNA parameter must precede any page detail parameters.

6. There is only one way to code specific dates in BTEXT, which is yy-mm-dd, where yy is the last two digits of the year, mm is the month (example: January is 01), and dd are the two digits of the day of the month (example: the first is 01). Two numeric digits are required for each field, and each field
must be separated by one dash. Support for years in the
next century are not currently implemented.

Dates format on the Audit Log Reports and Windows
according to the currently selected PC UI language. (Not
currently implemented for reports.)

Audit Log and Audit Log Administration window unique stale
dates display as **/**/**.

Audit Log Display and Audit Log Print window unique stale
dates display as **/**/**.

Sequence errors are indicated by an asterisk (‘*’) to the left
of the sheet count. Duplicate errors are indicated by a plus
sign (‘+’) to the left of the sheet count. Both indicators
appear if both errors occurred on that sheet.

OUTPUT command

The OUTPUT command controls the organization and format of a
report. This includes control of the number of copies of a
report, the forms to be printed with the variable data, report
offsetting, and the use of simplex or duplex mode. Many
graphics features may be invoked with this command; these are
explained in greater detail later in this chapter and in the Xerox
4635 Laser Printing System Operations Reference. The
parameters available with the OUTPUT command are summarized
in table 6-8.
Table 6-8. Summary of OUTPUT command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFORM</td>
<td>Form only on back side of a duplex page</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>COLLABATE</td>
<td>Collated or uncollated report pages</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>COPIES</td>
<td>Number of copies of a report</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>COVER</td>
<td>Cover pages picked from the auxiliary tray</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>CYCLEFORMS</td>
<td>Associates forms with report pages in a cyclic fashion</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>DENSITY</td>
<td>IG local density functionality is listed for compatibility purposes. It is not supported on the 4635 LPS.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>DESTINATION</td>
<td>Output destination of print jobs</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>Printing on both sides of a page</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FACEUP</td>
<td>Face up delivery of pages</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>FEED</td>
<td>Paper tray selection</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Selection of page format descriptor</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FORMS</td>
<td>Associates forms with report copies</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>GRAPHICS</td>
<td>Whether graphics are used and how</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Initial scaling and positioning for batch mode graphics</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>INVERT</td>
<td>Allows control of image orientation</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Associates CMEs with report copies</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>NTO1</td>
<td>On a report basis, whether all copies of a particular report are printed first page to last (1-N) or last page to first (N-1)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>NUMBER</td>
<td>Automatic page numbering control</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>OFFSET</td>
<td>Report offset control</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>OSTK</td>
<td>Specification of ordered stock parameter</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PAPERSIZE</td>
<td>Paper size to be used for job</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>PURGE</td>
<td>Default disposition of document interleaved graphics</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RESOLUTION</td>
<td>Output coordinate system in spots per inch (spi)</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SHIFT</td>
<td>Shifts image in the y-direction</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SF1FUNCTION</td>
<td>Controls the sheet finisher function number 1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SF2FUNCTION</td>
<td>Controls the sheet finisher function number 2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SIZING</td>
<td>Controls matching the paper specified when the page is formatted with the available paper sizes in the trays</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>STOCKS</td>
<td>Calls out a specific STOCKSET to be used in the report</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SYSSPR</td>
<td>Specifies the system page paper size for a job</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TMODE</td>
<td>Specifies system throughput parameters</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TRANS</td>
<td>Specifies whether transparencies are used for the job</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>UNITS</td>
<td>Specifies user-selected units for positioning graphics</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>Shifts image in the x-direction</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
OUTPUT command parameters

The following provides OUTPUT command parameter syntax and defines the parameter options.

BFORM

Syntax

```
OUTPUT BFORM = (form-id,[init],[copies]
NONE
```

Function

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

Option(s)

form-id - Specifies a one-to-six character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, which is called an FSL file, with the FDL system task (refer to the Xerox LPS Forms Creation Guide). A form-id may be the keyword NONE.

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. Note that init copies is only valid offline.

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

NONE - Specifies that no form is to be added to the associated report page of variable data. init parameter specifies the beginning ply (pass) number to which a specified form applies. This defaults to the first or next copy.

Default

Default is NONE

COLLATE

Syntax

```
OUTPUT COLLATE = COLLABE-opt
```

Function

Specifies whether pages are to be collated or uncollated.
PRINT FORMAT COMMANDS

**Option(s)**

**YES** - Specifies that the output pages are collated.

**NO** - Specifies that the output is not collated. If DUPLEX=YES, then COLLATE=NO is ignored.

Note that if NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES and notes this action with a warning in the JSL.

Refer to the “Points to note—OUTPUT command” section for additional information.

**Default**

Default is YES

---

**COPIES**

**Syntax**

```
OUTPUT COPIES= number
```

**Function**

Specifies the number of report copies to be printed.

**Option(s)**

**number** - An integer value with range 1 (the default) to 32767.

For the copies DJDE parameter, a value of 0 may be specified for the DJDE COPIES parameter, in which case no copies are printed. The copies parameter on the START command allows an operator to override this parameter when initiating a print job.

**Default**

Default is 1

---

**COVER**

**Syntax**

```
OUTPUT COVER= {NONE
(FRONT,SEP)
BACK
(both,SEP)}
```

**Function**

Specifies that cover pages are to be picked from the AUX cluster. These cover pages may be placed at the front or back of each copy of a report.

**Option(s)**

**NONE**

Specifies that no cover pages are to be picked.

**FRONT**

Specifies that a cover page is to be picked (from the aux cluster) at the front of each copy and is the first page of the copy.

**SEP**

Specifies that each front cover does not have any data printed on it; also, no report data is printed on the back of a SEP cover.
BACK
Specifies that a cover page is to be picked at the end of each copy. No report data is printed on back covers.

BOTH
Specifies that both front and back cover pages are to be picked and the front of each copy is the first page of the copy.

Considerations
The FEED MAIN operator command (refer to the Xerox 4635 LPS Operations Reference) should be used in conjunction with this parameter to prevent the AUX tray from being automatically selected when the MAIN tray becomes empty. This can be used in combination with the FEED parameter on the OUTPUT command and the FEED DJDE parameter (refer to chapter 8, “Specifying dynamic job descriptor entries (DJDEs)”)

If RTEXT is specified alone with (FRONT, SEP) or (BOTH, SEP), RTEXT pages are printed on the covers, and front covers are picked only on copies for which there is RTEXT. (Refer to ROUTE commands for further details on RTEXT.)

Refer to the “Job parameter modification restrictions” section in chapter 8 for more information on COVER specified in a selected JDE.

Default
Default is NONE

CYCLEFORMS

Syntax
OUTPUT CYCLEFORMS = \{ form-id
(form-id1,form-id2)[,...]
NONE \}

Function
Specifies a set of forms to be associated with report pages in a cyclical fashion. Refer to “Examples— OUTPUT parameter usage.”

Option(s)
form-id - Specifies a 1 to 6 character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, called an FSL file with the FDL system task (refer to the LPS Forms Creation Guide for further details on FDL).

NONE - Means that no form is to be added to the associated report page of variable data.

If multiple CYCLEFORMS left parts on the same OUTPUT command are specified, a single list consisting of all forms specified in the OUTPUT command is created.

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

Example:
OUTPUT CYCLEFORMS=(form1, form2,);
DENSITY

Syntax

\[ \text{OUTPUT DENSITY} = \{ \begin{array}{c} \text{FIX} \\ \text{NO FIX} \\ \text{DEFAULT} \end{array} \} \]

Function

This command is listed for compatibility purposes only; it is not supported on the 4635 LPS.

Default

Default is DEFAULT

DESTINATION

Syntax

\[ \text{OUTPUT DESTINATION} = \{ \begin{array}{c} \text{BIN} \\ \text{TRAY} \\ \text{EXPORT} \end{array} \} \]

Function

Specifies the output destination of paper.

Option(s)

- **BIN** - Specifies that the currently selected bin is used.
- **TRAY**
  - Sends output to the sample tray.
  - Transparencies should always be sent to the sample tray.
  
  **Note:** Use caution when selecting the TRAY option because of the sample tray's limited capacity.
- **EXPORT** - Specifies that sheets be sent to the bypass transport. The EXPORT option takes effect only if the current operator commands SELECT E or SELECT AUTO are specified. SELECT AUTO places output destination under data stream control. SELECT E causes output to be sent to the bypass transport option.

Default

Default is BIN

DUPLEX

Syntax

\[ \text{OUTPUT DUPLEX} = \{ \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \} \]

Function

Specifies whether printing is to occur in duplex (print on both sides of a sheet) or simplex (print on a single side).
**PRINT FORMAT COMMANDS**

**FACEUP**

Syntax:  

```
OUTPUT FACEUP = {YES NO}
```

Function: Specifies faceup delivery of pages.

Option(s)  

- **YES** - Specifies that the sheets are delivered to the tray faceup. Specify FACEUP=YES for labels. Delivering the sheets faceup prevents delamination when stacking.
- **NO** - Specifies sheets are delivered facedown to the stacker tray. Refer to the “Points to note—OUTPUT command” section for additional information.

Default: Default is NO

**FEED**

Syntax:  

```
OUTPUT FEED = {stock-reference, stock-name, MAIN, AUX, OPR}
```

Function: Controls the stock on which the page is printed.

Option(s)  

- **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 users to change stocks associated with a job, without altering the stock references in the data application itself.
- **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.)
- **MAIN, AUX, and OPR** - Provide compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN.

Default: Default is OPR
FORMAT

Syntax

OUTPUT FORMAT= pde-id

Function

Specifies a page descriptor entry (PDE) to be used in formatting the printed output, such as location of starting print line for each logical page on the physical page, font usage, and page orientation.

Option(s)

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.

Standard pde-ids are defined in table 6-13, 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 (refer to chapter 8).

Details on the PDE parameter are discussed in the “Page descriptors” section of this chapter. Creating and compiling PDE files with the PDL processor is discussed in the “Compiling the JSL” section in chapter 3.

Note: To ensure reliability, enter the entire parameter FORMAT. Do not abbreviate to “FOR”.

Default

Default is FMT1

FORMS

Syntax

OUTPUT FORMS= \{ form-id \{ (form-id,\[init [,copies]]) \} NONE \}

Function

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

Option(s)

form-id - Specifies a 1 to 6 character file name (may be numeric, alpha, or alphanumeric) which exists on disk. This file is created by compiling a forms description language source file, called an FSL file, with the FDL system task (refer to the Xerox LPS Forms Creation Guide for further details on FDL).

init - Specifies the beginning ply (pass) number to which a specified form applies. This defaults to the first or next copy. If the copies parameter is not specified, the last (or only) specified form applies to all copies beginning with copy number init. If the form is not the last one specified, copies defaults to 1. If neither init nor copies is specified, the form applies to all copies of the report.

Default

Default is NONE.
copies - The number of copies of the report to be printed.

Note: To ensure reliability, enter the entire parameter FORM or FORMS. Do not abbreviate to “FOR“.

NONE - Means that no form is added to the associated report page of variable data.

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

Default: Default is NONE

GRAPHICS

Syntax  OUTPUT GRAPHICS=

\{ NO
YES
MOVE
BATCH
(YES,NO SUB)
(MOVE,NO SUB) \}

Function: Specifies how graphics are to be processed in a job.

Option(s)
NO - Indicates that there are no graphics in this job.
YES - Indicates there are graphics.
MOVE - Indicates that all referenced graphic disk files are to be copied into the print file.
BATCH - Means that the input stream contains no text, only graphics which are to be imaged, one per page.

NO SUB - Must be specified in the JDE using the GRAPHICS parameter if no substitution for the missing graphic image is desired. No substitution may be desired where graphics are specified for imaging with a reference scale factor that is greater than 4. Because the maximum effective scale factor is 8, graphic images in this case would be printed smaller than if resolution were 300 spi, (RESOLUTION=300). Note that this appearance error would not be indicated by the system.

Considerations
• If you are using the IMAGE parameter in conjunction with GRAPHICS, make sure to list the GRAPHICS parameter first, as these two parameters are order dependent.
• The Xerox 4635 LPS does not process 2-color graphics.

Examples:
OUTPUT GRAPHICS=BATCH,
RESOLUTION =300;

OUTPUT GRAPHICS=MOVE,
RESOLUTION =300,
PURGE=NO;

Default: Default is NO
IMAGE

Syntax

\[
\text{IMAGE} = (\text{vpos UN CM IN DOTS XDOTS}, \text{hpos UN CM IN DOTS XDOTS})^{,}^{,}^{,}^{,}^{,}[n/d]]
\]

Function

This parameter specifies, for batch mode processing only, the initial graphic imaging parameters to be used.

Option(s)

- **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 \text{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, centimeters (CM), inches (IN), or user-defined unit (UN). If no units are specified, inches are assumed. An xdot is 1/600 unit of measurement.

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

- **n/d** - Specifies the reference scale factor. Each parameter, n and d, must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied with that specified at the time the graphic was digitized and the product is rounded to the nearest integer in the range 1 to 8. That integer becomes the effective scale factor with which the graphic is imaged.

Considerations

If the UN option is used to define positioning, the UNITS parameter (described subsequently) must precede this command in the parameter line.

- The IMAGE parameter must follow \text{GRAPHICS=BATCH} parameter, as these two parameters are order dependent.
- If two IMAGE parameters are listed consecutively, the system will process only the second one.

For example:

\[
\text{OUTPUT GRAPHIC=BATCH, IMAGE} = (1 \text{ IN}, 2 \text{ CM});
\]

Default

Default is top left corner scaled at 1/1

INVERT

Syntax

\[
\text{OUTPUT INVERT} = \{\text{FRONT| BACK| BOTH| NONE}\}
\]

Function

This parameter allows you to invert the current position of the page. A page can effectively be rotated 180°—head-to-head and
head-to-toe. Inversion is also used for page orientation for
finishing functions.

**Option(s)**
- **FRONT**: Specifies page inversion on front sides
- **BACK**: Specifies page inversion on back sides
- **BOTH**: Specifies page inversion on front and back sides
- **NONE**: Specifies turning page inversion off

**MODIFY**

**Syntax**
```
OUTPUT MODIFY = 
  \{ cme-id
    (cme-id,[init],[copies])
  \}
  NONE
```

**Function**
Specifies CMEs to be associated with report copies.

**Option(s)**
- **cme-id**: The command identifier of a CME. The CME may be
coded within a JSL or coded and cataloged separately for use by
several JSLs. Any CME invoked by a DJDE must be cataloged
separately (refer to the “Cataloged CMEs” section in this chapter
for further information). If the CME is within the JSL, it must
precede a reference to its identifier by the MODIFY parameter.
Details on CME usage and syntax are discussed in the “CME”
section of this chapter.
- **init**: Specifies the initial ply (pass) to which the associated CME
is to be applied.
- **copies**: Specifies the number of plies (passes) on which to apply
the CME. If copies is not specified, then the CME applies to all
copies beginning with the copy number specified by init.
- **NONE**: Specifies that data are processed without modification.

Different CMEs may be associated with different copies of a
report by the use of multiple MODIFY parameters on the same
OUTPUT command. CME copy ranges may not overlap.

If different CMEs are associated with different copies of a report,
or if a CME is applied to some but not all copies of a report,
then the CMEs are termed copy sensitive. Copy-sensitive CMEs
may not be used if COLLATE=NO, or online processing is
specified. Refer to the MODIFY DJDE in chapter 8, “Specifying
dynamic job descriptor entries (DJDEs)” for a discussion of how
CMEs may be changed on a page-by-page basis within a report
ply.

**Default**
Default is NONE

**NTO1**

**Syntax**
```
OUTPUT NTO1 = \{ YES
    NO \}
```

XEROX 4635 LASER PRINTING SYSTEM PDL/DJDE REFERENCE 6-55
**Function**
Specifies, on a report basis, whether all copies of a particular report are printed last page to first (n-to-1). NTO1 is the command parameter keyword.

**Option(s)**
- **YES** - Specifies last page to first printing. 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. When YES is specified, the system assumes a 50 sheet threshold, or maximum report size.
  - **NO** - The default and specifies first page to last (1TON) printing.
  - **n** - The NTO1 threshold. It specifies the maximum number of pages a report may have in order for it to be printed NTO1. If, however, the threshold count is reached and an end-of-report condition is not met, the system prints the entire report 1TON. The default value of NTO1 is 50 physical sheets, but n may range from 1 to 32767. When a value is greater than 100, the system generates a warning message.

**Considerations**
- When NTO1=YES, PDL overrides COLLATE=NO with COLLATE=YES, and notes this action with a warning in the JSL listing.
- N-to-1 printing takes longer to process than 1-to-N printing because the system must process the entire report before beginning to print the job.
- NT01=YES causes FACEUP=YES to be specified.

**Default**
Default is NO

---

**NUMBER**

**Syntax**

```
OUTPUT NUMBER= { (pnum,lnum,cnum [,findex]) NO }
```

**Function**
Specifies page numbering on the output pages of a report. The page number character string is placed on the page at the specified line number based on the line spacing of the specified font. If override line spacing is specified for the font, it does not affect where the page number character string is to be printed.

**Option(s)**
- **pnum** - Page number: specifies the starting number (an integer) for page numbering. The beginning page number may be non-positive. The number is incremented at page transitions but not printed until it goes positive. The maximum page number that will be printed is 2,147,483,647.
- **lnum** - Line number: specifies an integer line number on which the page number is to be placed.
PRINT FORMAT COMMANDS

cnum - Column number: specifies an integer ending column number for the page number sequence.

findex - Font index: the list of fonts specified in the FONTS parameter of the PDE command. If not specified, the first font in the list is used.

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

Example:

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

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

Default Default is NO

OFFSET

Syntax OUTPUT OFFSET=

Function Specifies offset control on a report basis.

Option(s) ALL - Results in an offset of each copy of each report.
FIRST - Specifies that an offset is to occur only on the first copy of a report.
NONE - Specifies that there is to be no offset at any time.

The OFFSET control of FIRST, ALL, or NONE may be modified by the ROFFSET command (refer to the “Report offsetting on a record” section in chapter 7, “Using logical processing commands”). If OFFSET-FIRST or ALL and ROFFSET criteria is satisfied on page one of a report, the normal offset from the preceding report is nullified. No change should be made to the OFFSET parameter in a selected JDE.

Default Default is ALL

OSTK

Syntax OUTPUT OSTK=[[clu-def,[mod],[pos],[RES|NORES],[TABS],[size[DOTS|IN|CM]],[L|P]]]

Function OSTK specifies ordered stocks.

Option(s) clu-def - The name of the cluster containing the ordered stock (either a cluster-name or a cluster-reference). No default.
mod - The number of sheets in the ordered stock. For example, 5 cut tabs would have a mod value of 5. The default is 1 and the range of values is from 1 to 254.
pos - The initial position relative to the mod value. The default is 1 and the range of values is between 1 and mod.

RES or NO RES - Keyword to indicate if the pos value should be reset at the start of the next report. The default is RES.

TABS - Keyword to indicate a tab stock. The default is no tabs.

size - Width of the tab. The value can be specified in DOTS, IN or CM. The default is 0.5 in. This parameter only applies when tabs are specified.

L or P - Indicates which edge of the paper the tab is located. If L, it is located in the scan counting direction. If it is P, it is located in the dot counting direction. The default is P.

Default Default is 1 for mod and pos RES no tabs >0.5 for size P

PAPERSIZE

Syntax OUTPUT PAPERSIZE=

Function Identifies the paper size to be used for printing the job. Paper size measurements are as follows:

Option(s)

<table>
<thead>
<tr>
<th>Paper Size</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>16.54 by 11.69 inches (297 by 420 mm)</td>
</tr>
<tr>
<td>A4</td>
<td>8.27 by 11.69 inches (287 by 270 mm)</td>
</tr>
<tr>
<td>B4</td>
<td>10.12 by 14.33 inches (257 by 364 mm)</td>
</tr>
<tr>
<td>B5</td>
<td>7.17 by 10.12 inches (257 by 364 mm)</td>
</tr>
<tr>
<td>USLEGAL</td>
<td>8.5 by 14 inches (216 by 356 mm)</td>
</tr>
<tr>
<td>USLETTER</td>
<td>8.5 by 11 inches (216 by 279 mm)</td>
</tr>
<tr>
<td>(x,y)</td>
<td>a variable paper size, that is, a paper size that is not one of the standard sizes.</td>
</tr>
</tbody>
</table>

If papersize is not specified, the LPS uses the system default paper size specified during system generation.

Considerations

- When centimeters (CM) are used to specify paper size, the maximum allowable value is 32.767 CM.
- Paper size cannot be changed through DJDE on a page basis. Paper sizes can be changed on a report basis, however.
- Whether you use the keyword or the (x,y) method for specifying papersize in your JSL, you must use the same method in your forms source library (FSL).
- PAPERSIZE=B4 and PAPERSIZE=(10.12,14.33) produce identical results in that they indicate long-edge feed, as opposed to short-edge feed format. Refer to the “SEFFNT”
section in chapter 4, “Specifying input parameters,” for detailed information on short- and long-edge feed.

• B5 paper size is an option that requires that a retrofit kit be installed.

• Include performance considerations when switching paper sizes across pitch modes. Refer to “Using pitch and the TMODE parameter effectively” in this chapter.

Example:

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

Default: Default is sysgen paper size

### PURGE

**Syntax**

```
OUTPUT PURGE= {NO YES}
```

**Function**

Specifies whether or not the system should delete graphic disk files at the end of report. This is only true for document interleaved.

**Option(s)**

- **NO** - Specifies that graphic disk files should not be deleted.
- **YES** - Specifies that all IMG graphics files created or replaced by document interleaved graphic processing for this report should be deleted at the end of the report printing.

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

Default: Default is YES

### RESOLUTION

**Syntax**

```
OUTPUT RESOLUTION = {300 600}
```

**Function**

Defines the output coordinate system (300 or 600 spots per inch). 300 specifies that output, including fonts, forms, images, and logos, be printed at 300 spots per inch (spi). 600 refers to all fonts, forms, images, and logos in FN6, FR6, LG6, and IM6 directories. The Resolution statement is only included for compatibility. The 4635 LPS accepts 300 spi fonts, forms, images and logos and converts them to 600 spi using the majority interpolation scheme for printing. The 4635 cannot accept 600...
SPI fonts, forms, images, and logos. On systems printing at 600 spi, references in the JSL to 1IN, 2.54 cm, or 300 dots are converted to a pixel count of 600 spi in the JDL, the system converts all measurements to XDOTS.

**Default**

Default is sysgen parameter (300 spi on the Xerox 4635 LPS)

### SF1FUNCTION

**Syntax**

\[
\text{OUTPUT SF1FUNCTION = } \begin{cases} 
  \text{YES} \\
  \text{NO}
\end{cases}
\]

**Function**

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.

Specifies to the finisher whether (YES) or not (NO) to begin a particular finishing function on the current sheet.

**Default**

Default is NO

### SF2FUNCTION

**Syntax**

\[
\text{OUTPUT SF2FUNCTION = } \begin{cases} 
  \text{YES} \\
  \text{NO}
\end{cases}
\]

**Function**

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.

Specifies to the finisher whether (YES) or not (NO) to begin a particular finishing function on the current sheet.

**Default**

Default is NO

### SHIFT

**Syntax**

\[
\text{OUTPUT SHIFT= } \begin{cases} 
  (v_1, v_2) \\
  \text{YES} \\
  \text{NO}
\end{cases}
\]

**Function**

Image shift is the shifting of the image data (system page) relative to the physical page. This shifting is in the scan counting direction (vertical for landscape and horizontal for portrait). The SHIFT parameter specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data in the y-direction for three-hole paper, binding, finishing, and edgemarking.
Option(s)  

\textbf{v}_1 \text{ - 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 \text{ of an inch.) Range for } v_1 \text{ is } -75 \text{ to } 75.\text{ } 

\textbf{v}_2 \text{ - A value in dots for the amount of shift on the even (back) side of a duplex page. Range for } v_2 \text{ is } -75 \text{ to } 75. \text{ A standard size shift occurs if YES is specified (} v_1 = 75 \text{ and } v_2 = -75).\text{ } 

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

\textbf{NO} \text{ - Specifies that no shift occurs (} v_1 = 0 \text{ and } v_2 = 0).\text{ } 

If the origin of a print line falls off the page, the entire line of data fails to print. This is especially important for duplex portrait-oriented pages, where a miscalculated shift could result in the entire page failing to print. When duplex shift is used, the left margin must be at least .25 inches (75 dots) wide. When using the \textbf{SHIFT} parameter, make sure that the data is not shifted past the margin and off the page.

Refer to the “Points to note—\textbf{VOLUME} command” section in chapter 4, Specifying input parameters,” for additional information.

Default  

Default is NO

SIZING

Syntax  

\textbf{OUTPUT SIZING} = \quad \left\{ \begin{array}{l} \textbf{SEMIAUTO} \\ \textbf{BEST} \\ \textbf{EXACT} \end{array} \right\}

Function  

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

Option(s)  

\textbf{SEMIAUTO} 

The standard mode of operation.

\textbf{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.

\textbf{EXACT} 

Intended for use when the LPS is unattended because the LPS does not cycle down. If the exact paper size is not available, the system aborts the report.

A paper size is equal in size to another paper size if neither dimension differs by more than 15/600 of an inch.

Can be used to print on paper larger than the specified paper size. Data may be lost if the paper size is smaller than the image size.
Note that A4 paper is treated as smaller than 8.5 by 11 inch paper because it is narrower.

Regardless of the parameter selected, if an exact match in paper size is found, the matching tray is used. The parameters differ only when no match is found.

If there is not an exact paper size match in the effective cluster, but there is paper in the effective cluster larger than the specified size, then the following occurs:

- SEMIAUTO cycles down and requires operator confirmation. When the message CON O displays, and the larger size is confirmed, the larger paper is used. Further mismatches of this type are ignored until:
  - The STOCKS= parameter changes, or
  - End-of-report and no STOCKS=is in effect, or
  - End-of-job.
  - The printer cycles down. The operator must either press CONTINUE to conform the larger paper, change the paper or the effective cluster before the printer will cycle up. The operator can also abort the report.

BEST causes the following result:
- The larger paper is automatically used.

EXACT causes the report to be aborted if the printer does not cycle down.

**Default** Default is SEMIAUTO

### STOCKS

**Syntax** `OUTPUT STOCKS= stockset-name`

**Function** Defines the stockset and its associated stock(s) to be used in a report. If the stockset-name does not reference a STOCKSET command coded earlier in the same JDL, the system assumes that the stockset exists globally as an STK file, which is read at print time. In this case, PDL inserts a message in the JSL file listing, indicating that an STK file will be used at print time.

Whenever a new stockset is chosen at start of report or through a DJDE JDE and JDL switch, each stock is checked to determine that the stock exists and can be made active. This provides an automatic method of changing stocks in the LPS as required by the data stream.

Refer to the “STOCKSET command” section in chapter 5 for additional information on defining stocks.

**Default** No default
**SYSPPR**

**Syntax**

```
OUTPUT SYSPPR=
```

**Option(s)**

- USLEGAL - 8.5 by 14 inches (216 by 279 mm)
- A3 - 16.54 by 11.69 inches (420 by 297 mm)
- A4 - 8.27 by 11.69 inches (210 by 297 mm)
- B4 - 10.12 by 14.33 inches (257 by 364 mm)
- B5 - 7.17 by 10.12 inches (178 by 254 mm)
- USLETTER - 8.5 by 11 inches (216 by 279 mm)
- (x,y) - x defines the slow scan direction in inches. y defines the fast scan direction in inches.

**Considerations**

- When centimeters (CM) are used to specify paper size, the maximum allowable value is 32.767 cm.
- SYSPPR is not available as a DJDE.
- Whether you use the keyword or the (x,y) method, for specifying papersize in your JSL, you must use the same method in your forms source library (FSL).
- PAPERSIZE=B4 and PAPERSIZE=(10.12,14.33) produce identical results in that they indicate long-edge feed, as opposed to short-edge feed format. Refer to the “SEFFNT” section in chapter 4, “Specifying input parameters,” for detailed information on short- and long-edge feed.

**Default**

No default

---

**TMODE**

**Syntax**

```
OUTPUT TMODE=
```

**Function**

Specifies paper sizes in order to improve throughput efficiency.
**PRINT FORMAT COMMANDS**

**Option(s)**

- **width** - The process direction measurement that determines the default pitch mode.

- **width unit** - Defines the unit of measurement for the default pitch mode in IN, CM, DOTS, and XDOTS. IN (INCHES) is the default.

Width values can range from the 7-inch equivalent to the 17-inch equivalent.

If you use inches (the default), refer to table 6-17 in the “Points to note—OUTPUT command” section for the value ranges, pages-per-minute speeds, and uses available for the various pitch modes.

**Default**

Default depends on the specified paper size; default value is in inches

---

**TRANS**

**Syntax**

```
OUTPUT TRANS=
```

**Function**

Specifies whether transparencies are used for the print job.

Transparencies cannot be handled by the high-capacity feeder/stackers. Due to this restriction, transparency jobs must feed from the processor feed trays (trays 1 or 2; also known as MAIN and AUX feed) and stack to the sample tray.

**Option(s)**

- **YES** - Indicates that transparencies will be used in the job.

- **NO** - Indicates that transparencies will not be used in the job.

**Default**

Default is NO

---

**UNITS**

**Syntax**

```
OUTPUT UNITS= size in dots
```

**Function**

Specifies the position of a graphic when referenced by the IMAGE parameter of the OUTPUT command or an IMAGE, ALTER, or GRAPHIC DJDE parameter.

**Option(s)**

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

**Default**

No default
XSHIFT

Syntax

```
OUTPUT XSHIFT = NO
```

Function

Specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data for two-hole paper, binding, finishing, and edgemarking. XSHIFT allows shifting in the x-direction for short-edge binding.

Options

- **NO** - Specifies that no shift occurs
- **value** - An integer that specifies a shift amount on the front and back side. The shift is in the x-direction and each dot is 1/300 of an inch. The allowable range is 75 to -75.

Example:

```
INVPDE: PDE PMODE=PORTRAIT, BEGIN=(1.3,.37), FONTS=P0612A;
INVPDE: JDE; OUTPUT INVERT=BACK, SHIFT=YES, XSHIFT=-40, FORMAT=INVPDE, PAPERSIZE=(8.5,11);
```

XMP

Syntax

```
OUTPUT XMP = options
```

Function

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.

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 are printed in the highlight color mode.

Default

Default is DEFAULT
Points to note—OUTPUT command

These points to note list parameters followed by the applicable action or system response.

- **DUPLEX=YES**
  - Any job that runs in simplex mode runs in duplex mode except transparencies.
  - A recoverable imaging error results in one additional hole in the paper path and up to seven aborted pages to the sample tray. A recoverable dispatching or imaging error on an odd-numbered page results in one additional hole in the paper path with no aborted pages to the sample page.

- **COLLATE**
  Once you have specified multiple copies, the value of the COLLATE parameter cannot be changed again within the report.

- **NTO1, FACEUP, DUPLEX, COLLATE**
  The NTO1, FACEUP, DUPLEX, and COLLATE parameters follow a specific hierarchy by which one command parameter forces changes in certain other command specifications when a YES option is selected. This hierarchy of forced conditions is shown in table 6-9. The first column gives the condition that will force the parameters shown in the columns to the right.

<table>
<thead>
<tr>
<th>Parameter specification</th>
<th>Results in:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NTO1=</td>
<td>FACEUP=</td>
</tr>
<tr>
<td>NTO1=YES</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>FACEUP=YES</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

**Example**  
FACEUP= YES, COLLATE= NO;

**Example**  
NTO1= YES, FACEUP= NO;

- **NTO1 and 1TON system defaults**
  - During system generation, you may select either 1TON or NTO1 as the system default in the `DEFAULT ORDER={1-N | N-1}` window.
  - When the system default is 1TON, NTO1 is processed only if it is invoked in the JDE or JDL. The system ignores NTO1 in DJDE-selected JDEs or JDLs.
When the system default is NTO1, the system assumes each report is NTO1 until the first data page, even if 1TON is specified in the JDE or JDL. This arrangement allows you to use DJE-selected JDEs and JDLs that invoke NTO1 after a series of banner pages.

The following examples illustrate these conditions:

**Example 1:** The default order is 1TON, OUTPUT NTO1=NO is specified, and a JDE or JDL dynamic job descriptor entry (DJDE) containing NTO1=YES is invoked.

The system does not process the DJDE.

**Example 2:** The default order is 1TON and OUTPUT NTO1=YES is specified.

The system allows NTO1 printing if NTO1 threshold values (n) are within permitted limits. Note that NTO1 DJDEs are never allowed with the 1TON system default.

**Example 3:** The default order is NTO1 and OUTPUT NTO1=NO is specified.

The job prints 1TON unless a JDE or JDL DJDE specifying NTO1 is invoked in the first data record after banner pages.

**Example 4:** The default order is NTO1, OUTPUT NTO1=YES is specified, and a JDE or JDL DJDE specifying NTO1=NO is invoked.

The DJDE is processed and the job prints 1TON, if the system encounters it in the first data record after banner pages.

**NTO1, O STK**

When printing ordered stock in NTO1 sequence, note the following:

- The ordered stock must be reverse collated in the feeder so that the first sheet of the set is on the bottom, the second sheet next, and so on, with the last sheet of the set on top to be fed first. If the report is larger than the N to 1 threshold, it will print in 1 to N order and the reverse collated stock is then inappropriate. The system displays the following message and cycles down when this condition occurs:

  OS6908  Page ordering has been reversed. Check ordered stock.

**NTO1 and segments**

When finishing segments are defined in an NTO1 report, the entire report prints N to 1. The system does not process the segments separately. If the entire report contains more pages than the N to 1 threshold allows, even though individual segments may be less than the threshold, the entire report is printed 1 to N.

**NTO1=NO, FACEUP=NO, DUPLEX=NO**

- In this sequence, the NTO1 and FACEUP parameters specify the NO option so other potentially affected parameters, such as DUPLEX, cannot be overridden. The DUPLEX parameter, therefore, maintains either the YES or NO option.
If hierarchical overrides occur, the system issues warning messages in the JDL and then forces parameters lower in the hierarchical structure to their workable values.

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

If transparency (TRANS) is specified, the input task forces these parameters. Any operator key-ins, however, override the JSL specifications.

- **CYCLEFORMS, FORMS**

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

- **RAUX command, FEED parameter of OUTPUT command**

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

- **SYSPAGE parameter of STOCKSET command**

System pages are checked for ordered stocks through the SYSPAGE parameter. If there is no cluster name active for the current page, the input task checks to see if there was an operator FEED cluster-name specified. If so, the input task tests on the operator's specified cluster-name.

- **Ordered stock (OSTK)**

All applications that are printed on the tab have pages formatted for tab printing. If there is a PDE switch in the job stream that causes the BEGIN value to be shifted off the page, (BEGIN =(18, 8.6)), the cluster criteria must already be met before processing the new BEGIN. This allows the input task to bypass BEGIN value checking for tab stocks.

  — PDL ensures the minimum value for MOD is 1 and the maximum is 254 (decimal). It also ensures the value of POS is greater than or equal to 1 and less than or equal to MOD.

- **Printing on a tab**

  — The 4635 prints on tabs in both simplex and duplex modes.

  When printing on 9-inch stock or 8.5-inch stock with 1/2 inch tabs, the full imaging area of 9-inch is enabled. It is limited, however, by the amount of image shift enabled.

  — When running a tab stock job, you must not include the width of the tab when specifying the PAPERSIZE command. However, the width of the tab must be factored in when reporting the paper size to the system with the SET TRAY SIZE command on the “Tray size” screen or on the printer monitor.

  — A command DJDE FEED=Cluster name/cluster reference is required to switch to tab stock or whatever it would need to switch the current cluster to a tab stock cluster. This may include RAUX, RFEED, COVER=FRONT, operator keyin, etc.
INPUT processes tabs by keying off the current cluster name-reference at the time that the page is formatted for OUTPUT. For example, if the page satisfies the RAUX criteria and AUX is specified as the cluster name in the ordered stock statement, INPUT recognizes this as a tab stock. Having recognized it, INPUT then adds n dots (width of the tab) to the paper size in the slow scan direction. The IOT disables any dynamic width checking whenever it encounters a tab.

The portrait/landscape parameter of the OSTK statement allows either to be specified. The only one supported by the 4635 is portrait (P), and this is the default condition. If landscape (L) is selected, and the actual tab stock is loaded such that the tab is located on the registration edge (outboard), this combination will not run due to skew induced jams within the IOT.

In ordered stock tab jobs, care must be exercised when selecting the PFEED tray. Ordered stock jam recovery does not work properly if the tray selected for PFEED contains ordered stock.

Example

- Job 1 starts with Tab 1 (located in the upper right corner of a portrait page) and Job 2 must start with Tab 5 (located in the lowest right corner of the portrait page).

- The number of tabs Job 1 will use is 2.

In order to run the job in succession, you must code the JSL with MOD=5 and POS=1 for Job 1 and MOD=5 and POS=5 for Job 2. The 4635 does keep track of the last tab position in tray 2 at the end of Job 1. Upon starting Job 2, the 4635 purges Tab 3 and Tab 4 because Job 2 starts with POS=5 and Job1 used Tab 1 and Tab 2.

You can improve performance by ordering application dependent tab stock. By using ordered tab stock which supports only those number of tabs required for a specific job, eliminates the need for purging unused tabs.

If ordered stock is being fed from tray 7, it must be purged manually by the operator, and then OSTK must be used to set the stock position to reflect the contents of the feeder.

- At the time the system makes validity checks for BEGIN values and graphic positioning, input processing cannot determine if the current physical page is tab stock. However, if the OSTK command is specified, the input task will allow a tolerance of n inches (where n is the width of the tab specified in the OSTK command when performing validity checks on BEGIN values and graphic positioning). In addition, the FEED=tab-stock DJDE must precede any printable text data which will be printed on the tab.

- When running tab stock jobs:
  - Do not include the width of the tab when specifying papersize in the job setup.
- Include the width of the tab when reporting it to the printer on the printer control console “Tray size” screen or the system UI Set Tray Size command.

- Dedicate one tray for tab stock.

When using DJDEs, specify:

```
FEED=tab-stock
SIMPLEX=YES or DUPLEX=YES
```

When feeding from tab stock:

- Be sure to revert the DJDEs back to:

```
FEED=regular-stock
SIMPLEX=YES or DUPLEX=YES
```

- Ensure that PFEED does not specify the tray that contains the tab stock.

- Use the OSTK command before running a tab job.

• **Specifying a tab as a form**

To define a form that is to be used as a tab in a form callout, it is recommended that you code your applications with the following specifications:

- JSL must specify PAPERSIZE=USLETTER or (8.5, 11)

- FSL must specify PAPER IS 9 BY 11 or 8.5 by 11 with 1/2 inch tab.

- One tray must be dedicated for tabs. The JSL must specify which cluster name/reference contain the tab stock. Be sure to select an alternate tray as the PFEED source.

- A DJDE JDE, JDL switch that defines the following:

```
PAPERSIZE=(9, 11) or 8.5 by 11 with 1/2 inch tab
FEED=tab-cluster
DUPLEX=YES or DUPLEX=NO
FORM=XXXXXX
```

Width specified for ordered stock must be zero.

Specify TABS in the ordered stock statement so that dynamic width checking will be disabled.

A corresponding DJDE JDE, JDL switch to print regular paper stock. Note that the ordered stock statement specified in a DJDE JDE, JDL and a regular stock DJDE JDE, JDL must be exactly alike.

You may define the ordered stock statement at the job or the system level. If you define it at the job level, you must ensure that the ordered stock command defined from one JDE, JDL to the other are exactly alike. The reason is because the input task contains an internal mod/pos table that keeps track of the current mod/pos values for the job. This table is initialized at report set-up time through values found in the START JDE or JDL command. When a DJDE JDE or JDL is specified (with an ordered stock command), the input task will then reinitialize its internal mod/pos table with values in the
DJDE JDE or JDL if it finds that the ordered stock command found in the START JDE or JDL is different from those defined in the DJDE JDE or JDL. However, if both commands are alike, the input task will not reinitialize the internal mod/pos table. In order to check if the internal mod/pos table is being initialized, you will experience tab stocks that start purging before every tab is imaged (because MOD/POS is always initialized for every tab page).

- **Paper sizes**

The PAPERSIZE parameter is not supported as a DJDE. Therefore, a JDE page-oriented DJDE that calls out the desired paper size is needed.

- When specifying paper sizes, if the JSL specifies the paper size using the keyword method (such as USLEGAL, A4, or B4), then the FSL must also use the same method. If the JSL specifies 8.5,14 and FSL specifies USLEGAL, the INPUT will display an error message OS6670 FORM PAPER SIZE TOO SMALL. FORM WILL NOT BE PRINTED or OS6680 FORM PAPER SIZE TOO LARGE. FORM WILL NOT BE PRINTED USLETTER and B4 are the only exception to this rule where the keyword and x,y method will not create any incompatibilities.

- When printing system pages, if SYSPAGE=cluster is not specified, then INPUT will default to cluster MAIN. SYSPPR defines the paper size to use for the system page. Therefore, if cluster MAIN contains a mix of USLETTER and USLEGAL, then OUTPUT will search for the appropriate tray that contains the required paper size. If SYSPPR is not defined, then INPUT will use the current job's paper size.

- **SYSPPR**

The SYSPPR parameter will take effect for the following system pages:

- DJDE error page (OPRINFO parameter of the IDEN command)
- Tape label page (PLABEL parameter of the VOLUME command)
- Report delimiter page (PRINT parameter of the RSTACK command)
- Accounting page (ACCT command)
- Graphic error summary page
- Short-Edge Feed error page

SYSPPR defines a paper size, whereas SYSPAGE defines a cluster-name reference. Both parameters must agree in order to prevent errors, similar to the way the FEED and PAPERSIZE parameters interrelate to each other.

- **Using pitch and the TMODE parameter effectively**

Paper sizes each have an associated pitch mode.

There are six processing modes available on your LPS. These are called pitch modes. A pitch is a term describing the time it takes to image a page on the printer photoreceptor belt. A pitch mode is a phrase that describes how many pitches can occur
during one complete photoreceptor revolution. For example, while processing in 7-pitch mode, seven images can be placed on the photoreceptor belt during one revolution of the belt.

As page size increases, fewer images can be placed on the photoreceptor belt during one revolution. Since the belt runs at a constant speed, a decrease in the number of images that can be placed on the belt causes a corresponding drop in the pages-per-minute (ppm) throughput of the LPS. Tables 6-10 and 6-11 show the various pitch modes, their corresponding ppms, and the default paper sizes for each pitch mode. The paper size used in a job determines the highest pitch mode in which the system will run that job.

If you change paper sizes within a job, and this change crosses a pitch boundary, the system performs a xerographic quality adjustment, which may significantly impact performance.

Specifying the pitch (also called the throughput mode) with the TMODE parameter of the OUTPUT command allows you to control the default pitch mode with the following benefits:

- Matching system throughput with finishing device restrictions: If the finishing device attached to the printer cannot accept output at rated speed, the printer inserts gaps in the paper path to ensure that pages do not arrive at the device too rapidly. For example, if this is done at 135 ppm, it would effectively slow throughput to 68 ppm. This is true even if the finishing device could accept pages at up to 120 ppm. If you adjusted this 7-pitch job into 6-pitch mode, the job would run faster than 116 ppm.

- Eliminating process adjustment cycles during jobs that cross pitch boundaries: Each time a job crosses a pitch boundary, the system performs a xerographic quality adjustment. If this is done frequently within a job, the adjustments may take a substantial amount of the overall print time. Lowering the pitch mode so that the entire job runs at the same mode may enable you to significantly increase performance.

- Estimating the time needed to run your print jobs for scheduling purposes. To better understand the potential value of specifying pitch modes in your JSLs, consider these hypothetical situations:

  - Your finishing device folds and binds your printed output but accepts pages at a considerably slower rate than the 135 pages-per-minute capability of your 4635 laser printing system. You may want to slow down the rate at which your 4635 LPS sends printed sheets to this device by specifying a 4-pitch mode. Assuming that your application calls for 8.5 by 11-inch paper, you could use 5-pitch mode, but by selecting 4-pitch, you reduce the pages-per-minute (ppm) from 96 to 77, which better matches your finishing device's capabilities and reduces the potential for paper jams.

  - Your application uses four paper sizes, ranging from 8 by 11-inches for the first few pages to 11 by 17-inches for the last several pages. Each time your printer detects a different paper size, it adjusts to accommodate the new paper size, thus noticeably reducing throughput time. By specifying a large
enough pitch mode to accommodate all the paper sizes in your application, in this case, 3-pitch mode, the entire job runs at the maximum possible throughput speed because paper size adjustments are limited. Keep in mind that you may select a lower pitch mode (allowing a greater scan value) for smaller size paper but that you would not want to select a pitch mode too small to accommodate a larger paper size.

- You want to estimate how many applications you could run on a particular shift. But, given the size and complexity of your applications, you are uncertain as to how to plan. By looking at the paper sizes, and thereby, the pitch modes required by each application, and the number of pages in each application, and by associating these with the ppms provided in tables 6-10 and 6-11, you can then roughly estimate the length of time each application requires for printing.

As you can see, there are no absolutes and no set methods for accomplishing your printing goals. The variables involved include individual finishing device requirements, the paper sizes required by your applications, the complexity of each page, the number of pages in each application, and so forth. But a general understanding of the relationship between paper size and pitch mode, and the advantages of specifying pitch modes will help you optimize your printing activities.

- The 4635 LPS operates in multiple pitch modes: 3, 4, 5, 6, 7, and 8. Table 6-10 shows the pitch mode minimum and maximum boundary values, in inches, along with their associated print speeds, shown in pages per minute, which can be specified with the TMODE parameter of the OUTPUT command.

<table>
<thead>
<tr>
<th>IN</th>
<th>CM</th>
<th>MM</th>
<th>DOTS</th>
<th>XDOTS</th>
<th>Pages per minute (ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.31&lt; x ≤ 17.00</td>
<td>38.9 to 43.18</td>
<td>389 to 431.8</td>
<td>4595 to 5100</td>
<td>9190 to 10200</td>
<td>58</td>
</tr>
<tr>
<td>12.12&lt; x ≤ 15.31</td>
<td>30.8 to 38.9</td>
<td>308 to 389</td>
<td>3639 to 4594</td>
<td>7277 to 9189</td>
<td>78</td>
</tr>
<tr>
<td>10.19&lt; x ≤ 12.12</td>
<td>25.9 to 30.8</td>
<td>259 to 308</td>
<td>3060 to 3638</td>
<td>6119 to 7276</td>
<td>98</td>
</tr>
<tr>
<td>9.01&lt; x ≤ 10.19</td>
<td>22.9 to 25.9</td>
<td>229 to 259</td>
<td>2705 to 3059</td>
<td>5410 to 6118</td>
<td>117</td>
</tr>
<tr>
<td>7.40&lt; x ≤ 9.01</td>
<td>18.8 to 22.9</td>
<td>188 to 229</td>
<td>2221 to 2704</td>
<td>4442 to 5409</td>
<td>135</td>
</tr>
<tr>
<td>7.00&lt; x ≤ 7.40</td>
<td>17.8 to 18.8</td>
<td>178 to 188</td>
<td>2100 to 2220</td>
<td>4205 to 4441</td>
<td>154</td>
</tr>
</tbody>
</table>

- In 7-pitch mode, the system can process paper from 8 inches by 10 inches to 9.01 inches by 14.33 inches (203 x 254 to 229 x 364 mm) in a long-edge feed mode. Long-edge feed mode means the sheet is fed
into the paper path with its long edge first. In 7-pitch mode, the 4635 prints with full productivity at 135 pages per minute, with seven page images per photoreceptor revolution. For papers whose width ranges from 8.5 inches to 9.01 inches, the 4635 prints with slightly reduced productivity. The reason for this is that for papers in this width range, the normal image area extends into the inter-image zone that is normally used for xerographic process controls. In order to maintain the best productivity possible, xerographic process controls are temporarily suspended. Periodically, the 4635 will induce about 14 skipped “images” in order to perform a xerographic process control check and ensure that the 4635 maintains the highest print quality. It will then resume printing at full productivity.

For any paper size wider than 15.32 inches, 389 mm, in width, the system automatically switches to 3-pitch mode, which prints up to 58 pages per minute. For very large paper, 11 inches by 17, inches for example, paper is short edge fed, so that the 11 inch side leads through the printer.

— You may also specify pitch mode boundaries in centimeters (CM), DOTS, and XDOTS. The minimum and maximum scan values with these units of measurement and the pitch mode ppm print speeds are shown in table 6-10.

• **Purging stock**

— The 4635 LPS can purge tab sheets whenever the tab cycle is not in sync with the formatted data. This may be caused by paper jams or running multiple jobs in succession. For example, if a typical application had the following scenario:

— Two tab stock jobs to run in succession. Both jobs require tray 1.

— Tray two contains the tab stock which has MOD=5.

— Job 1 starts with Tab 1 (located in the upper right corner of a portrait page) and Job 2 must start with Tab 5 (located in the lowest right corner of the portrait page.)

— The number of tabs Job1 will use is 2.

In order to run the job in succession, you must code the JSL with MOD=5 and POS=1 for Job 1 and MOD=5 and POS=5 for Job2. The 4635 will keep track of the last tab position in tray 2 at the end of Job 1. Upon starting Job 2, the 4635 LPS will purge Tab 3 and Tab 4 because Job 2 starts with POS=5 and Job 1 used Tab 1 and Tab 2.

You can improve performance by ordering application-dependent tab stock. By using ordered tab stock which supports only those number of tabs required for a specific job, you will eliminate the system from purging unused tabs.
Examples—OUTPUT command usage

Example 1

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

Example 2

```
OUTPUT COPIES=3, DUPLEX=Yes, SHIFT=Yes, OFFSET=FIRST,
NUMBER=(1, 66, 132), COVER=(FRONT, SEP), FORMS=GBAR;
```

Pages printed with this OUTPUT command are in duplex mode with a margin shift of 75 dots. Three copies of the report are made, the first of which is offset; the second and third copies are stacked on top of the first. Each page is numbered at the end of line 66. The form GBAR appears on each data page of the report.
Example 3

\[
\begin{align*}
\text{OUTPUT COPIES} &= 2, \\
\text{OFFSET} &= \text{ALL}; \\
\text{COPY 2 REPORT 2} &\quad \text{COPY 1 REPORT 2} \\
\text{COPY 2 REPORT 1} &\quad \text{COPY 1 REPORT 1}
\end{align*}
\]

\[
\begin{align*}
\text{OUTPUT COPIES} &= 2, \\
\text{OFFSET} &= \text{FIRST}; \\
\text{COPY 2 REPORT 2} &\quad \text{COPY 1 REPORT 2} \\
\text{COPY 2 REPORT 1} &\quad \text{COPY 1 REPORT 1}
\end{align*}
\]

\[
\begin{align*}
\text{OUTPUT COPIES} &= 2, \\
\text{OFFSET} &= \text{NONE}; \\
\text{COPY 2 REPORT 2} &\quad \text{COPY 1 REPORT 2} \\
\text{COPY 2 REPORT 1} &\quad \text{COPY 1 REPORT 1}
\end{align*}
\]

The commands above illustrate use of the offsetting features of the \text{OUTPUT} command. The parameter \text{ALL} offsets all copies of all reports; \text{FIRST} offsets just the first copy of each report and \text{NONE} inhibits offsetting.

Example 4

\[
\begin{align*}
\text{OUTPUT CYCLEFORMS} &= (\text{FORM1, NONE, FORM3}); \\
\text{or} \\
\text{OUTPUT CYCLEFORMS} &= (\text{FORM1, FORM3});
\end{align*}
\]

If there are ten pages in the report, either of the above commands would cause \text{FORM1} to be applied to pages 1, 4, 7, and 10; no forms to be applied to pages 2, 5, and 8; and \text{FORM3} to be applied to pages 3, 6, and 9.

Example 5

\[
\begin{align*}
\text{OUTPUT COPIES} &= 2, \\
\text{COLLATE} &= \text{NO}, \text{DUPLEX} = \text{NO}, \\
\text{CYCLEFORMS} &= (\text{FORM1, NONE}); \\
\text{COPY 2 PAGE4} &\quad \text{COPY 1 PAGE4} \\
\text{COPY 2 PAGE3} &\quad \text{COPY 1 PAGE3} \\
\text{COPY 2 PAGE2} &\quad \text{COPY 1 PAGE2} \\
\text{COPY 1 PAGE1} &\quad \text{COPY 1 PAGE1} \\
\text{FORM1} &\quad \text{FORM1}
\end{align*}
\]

\[
\begin{align*}
\text{OUTPUT COPIES} &= 2, \\
\text{COLLATE} &= \text{YES} \\
\text{DUPLEX} &= \text{YES} \\
\text{CYCLEFORMS} &= (\text{FORM1, NONE}) \\
\text{COPY 2 PAGE4} &\quad \text{COPY 1 PAGE4} \\
\text{FORM1} &\quad \text{COPY 2 PAGE3} \\
\text{COPY 2 PAGE2} &\quad \text{COPY 2 PAGE1} \\
\text{COPY 1 PAGE1} &\quad \text{COPY 1 PAGE1} \\
\text{FORM1} &\quad \text{FORM1}
\end{align*}
\]

The commands above illustrate the use of cycleforms with multiple copies. Note that every copy of a particular page has the same form whether \text{COLLATE}=\text{YES} or \text{NO}. The pages in the preceding example are not logical pages but are one side of a logical sheet of paper.
PDE command

The PDE command specifies a page descriptor entry (PDE) which defines formatting information for each page of a report. This formatting information includes page orientation (landscape or portrait), location of the beginning print line for each logical page, and the fonts to be used. The parameters available for the PDE command are summarized in table 6-11.

Table 6-11. Summary of ac:PDE command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td>Location of starting print line on page</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>FONTS</td>
<td>Fonts for printing variable and CME data</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PMODE</td>
<td>Landscape or portrait printing mode</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>

You must specify an identifier (of type ac) when defining the PDE and reference it with the FORMAT parameter of the OUTPUT command. The PDE command must precede the OUTPUT command.

PDEs may be coded as part of the JDL or created as separate files so that they may be referenced by one or more JDLs or by DJDEs. PDEs are called out on the FORMAT parameter of the OUTPUT command or a DJDE FORMAT parameter. PDEs must be precompiled to be accessed by a DJDE. Standard PDE specifications are provided on the operating system software (OSS) system tapes as described in table 6-12.

A PDE created as a separate disk file may be used as if it were part of the JDL that references it. This is done by creating a JSL file containing only PDE commands and using the PDL processor to compile it. For each PDE command, PDL creates a control file on disk, cataloged in the PDE directory. The PDE command identifier is used as the file name. When the PDE is referenced, the system searches the PDE directory for the named PDE and, if found, loads it into memory for use in processing the report.

Refer to the “Compiling the JSL” section in chapter 3 for further information on this feature.
Table 6-12. **Standard LPS print format**

<table>
<thead>
<tr>
<th>PDE id</th>
<th>No. of lines</th>
<th>No. of columns</th>
<th>lpi</th>
<th>cpi</th>
<th>Approx. point size</th>
<th>Page size and orientation*</th>
<th>BEGIN values</th>
<th>Default font id</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMT1</td>
<td>66</td>
<td>132</td>
<td>8.1</td>
<td>13.6</td>
<td>9</td>
<td>11 by 8.5</td>
<td>(.18,.66)</td>
<td>L0112B</td>
</tr>
<tr>
<td>FMT2</td>
<td>66</td>
<td>150</td>
<td>8.1</td>
<td>15</td>
<td>9</td>
<td>11 by 8.5</td>
<td>(.18,.50)</td>
<td>L0212A</td>
</tr>
<tr>
<td>FMT3</td>
<td>88</td>
<td>132</td>
<td>10.7</td>
<td>13.6</td>
<td>7</td>
<td>11 by 8.5</td>
<td>(.14,.66)</td>
<td>L0312A</td>
</tr>
<tr>
<td>FMT4</td>
<td>88</td>
<td>150</td>
<td>10.7</td>
<td>15</td>
<td>7</td>
<td>11 by 8.5</td>
<td>(.14,.50)</td>
<td>L0412A</td>
</tr>
<tr>
<td>FMT5</td>
<td>49</td>
<td>100</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>11 by 8.5</td>
<td>(.17,.50)</td>
<td>L0512A</td>
</tr>
<tr>
<td>FMT6</td>
<td>80</td>
<td>100</td>
<td>8.1</td>
<td>13.6</td>
<td>9</td>
<td>8.5 by 11</td>
<td>(.57,.58)</td>
<td>P0612A</td>
</tr>
<tr>
<td>FMT7</td>
<td>60</td>
<td>90</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>8.5 by 11</td>
<td>(.50,.50)</td>
<td>P07TYA</td>
</tr>
<tr>
<td>FMT8</td>
<td>60</td>
<td>75</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>8.5 by 11</td>
<td>(.50,.50)</td>
<td>P0812A</td>
</tr>
<tr>
<td>FMT9</td>
<td>80</td>
<td>200</td>
<td>10.0</td>
<td>20.0</td>
<td>7</td>
<td>11 by 8.5</td>
<td>(.25,.25)</td>
<td>L0912A</td>
</tr>
<tr>
<td>FMT10</td>
<td>132</td>
<td>132</td>
<td>12.5</td>
<td>17.6</td>
<td>6</td>
<td>8.5 by 11</td>
<td>(.22,.51)</td>
<td>P1012A</td>
</tr>
<tr>
<td>FMT11</td>
<td>132</td>
<td>150</td>
<td>12.5</td>
<td>20.0</td>
<td>6</td>
<td>8.5 by 11</td>
<td>(.22,.50)</td>
<td>P1112A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.5 by 11</td>
<td>(.50,.50)</td>
<td>P0612A</td>
</tr>
<tr>
<td>FMT13</td>
<td>66</td>
<td>172</td>
<td>8.1</td>
<td>13.6</td>
<td>9</td>
<td>14 by 8.5</td>
<td>(.18,.66)</td>
<td>L0112B</td>
</tr>
<tr>
<td>FMT12</td>
<td>104</td>
<td>100</td>
<td>8.1</td>
<td>13.6</td>
<td>9</td>
<td>8.5 by 14</td>
<td>(.57,.58)</td>
<td>P0612A</td>
</tr>
<tr>
<td>FMT1A</td>
<td>66</td>
<td>132</td>
<td>8.3</td>
<td>12.5</td>
<td>9</td>
<td>11.69 by 8.27</td>
<td>(.18,.57)</td>
<td>R112BL**</td>
</tr>
<tr>
<td>FMT2A</td>
<td>66</td>
<td>150</td>
<td>8.3</td>
<td>14.3</td>
<td>9</td>
<td>11.69 by 8.27</td>
<td>(.18,.60)</td>
<td>R212BL**</td>
</tr>
<tr>
<td>FMT3A</td>
<td>88</td>
<td>132</td>
<td>11.1</td>
<td>12.5</td>
<td>7</td>
<td>11.69 by 8.27</td>
<td>(.18,.57)</td>
<td>R312BL**</td>
</tr>
<tr>
<td>FMT4A</td>
<td>88</td>
<td>150</td>
<td>11.1</td>
<td>14.3</td>
<td>7</td>
<td>11.69 by 8.27</td>
<td>(.18,.60)</td>
<td>R412BL**</td>
</tr>
<tr>
<td>FMT5A</td>
<td>48</td>
<td>100</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>11.69 by 8.27</td>
<td>(.22,.85)</td>
<td>R512BL**</td>
</tr>
<tr>
<td>FMT6A</td>
<td>80</td>
<td>100</td>
<td>8.1</td>
<td>13.6</td>
<td>9</td>
<td>8.27 by 11.69</td>
<td>(.91,.46)</td>
<td>R612BP**</td>
</tr>
<tr>
<td>FMT7A</td>
<td>60</td>
<td>90</td>
<td>6</td>
<td>12</td>
<td>12</td>
<td>8.27 by 11.69</td>
<td>(.85,.39)</td>
<td>R7TIBP**</td>
</tr>
<tr>
<td>FMT8A</td>
<td>60</td>
<td>75</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>8.27 by 11.69</td>
<td>(.85,.39)</td>
<td>R812BP**</td>
</tr>
<tr>
<td>FMT9A</td>
<td>80</td>
<td>200</td>
<td>10.0</td>
<td>20.0</td>
<td>7</td>
<td>11.69 by 8.27</td>
<td>(.14,.85)</td>
<td>R912BL**</td>
</tr>
<tr>
<td>FMT10A</td>
<td>132</td>
<td>132</td>
<td>12.5</td>
<td>17.6</td>
<td>6</td>
<td>8.27 by 11.69</td>
<td>(.57,.39)</td>
<td>RA12BP**</td>
</tr>
<tr>
<td>FMT11A</td>
<td>132</td>
<td>150</td>
<td>12.5</td>
<td>20.0</td>
<td>6</td>
<td>8.27 by 11.69</td>
<td>(.57,.39)</td>
<td>RB12BP**</td>
</tr>
</tbody>
</table>

* First dimension given is the horizontal dimension. Thus, 11 by 8.5 is landscape orientation, while 8.5 by 11 is portrait orientation. (Dimensions are given in inches.)

** A4 European international formats.
Multiple logical pages on physical page

Multiple BEGINs define multiple logical pages (a user-defined page image bordered by top of form and bottom of form, left and right margins) on one physical page (one side of a sheet). Up to 63 logical pages may be defined per physical page. These logical pages may be defined in any order on the physical page and placed on the physical page in the order the BEGIN parameters appear in the PDE command. The first BEGIN specified, whatever its physical position on the page, is considered logical page one. Structure your variable data in the same order that the logical pages are defined, using either spacing or skipping printer carriage controls to move from one logical page to the next. Generally, a skip-to-channel-1 parameter is the easiest way to position to the next logical page. The following should be noted:

- Each online banner page is positioned as the first logical page of a new physical sheet (if BANNER command parameter TYPE=BANNER).
- ROFFSET causes the logical page containing the matching criteria to be the first logical page of a new physical sheet.
- Page numbering occurs on each logical page rather than each physical page.
- Commands that specify line numbers, such as CRITERIA, VFU, CME, and NUMBER, refer to the line on the current logical page (ranging from TOF to BOF).
- RTEXT is imaged on a separate physical sheet. Line and column for positioning of the text refers to the first logical page specified in the PDE.
- RAUX criteria found on any logical page on the physical page causes that sheet to be picked from the auxiliary tray.
- Accounting statistics are accumulated on the basis of physical pages.
- If any logical page overflows the physical page before encountering BOF, the next line is positioned to the first logical page of the next side, and all subsequent logical pages are repositioned on BEGIN.
- With the exception of DEPT and SHIFT, all page-oriented DJDEs are applied at a logical page boundary.
- The following DJDEs, when applied, cause the current page position to go to the first logical page of the front side of a new physical sheet:
  - DUPLEX
  - COPIES
  - JDL
  - JDE
  - BFORM
  - OTEXT
  - RTEXT
The following DJDEs when applied cause the current page position to go to the first logical page of the next physical side:

```
FORMS
FORMAT
```

## BEGIN

**Syntax**

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

**Function**

Specifies the location of the starting print line of a logical page for graphics. The `vpos` parameter specifies the vertical position of the first character of the first print line on the logical page. It may be specified in inches (IN) or centimeters (CM). The default is inches.

**Option(s)**

- `vpos` - Specifies a decimal number with up to three digits to the right of the decimal point, such as 0.563 IN and 2.35 CM, are all legal specifications). The default is .18 IN.
- `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.

**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.

**Default**

Default is (.18 IN, .66 IN)

## FONTS

**Syntax**

```
ac:PDE FONTS=\{(f1,[f2],[...])|((f1,s1),(f2,s2),[...])|((f1,s1,[LPI|XDOTS|DOTS]),(f2,s2,[LPI|XDOTS|DOTS]),[...])\}
```

**Function**

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

**Option(s)**

- `f` - Specifies a one-to-six alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability).
- `s` - Specifies an optional override line-spacing value defined by either the print lines per inch (LPI) or the dots per print line (DOTS) to be associated with the font. Maximum value for `s` is 30 LPI; minimum value is 10 dots. If unit (LPI or DOTS) 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.

Default
No default for LPI for $s_1$

PMODE

Syntax
\[
\text{ac:PMODE} = \begin{cases} \text{LANDSCAPE} \\ \text{PORTRAIT} \end{cases}
\]

Function
Specifies the printing mode for each physical sheet.

Option(s)
- LANDSCAPE - Indicates that printing is to be parallel to the long edge of paper.
- PORTRAIT - Indicates that printing is to be parallel to the narrow edge of paper.

Default
Default is LANDSCAPE

Points to note—PDE command

- The maximum number of fonts that may be used in printing a given page or a given job varies depending on a number of factors. For both the input and output tasks, the number of fonts specified in the FONTS parameter must be at least equal to the largest number of fonts required to print any page within the job.

  Particularly for the input task, the limitations of dynamically allocated memory, (that is, when input is unable to receive an allocation of dynamic memory, the job is aborted), impose additional constraints which effectively reduce the maximum number of fonts that may be used to print a page. The number of fonts specified in the FONTS parameter is used to determine the size of a dynamically allocated table where information about the fonts is cached.

  If the patch to enable the use of Merganthaler fonts is selected when the LPS is sygened, the use of dynamic memory for fonts increases significantly since a proportionally spaced font requires an additional 120-260 bytes of dynamic memory.
The maximum block length for an offline job can be a size such that the allocation of one input buffer or two of equal size can cause dynamic memory allocation problems when used in conjunction with a large PDE FONTS parameter and with Merganthaler fonts enabled.

- An xdot is 1/600 unit of measurement. A form specifying xdots, however, can be created, edited, and compiled on any LPS with version 3 software, but the form, which is 600 spots per inch (spi), does not print on your 4050, 4090, or 4635 which accepts only 300 spi.

Examples—PDE command usage

Example 1

```
PDE1: PDE PMODE=LANDSCAPE,BEGIN=(.861,.7),FONTS=(L0112B,L01BOB);
PDE2: PDE PMODE=PORTRAIT,BEGIN=(1.3,.37),FONTS={(P08TYA,6.8),(P08OAA,6.8)};
LI: JDE;
   OUTPUT FORMAT=PDE1;
P1: JDE;
   OUTPUT FORMAT=PDE2;
```

Two PDEs are defined and referenced in separate JDEs. PDE1 specifies a landscape page and two landscape fonts; PDE2 defines a portrait page and 2 portrait fonts with override line spacing.

Example 2

```
2UP: PDE BEGIN=(.5 IN,.5 IN),BEGIN=(6.5 IN,.5 IN),
    PMODE=PORTRAIT,FONTS=P1012B;
```

Example of multiple BEGINs. Two logical pages are defined on one physical page.

Example 3

```
PDE99:PDE PMODE=PORTRAIT,BEGIN=(.10,0),
    FONTS=(P07ITA,P08OAA);END;
```

Example of a PDE that may be compiled by PDL separately from a JDL. After compilation, a PDE object file is created in the PDE directory which then may be referenced from within a JDL or by a DJDE. The name of the object file in the PDE directory is PDE99 (the same as its identifier on the source command).

ROUTE commands

The ROUTE command allows the user to print identifying information (text and an optional form) on the page preceding a report. The ROUTE command may be coded within a JDL or in a disk file of ROUTE commands.

The parameters available for the ROUTE command are summarized in the following table. A detailed description of these parameters is provided in table 6-13.
Table 6-13. **Summary of ABNORMAL command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFORM</td>
<td>Form to be printed with RTEXT preceding a report copy.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>RTEXT</td>
<td>Text to be printed preceding a report copy.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

**Cataloged RTEXT files**

When there are a large number of RTEXT parameters, they should be precompiled and stored in a disk file instead of including them in a JDL. After they are compiled, they may be accessed (by referring to their object file name) from either a JDL or a DJDE. RTEXT parameters must be precompiled to be accessed with a DJDE.

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

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

After the above commands are compiled by PDL, a file is created under the TST file directory, such as ROUTE1.TST. This file may then be accessed by name with the ROUTE command (RTEXT=rtext-id) from within a JDL (or with a DJDE). For this example the RTEXT parameter would be RTEXT=ROUTE1.

RFO RM, if needed, must be specified along with the reference to the RTEXT file (RFO RM is never cataloged with RTEXT). This allows the user to select different RFO RM s in different JDEs with the same RTEXT information.

**ROUTE command parameters**

The following provides the syntax of ROUTE command parameters and defines the parameter options.

**RFORM**

**Syntax**

```
ROUTE RFORM = {form-id | NONE}
```

**Function**

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.
**PRINT FORMAT COMMANDS**

**Option(s)**

**form-id** - Is the name of a file cataloged in the FRM directory. It is created by compiling a file of forms source commands, called a JSL, with the FDL compiler. (Refer to the 4635 LPS Forms Creation Guide.) The RFORM parameter is not allowed in a ROUTE command which is being defined as a cataloged file but is specified in the ROUTE command in the JSL invoking the cataloged RTEXT file. Refer to the “Cataloged RTEXT files” section in this chapter for further details.

**NONE** - Specifies that no form is to be printed.

**Default**

Default is **NONE**

**RTEXT**

**Syntax**

ROUTE RTEXT=**options**

ROUTE RTEXT= \( \{ sc, \left[ passnum \mid ALL \right], \left[ \left[ line, \left[ col, \left[ \text{fontindex} \right] \right] \right] \right] \} \)

**Option(s)**

**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).

**passnum** - May be either an integer number or the keyword **ALL**. An integer number specifies the pass (copy ply) to which the text applies.

**ALL** - Specifies that the text applies to all copies. If there is more than one RTEXT parameter in a DJDE packet, passnum must be in increasing numeric order. The default is passnum = 1.

**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** - 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.

**NONE** - Specifies that no text is to be printed on a separate page.

**rtext-id** - Is used to reference a cataloged file of RTEXT parameters previously compiled by PDL.

**Default**

Default is **NONE**
Considerations

- The font index is associated only with a particular string. To print an entire multiline RTEXT page in the same font, the font index must be given with each string. Strings without a font index are printed with the first alphanumeric font (specified in the font parameter of the PDE command).

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

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

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

Example—ROUTE command usage

ROUTE RTEXT=('USER 1',1,33,64),RTEXT=('USER 2',2,33,64);

The text USER 1 and USER 2 is to be printed in the center of a page preceding the respective copies of a two-copy report. The page is 132 columns by 66 lines.

STOCKSET command

The STOCKSET command defines a set of stocks used in a report. This allows output to verify the exclusivity of all stocks in terms of trays currently assigned to them. For your LPS, a stock-name is represented by a cluster. For more information refer to the feeder tray chapter in your Xerox 4635 LPS Operator Guide.

Stocksets are also the means of associating stock references with stock names.

An identifier (of the type ac) is specified to name the STOCKSET command for future reference by the STOCKS parameter of an OUTPUT command.

The parameters available for the STOCKSET command are summarized below, and a detailed description of these parameters is provided in table 6-14.
Table 6-14. Summary of ac:STOCKSET command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Offline</th>
<th>Online</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>Lists stock names and association of stock references with stock names</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>INIFEED</td>
<td>Specifies which stock to use in the absence of any parameters of the FEED parameter</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SYSPAGE</td>
<td>Specifies which stock to use for system-generated messages</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

STOCKSET command parameters

The following provides STOCKSET command parameter syntax and defines the parameter options.

ASSIGN

Syntax

```
ac:STOCKSET ASSIGN={stock-descriptor| (stock-descriptor1[,stock-descriptor2][,...])}
```

Function

Associates stocks with the identified stockset, where:

**Option(s)**

```
stock-descriptor = 
( stock-name
  (stock-name,stock-reference) )
```

Default

No Default

INIFEED

Syntax

```
ac:STOCKSET INIFEED = 
( stock-name
  stock-reference )
```

Function

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

Default

Default is first stock-name
SYSPAGE

Syntax  
\[
\text{ac:STOCKSET SYSPAGE=}
\begin{cases}
\text{stock-name} \\
\text{stock-reference} \\
\text{MAIN} \\
\text{AUX}
\end{cases}
\]

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

Default  
No Default

Point to note—STOCKSET command

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

For example, assume that you try to print this page on 8.5 by 11-inch paper, subject to the cluster representing the SYSPAGE stock, or the operator key in. If the paper size is not available, the system attempts to print the page on a different paper size chosen from the same set of trays as would normally be used. This occurs without any warning, caution, or cycle down. If a larger paper size is available, it is used. If a larger paper size is not available, a smaller paper size is used. This is the only time a page may be printed on a paper size smaller than what was specified to format the page.

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

Example—STOCKSET command usage

The STOCKSET command is similar to PDEs and CM Es, because it can be compiled separately. For each STOCKSET encountered before an initial JDL or SYSTEM command, PDL creates a file type STK, which contains information. The .STK file can then be referenced in multiple JSL source files.

```plaintext
BILL S:STOCKSET
ASSIGN   = ('F1302', COVER),
ASSIGN   = ('F1415', (BODY, SUMMARY)),
ASSIGN   = (('F6204', LATE), ('F9999', DISCON)),
INIFEED  = COVER, /*INIFEED = 'F1302' WORKS AS WELL*/
SYSPAGE  = COVER;
```
In the example above, a stockset called BILLS is created and consists of four stocks: F1302, F1415, F6204, and F9999. Stock references are created for these four stocks; COVER for F1302, BODY and SUMMARY for F1415, LATE for F6204, and DISCONNECT for F9999. In the absence of a specific parameter of the FEED parameter at the start of the report, INIFEED directs output to feed from the stock referred to by COVER. A billing application makes use of this stockset by coding an OUTPUT STOCKS=BILLS command in its JDE, and through successive DJDE FEED records, directs feeding the cover, body and summary, late notice, and disconnect pages from the stock referenced by COVER, BODY, SUMMARY, LATE, and DISCONNECT, respectively. Alternatively, the DJDE FEED record could refer directly to stocks by name, such as FEED='F1302'. Note that the use of stock references is preferred over stock names.

VFU command

The VFU command is used to assign output line numbers to printer carriage control channels. These line-to-channel assignments perform the same function as the printer carriage control tape on a conventional line printer. The VFU command is also used to assign line numbers to the top of form and the bottom of form. Top of form indicates the number of lines from the top (as defined by the PDE BEGIN values) of an output page to the first print line on the page. TOF defines the first line on which printing appears. Bottom of form indicates the number of lines from the top of an output page to the last print line on the page. BOF defines the last line on which printing appears.

Top and bottom of form are used for prejob page alignment and for page overflow processing. For all PCCTYPEs except ANSI and user-defined PCCs, the prejob page alignment is to top of form in the expectation that the first carriage control parameter of the job will be print and space one line, or something similar. Selection of ANSI causes alignment to bottom of form to handle the skip-to-channel-1-and-print parameter, which usually begins a job of that carriage control type. User-defined PCCs may set alignment at either TOF or BOF.

Page overflow occurs when spacing to the next line causes the bottom-of-form line number to be exceeded. Page transition occurs, and line spacing is continued from the top-of-form line number. Honeywell 2000 carriage control and Xerox carriage control are exceptions to this processing (refer to the LPS Tape Formats Manual).

You must specify an identifier of the type ac when defining the VFU table and reference it in the VFU parameter of the LINE command. The VFU command must precede the LINE command. A command identifier of the type ac may consist of one-to-six alphanumeric characters (A-Z and 0-9). One of the characters must be alpha.

The parameters available with the VFU command are summarized in table 6-15. A detailed description of these parameters is provided in table 6-16.
Table 6-15. **Summary of ac:VFU command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>Assign line numbers to output channels</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>BOF</td>
<td>Bottom-of-form line number assignment</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TOF</td>
<td>Top-of-form line number assignment</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

**VFU command parameters**

The following provides the syntax of VFU command parameters and defines the parameter options.

**ASSIGN**

Syntax

\[
\text{ac:VFU ASSIGN} = \left\{ \text{(channo,lineno)} \right\} \left\{ \text{(channo, (lineno[,lineno][,...])}) \right\}
\]

Function

The set of ASSIGN left parts specifies the output line-to-channel assignments.

Option(s)

- channo - The number of the channel being assigned. It is an integer in the range 0 to 15. The user may end the VFU command with a semicolon and start another VFU command without an id field to continue specification for the same channel or a different channel.
- lineno - The number of the output print line being assigned to a particular channel. It is an integer in the range TOF to BOF. There are no default assignments for any channel, including channels 1, 9, and 12.

Default

No Default

**BOF**

Syntax

\[
\text{ac:VFU BOF} = \text{value}
\]

Option(s)

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

Default

Default is 66 (VFU command)
TO F

Syntax

ac:VFU TO F= value

Option(s) value - Specifies the number of lines from the top of the output page to the first print line on the page (top of form). The top-of-form specification is independent of channel assignments. It should be greater than or equal to 1 and less than or equal to the smallest line number assigned to a channel.

Default Default is 1

Points to note—VFU command

- Any channel operation encountered during printing for an unassigned channel causes a print-and-space-1-line operation. This operation occurs even though some vendor formats normally specify the default as a space-1-line-and-print operation.

- Multiple line numbers may be assigned to the same channel number. This simulates the vertical tabbing feature of an impact line printer where a skip-to-channel parameter causes transition to the next punched hole in the specified channel of the paper tape. This tape, which controls the printer, facilitates spacing a fixed number of lines down the print page. There may be multiple punches in any vertical format channel on the impact printer’s tape.

- A skip-to-channel parameter in the LPS causes selection of the next line number in the ASSIGN list (for that channel) larger than the current line number. Page transition and alignment to the first line number in the list occurs if no line number is larger than the current line.

Example—VFU command usage

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),TO F=5,B0 F=55;

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

If a skip-to-channel-2-and-print parameter is issued when the current line number is 11, the next line to be printed would be line 15 of the current page. Lines 10, 15, 20, and so forth, are also assigned to channel 2, but since the current line number is 11, the next consecutive line number assigned to channel 2 (greater than 11) is line 15.

If a skip-to-channel-12-and-print parameter is issued, the next line to be printed is line 55 of the current page.
Overview

The logical processing (sometimes referred to as “special processing”) commands enable the user to specify logical functions performed on either a record, a set of records, or on a block basis. The following table summarizes the functions that may be performed and the corresponding PDL commands.

Table 7-1. Summary of commands associated with logical processing

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANNER</td>
<td>Banner page detection test (online only)</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>BDELETE</td>
<td>Block deletion test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BSELECT</td>
<td>Block selection test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CRITERIA</td>
<td>Logical processing test specifications</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RAUX</td>
<td>Select page from auxiliary tray test</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RBAR</td>
<td>Detects the presence of bar code information</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RDELETE</td>
<td>Record deletion test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>RFEED</td>
<td>Changes cluster names on page-by-page basis without using DJDEs</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>ROFFSET</td>
<td>Page offset test</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RPAGE</td>
<td>Logical page reposition test</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>RRESUME</td>
<td>Resume printing test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>RSELECT</td>
<td>Record selection test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>RSTACK</td>
<td>End of report test</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>RSUSPEND</td>
<td>Suspend printing test</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>TABLE</td>
<td>Table of constants for logical processing</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</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 text-exp portion of the command defines a test to be performed on either one or two specified fields and their associated constants for a true or false value. The fields in the record or block are compared with their associated set of constants using either an equal (EQ) or a not equal (NE) operator. The basic element used to describe a test for a logical function is the CRITERIA command. Refer to the CRITERIA section in this chapter for more detailed information.

Logical processing commands with TEST parameters

The available logical processing commands with TEST parameters are listed below and are described separately in the following sections. Each individual command can be specified only once per job descriptor entry. If any one command is specified more than once, the last occurrence is used without notification of any error.

Table 7-2. Logical processing commands with TEST

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANNER</td>
<td>Banner page detection</td>
</tr>
<tr>
<td>BDDELETE</td>
<td>Block selection</td>
</tr>
<tr>
<td>BSELECT</td>
<td>Block selection</td>
</tr>
<tr>
<td>RBAR</td>
<td>Bar code detection</td>
</tr>
<tr>
<td>RDDELETE</td>
<td>Record deletion</td>
</tr>
<tr>
<td>RFEED</td>
<td>Change cluster names</td>
</tr>
<tr>
<td>ROFFSET</td>
<td>Page offsetting</td>
</tr>
<tr>
<td>RSELECT</td>
<td>Record selection</td>
</tr>
<tr>
<td>RSEND</td>
<td>Printing suspension</td>
</tr>
<tr>
<td>RRESUME</td>
<td>Printing resumption</td>
</tr>
<tr>
<td>RSTACK</td>
<td>End of report</td>
</tr>
<tr>
<td>RAUX</td>
<td>Auxiliary tray page selection</td>
</tr>
<tr>
<td>RPAGE</td>
<td>Logical page repositioning</td>
</tr>
</tbody>
</table>

All of the above commands except BANNER are available for offline processing. BANNER, RBAR, RAUX, RPAGE, ROFFSET, and RSTACK are available for online.

CRITERIA command

The basic element used to describe a test for a logical function is the CRITERIA command. Each CRITERIA command describes a field in either a record or block and the specific test to be performed. The CRITERIA command requires an identifier that can be specified in any of the logical processing commands, as a TEST parameter. The definition of the TEST parameters for logical
processing commands is described in the “Test expressions” section.

Specific TEST parameters are described in the individual logical processing command sections.

A CRITERIA command may use any or all of its three parameters to specify the conditions which must be met in order for a given operation to be completed. A JSL may contain any number of CRITERIA commands which you can specify for a particular print job by calling out the appropriate CRITERIA identifier. For example, the following criteria are specified at the system level:

T1: TABLE MASK='?',CONSTANT='HE?DE? PAGE';
C1: CRITERIA CONSTANT=(1,11,EQ,T1),LINENUM=(1,10);
T2: TABLE CONSTANT='TRAILER PAGE';
C2: CRITERIA CONSTANT=(0,12,EQ,T2),LINENUM=(1,10);

Criteria C1 and C2 are then specified at the job level:

HDRP: JOB;
   BANNER TEST=C1,HCOUNT=2,TCOUNT=0;
TRLP: JOB;
   BANNER TEST=C2,HCOUNT=0,TCOUNT=3;
BOTH: JOB;
   BANNER TEST=(C1 OR C2),HCOUNT=2,TCOUNT=3;

Refer to “Test expressions” section for additional information.

CHANGE

**Syntax**

```
CRITERIA CHANGE= (offset, length, NE, LAST)
```

**Function**

Defines test specifications for a logical processing function with change mode criteria.

**Option(s)**

- **offset** - The offset in bytes (relative to zero) from the start of the user's portion of the record to the control field within the record or from the start of the physical tape block to the control field within the block.
- **length** - The length in bytes of the control field. Its range is 1 to 255.
- **NE** - Indicates not equal to.
- **LAST** - Indicates that the control field of the current record (or block) is being compared to the control field of the previous (last encountered) record (or block). If a control field of the current record (or block) is less than the specified length, the comparison is not done and the test fails. (For online and HIP jobs, where trailing blanks are truncated by the host, control fields which are less than the specified length may be padded with blanks so that a comparison can be done. If you are using a value less than the length specified, or if the number is the same but less than zero, the values are not internally the same and thus, the test is met.)

**Default**

No default
CONSTANT

Syntax \[ \text{CRITERIA CONSTANT=} (\text{offset}, \begin{cases} \text{EQ} \\ \text{NE} \end{cases}, \text{tab-id}) \]

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

Option(s) \textbf{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's portion of the record to the field in the record being compared.

\textbf{length} - The length in bytes of the test field. Its range is 1 to 255.

\textbf{EQ} and \textbf{NE} - Keywords that specify the operation of equal to or not equal to.

\textbf{tab-id} - The identifier of a TABLE command.

Default No default

LINENUM

Syntax \[ \text{CRITERIA LINENUM=} (\text{init}, \text{count}) \]

Function Specifies the range of consecutive line numbers for which the CRITERIA command is fully evaluated. If not specified, the default range is all lines.

Option(s) \textbf{init} - An integer specifying the number, on each page, of the beginning line for which the CRITERIA command is evaluated.

\textbf{count} - An integer specifying the consecutive number of lines (starting with init) for which the CRITERIA command is evaluated.

Default Default is all lines

Test expressions

The activation of testing is done by coding the TEST parameter of any logical processing command described in this chapter. Either one or two CRITERIA commands can be specified.

The following is a definition of the test expression for a logical processing command. The test definition is referred to in the syntax of each of the logical processing commands that follow:

\[
\begin{align*}
\text{TEST} &= \text{cri-id}; \\
\text{TEST} &= (\text{cri-id}_1, \text{AND}, \text{cri-id}_2); \\
\text{TEST} &= (\text{cri-id}_1, \text{OR}, \text{cri-id}_2);
\end{align*}
\]
The cri-id1 and cri-id2 are identifiers for either the change mode or constant mode CRITERIA commands discussed in this section. If only cri-id is present, the test is satisfied if the criteria in cri-id is satisfied.

If two CRITERIA commands are specified for a test, the testing of two is linked by either an AND or an OR logical operator. If AND is coded, both CRITERIA commands must be true for the same data record in order for the TEST to be successful. If OR is coded, then either CRITERIA command is satisfied for a data record, the TEST is successful. The second and third CRITERIA commands are optional; but if either is specified, the other must also be present. If cri-id1 and cri-id2 are both present and the keyword AND is coded, the test is true only if criteria in both cri-id1 and cri-id2 are satisfied. If the keyword OR is coded, the test is true if the criteria in either cri-id1 or cri-id2 are true. If the test is satisfied, the logical processing function is performed.

For example, the following CRITERIA command describes a test for a subfield equal to a specific constant table.

cri-id: CRITERIA CONSTANT = (offset,length,EQ,tab-id);

The CONSTANT parameter specifies that the content of a record is to be compared to the values in a table command with identifier tab-id (refer to the “TABLE command” section later in this chapter). The subfield to be compared is located offset bytes from the start of the record or block, with length in bytes. When the subfield in the record or block matches a constant in the table, the CRITERIA command is true and the processing called for in the parameter is initially completed or acted upon. The particular logical processing function is performed when the test expression is true as specified by the parameter.

**Specifying one CRITERIA command**

To complete the description of the entire test for a logical function, the TEST parameter requires a parameter that specifies either one or two CRITERIA commands. If only one test is to be performed to determine the value of a particular logical processing function, the form of the TEST parameter is as follows:

TEST = cri-id;

The cri-id is the identifier for the particular CRITERIA command. Parentheses in this format are optional.

**Specifying two CRITERIA commands**

If two CRITERIA commands are needed to determine the true or false value for a logical processing function, they may be linked by either an AND or an OR. The formats of these TEST parameters may be as follows:

```
TEST = (cri,AND,cri-id-2);
```

```
or
TEST = (cri,OR,cri-id-2);
```

The cri-id and cri-id-2 are the identifiers for two CRITERIA commands. The parentheses in this format are required.

For example, the following three commands specify that a record
should be selected (RSELECT parameter) if a particular field is equal to a given string constant (sc):

\[
T1: \text{TABLE CONSTANT}=(sc);
C1: \text{CRITERIA CONSTANT}=(\text{offset, length, EQ}, T1);
\text{RSELECT TEST}=(C1);
\]

There are two formats for the CRITERIA commands: constant mode and change mode.

Each CRITERIA command may be either constant mode or change mode, but not both.

**Constant mode**

In the constant mode, the user must specify the location, length, and contents of a fixed field within a user's record or block. Each user's record or block is examined at the specified location to determine if the constant is present (the identifier tab-id defines the table containing the constant). If present, the CRITERIA command is true; if not, the command is false. A constant mode CRITERIA command can be coded as follows:

\[
\text{cri-id: CRITERIA CONSTANT}=(\text{offset, length, EQ|NE, tab-id}),
\text{LINENUM}=(\text{init, count});
\]

The tab-id is the name (or identifier) of a value TABLE.

**Change mode**

In change mode, the user must specify the length and location of a control field in each record or block. When the content of the control field of one record (or block) differs from the content of the control field of the previous record (or block), the CRITERIA command is true.

A change mode CRITERIA command can be coded as follows:

\[
\text{cri-id: CRITERIA CHANGE}=(\text{offset, length, NE, LAST}),
\text{LINENUM}=(\text{init, count});
\]

**LINENUM parameter**

The LINENUM parameter of a CRITERIA command specifies that only a subset of a report's data records are to be tested for the user-defined criteria. This subset is specified as a set of contiguous line numbers for a page. If the carriage control specification for a given record is such that the record is placed within the set of lines specified by LINENUM, the record is examined according to the CRITERIA command. Otherwise, it is not processed.

**Combining change and constant modes**

The CRITERIA tables may specify either change mode or constant mode functions; there are no restrictions on their usage or combination.

There are two special cases: when the record or block is too short to include the field being tested and when the line being
tested contains no record, that is, it has been skipped. If the test specifies a constant mode function, the CRITERIA fails.

**Note:** If the test specifies a change mode function, the CRITERIA fails because no change has occurred, but the value for LAST option of the CHANGE parameter is unchanged for comparison with the next record. (See additional information under the “Criteria parameter functions” section.)

A compiler error occurs if you attempt an AND test on two different criteria which appear on two non-overlapping line ranges. For example, a logic problem occurs with ROFFSET where line 3 contains Criteria 1 and Line 2 contains Criteria 2, that is, Criteria 1 and Criteria 2 are not equal to LAST.

**Points to note—CRITERIA command**

- When change mode CRITERIA commands are evaluated, a string from the current line is compared with a string saved from the corresponding part of a previous line.
  - If the comparison is not equal, the string from the current line becomes the saved comparison string for subsequent lines and the criteria is considered true.
  - If two records are overprinted on the same line but with different data in the field defined by the CRITERIA command, the string from the last overprint record becomes the saved comparison string and the CHANGE CRITERIA is satisfied.
  - If the LINENUM parameter is used, lines within the specified range are processed normally, and lines outside the range are not evaluated. Thus, change mode comparison strings are not saved from lines outside of a LINENUM range.

- No print records (carriage control specifies no printing but only skipping or spacing) are evaluated for logical processing in the following manner:
  - If change mode is specified, no print records are evaluated.
  - If constant mode is specified, no print records are evaluated.

**Examples—CRITERIA command usage**

**Example 1**

The following are examples of CRITERIA commands including the TABLE commands they reference. The identifiers C1 and C2 are referenced by a logical processing TEST parameter.

T1: TABLE CONSTANT=('$$//');  
C1: CRITERIA CONSTANT=(4,4,EQ,T1);  
T2: TABLE CONSTANT=('//JOB ','//EXEC');  
C2: CRITERIA CONSTANT=(1,6,EQ,T2);

**Example 2**

T1: TABLE CONSTANT=('EFGH');  
C1: CRITERIA CONSTANT=(104,4,NE,T1),LINENUM=(10,6);  
RDELETE TEST=(C1);
The LINENUM specification (10,6) means that lines 10 through 15 on each page are being examined for the RDELETE test. Any line not in that range is not examined.

String comparison concepts

String comparisons

String comparisons for logical processing are specified using the CRITERIA and TABLE commands. These comparisons test for one of the following conditions:

- Absolute equality—tested for by either a change mode CRITERIA command or by a constant mode CRITERIA command which references a TABLE command that does not have a MASK parameter coded (refer to processing sequence 1 in figure 7-1).

- Equality under a mask—tested for when the referenced TABLE command has a MASK parameter coded (refer to processing sequence 2 in figure 7-1).

Comparing strings for equality under a mask means that the user has specified, for each character position of the input data string, that the test performed is one of the following:

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

Character types

An attribute called type may be associated with any character(s) in a character set. The possible types are identified by the integers 1 through 7. Any single character in a set, for example \\, will either have a type (type 1, type 2, type 3, and so forth.) or not. If it has no type, it is referred to as untyped. It is possible for any character to be untyped, have one type, or have multiple types, for example, be type 2 and type 3.

The defining of types, that is, the associating of type numbers with any group of characters in a character set, is described in the “Character type assignments—TCODE command” section of the “Specifying input parameters” chapter.

For every standard character set, there is a set of standard default type assignments which, for most applications, is sufficient without modification. These assignments define the numeric characters 0-9 to be type 1 characters and the uppercase and lowercase alphabetic characters A-Z and a-z to be type 2 characters. Use of these default type assignments can be invoked by the TCODE parameter of the VOLUME command.
Masked comparisons using default type assignments

Performing masked comparisons using unmodified standard default type assignments requires coding the following:

- **TCODE command.** Used either to modify a set of standard default type assignments or to define a completely new set (refer to processing sequence 3 in figure 7-1). Refer to the “Character type assignments” section in chapter 4, “Specifying input parameters,” for detailed information on the TCODE command.

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

- **MASK and CONSTANT parameters of one or more occurrences of the TABLE command.** Together, these parameters define exactly how the comparisons for that TABLE command are made using the set of type assignments specified by the TCODE parameter of the VOLUME command.

Masked comparisons using non-default type assignments

Performing masked comparisons using either a nonstandard character set or a standard set for which the default type assignments are not suitable requires coding the following:

- **TCODE command.** This command is used either to modify a set of standard default type assignments or to define a completely new set (refer to processing sequence 3 in figure 8-1). Refer to the “Character type assignments” section in chapter 4, “Specifying input parameters,” for detailed information on the TCODE command.

- **TCODE parameter of the VOLUME command.** This parameter is used to specify the set of assignments defined by the TCODE command as the one used for masked comparison.

- **The MASK and CONSTANT parameters of one or more occurrences of the TABLE command.** The use of these parameters is the same as for comparisons using unmodified defaults.
The stacked reports feature enables the online and offline user to define a series of reports in a single file. This is accomplished by specifying an end-of-report condition in the coded logical processing commands RSTACK (online and offline) and BANNER (online use only). End-of-report is that point in processing a report when all of the pages of a copy of a report have been formatted to disk and processing has begun on the next report.

Reports are stacked in a file if more than one report is included in a single file and separated from each other logically but not physically (with tapemarks, operating system labels, and so forth). In processing stacked reports, the system checks each record for the logical end-of-report specification as defined by the TEST parameter of the RSTACK and BANNER commands. When BANNER is coded, the user specifies the number of consecutive banner pages which must satisfy the test criteria before end-of-report occurs.

Report separation based on banner page detection is performed with the BANNER command. This includes a test expression to detect a banner page and a count of the header or trailer pages used to separate reports. Table 7-3 lists the function and use of each BANNER parameter. The syntax and format of the BANNER command are described in table 7-4.
Table 7-3.  Summary of BANNER command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOUNT</td>
<td>The total number of consecutive header banner pages.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>HJOBNO</td>
<td>Subfield of the first record to satisfy banner selection criteria be displayed as a CUST JOB number.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>HRPTNA</td>
<td>Subfield of the first record to satisfy banner selection criteria to be displayed as a REPORT NAME.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TCOUNT</td>
<td>Total number of consecutive trailer banner pages needed for banner page detection.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TEST</td>
<td>Test expression for banner page detection.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>TYPE</td>
<td>Use of selected data pages as banner pages when reports do not contain this information.</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

### HCOUNT

**Syntax**

```
BANNER HCOUNT = value
```

**Option(s)**

`value` - Specifies the total number of consecutive header banner pages which must be detected to satisfy the test expression. 0 is the default.

**Default**

Default is 0

### HJOBNO

**Syntax**

```
BANNER HJOBNO = \{(offset, length) \}
```

**Function**

Specifies that for each report a subfield of the first record satisfying the banner selection criteria is to be displayed as CUST JOB # on the LPS system controller in response to the JOBS command or STATUS function key.

**Option(s)**

- `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 (0 to 6).
- `NONE` - Specifies that no HOST job number is to be selected.

**Default**

Default is NONE
HRPTNA

Syntax
BANNER HRPTNA= \{(offset, length)\} \{NONE\}

Function
Specifies that for each report a subfield of the first record satisfying the banner selection criteria is to be displayed as the REPORT NAME on the LPS system controller in response to the JOBS command or STATUS function key. (For jobs with HCOUNT=0, that is, trailers only, the report name is associated with the report that precedes the trailer banner page.)

Option(s)
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 16).
NONE - Specifies no report name is to be selected.

Default
Default is NONE

TCOUNT

Syntax
BANNER TCOUNT= value

Option(s)
value - For online jobs, specifies the total number of consecutive trailer banner pages which must be detected to satisfy the test expression. 0 is the default.

Default
Default is 0

TEST

Syntax
BANNER TEST= test-exp

Function
For online jobs, defines the test expression for detection of a banner page for either change mode or constant mode criteria.

Option(s)
test-exp - The format and syntax of a test-exp are defined in the “Test expression definition” section. If a test-exp is satisfied, the page containing the record tested is considered a banner page.

Default
No default
TYPE

Syntax

BANNER TYPE= \{ DATA \; BANNER \}

Option(s)

DATA - For online jobs, use when reports do not contain user-specified banner pages. This parameter allows the user to treat selected data page(s) as banner pages for report separation. This type of banner page is printed with a form (if specified). In duplex mode, this type of banner page is printed as a duplex page with data on the back. If multiple PDE begins are specified and TYPE=DATA, only the first header page of the report is repositioned to the first logical page of a new physical sheet.

BANNER - Should be coded when reports contain user-specified banner pages. These banner pages are printed without forms. In duplex mode, the banner pages are printed with blank backs.

Default

Default is BANNER

Points to note—BANNER command

Table 7-4 shows the effect of the TCOUNT and HCOUNT parameters of the BANNER command under the conditions listed and the results of each.

Table 7-4. Summary of BANNER command parameter conditions and results

<table>
<thead>
<tr>
<th>Condition</th>
<th>Parameter</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS positions to TOF after end-of-report processing</td>
<td>If TCOUNT = 0</td>
<td>The page after the first header is positioned at TOF.</td>
</tr>
<tr>
<td></td>
<td>If TCOUNT not 0</td>
<td>The page after the last trailer is positioned at TOF.</td>
</tr>
<tr>
<td>Running online with header pages only as the offset criteria</td>
<td>TCOUNT = 0</td>
<td>Job parameters such as DJDEs and page numbering for a given report are applied to the first header page of the following report.</td>
</tr>
<tr>
<td>The system encounters a non-banner page</td>
<td>TCOUNT = 0</td>
<td>The next banner page is treated as the first header of a new report, even if HCOUNT is not satisfied.</td>
</tr>
<tr>
<td>The system encounters a non-banner page before HCOUNT is satisfied</td>
<td>HCOUNT not 0 and TCOUNT not 0</td>
<td>LPS assumes that the report has fewer header pages than were specified and begins looking for trailer pages.</td>
</tr>
</tbody>
</table>

Examples—BANNER command usage

Example 1

The following commands define the BANNER criteria for a job stream with two trailer pages and one header page. A character
string of 120 asterisks beginning in print position one of line 66 occurs on header and trailer pages.

\[
\text{T1: TABLE CONSTANT=(120)'*';}
\]

\[
\text{C1: CRITERIA CONSTANT=(0,120,EQ,T1), LINENUM=(66,1);} \\
\text{BANNER TEST=C1, HCOUNT=1, TCOUNT=2, HJOBNO=(121,6);}
\]

Example 2

If a job stream has two trailer pages with the character string DATE=mm/dd/yy appearing on line 10 or 11 and beginning in print position 20, the BANNER criteria could be coded as follows:

\[
\text{T2: TABLE CONSTANT=('DATE=::/:::'), MASK=(':');} \\
\text{C2: CRITERIA CONSTANT=(19,13,EQ,T2), LINENUM=(10,2);} \\
\text{BANNER TEST=C2, HCOUNT=0, TCOUNT=2;}
\]

The MASK parameter specifies a masking character. The CONSTANT parameter specifies both the constant data being compared and the character positions which contain variable data which are not being compared. In this example, positions 1 through 5 (DATE=) as well as positions 8 and 11 (the /’s) of the specified character string contain fixed data. Positions 6, 7, 9, 10, 12, and 13, which are identified by the mask character (:), contain variable data and are not compared. Header page count has been set to zero because criteria defining header pages have not been specified.

### BSELECT and BDELETE commands

Interspersed blocks within one offline report or file may be either selected for or deleted from printing by use of the BDELETE and BSELECT commands.

### TEST—BSELECT and BDELETE commands

These commands can also be used selectively to delete specialized blocks, for example, control blocks and unsupported labels, that are on the data tape but are not to be printed.

Syntax for the TEST parameter of BSELECT and BDELETE commands is as follows:

**TEST**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>TEST = test-exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option(s)</td>
<td>test-exp</td>
</tr>
<tr>
<td>Default</td>
<td>No default</td>
</tr>
</tbody>
</table>

- Defines test expression for selecting blocks for printing. The format and syntax of a test-exp are defined in the “Test expression definition” section. The block is selected (BSELECT) for printing or deleted (BDELETE) if a test-exp is satisfied.
Points to note—BSELECT and BDELETE commands

- It is important to note that block selection or block deletion is performed before the extraction of the records from the block. If a block is deleted from or not selected for printing, none of the records contained within that block are processed or are available for any other logical processing functions.
- If a block does not match the same format as the normal blocks, it can be deleted and thus not cause a processing error. For example, a control block in a fixed blocked file may cause a processing error unless it is first deleted.
- When defining TABLE CONSTANTS or CHANGE parameters, offsets to subfields of a block specify in bytes, relative to zero from the start of block to the beginning of subfield.

Examples—BSELECT and BDELETE commands usage

Example 1
The following commands illustrate the use of BSELECT to process interspersed reports (refer to the following figure) on a block basis.

T1: TABLE CONSTANT=('P');
C1: CRITERIA CONSTANT=(0,1,EQ,T1);
BSELECT TEST=(C1);

The contents of the first byte of each tape block (offset=0, length=1) is examined for the character constant 'P'. When a 'P' is detected, the entire block is selected (BSELECT) for printing. When the first byte of any block does not contain a 'P', that block is bypassed and not printed. In this example, only block 1 is printed.

Figure 7-2. Sample BSELECT and BDELETE command usage
Example 2  The BDELETE command to delete the blocks on the tape that have $$11 is shown below:

T1: TABLE CONSTANT=('$$11');
C1: CRITERIA CONSTANT=(4,4,EQ,T1);
BDELETE TEST=C1;

RAUX command

Feeding a single sheet of paper from the auxiliary tray may be controlled from within the input data stream. If a data record satisfying the RAUX test criteria is found, the page on which the record is found is printed on a sheet of paper picked from the auxiliary tray. For simplex printing, the next page is printed on a sheet of paper selected from the main tray, unless it also contains a record satisfying the RAUX test criterion. For duplex printing, the need to pick a sheet from the auxiliary tray (as the result of detecting the specified RAUX criterion on either the front or back side of a page) is ascertained on a page-pair basis before the first side is printed. The FEED MAIN operator command (refer to chapters 5 and 6 of this manual) should be used with this parameter to prevent the auxiliary tray from being automatically selected when the main tray is emptied.

This parameter can also be used in combination with the PDL parameter FEED of the OUTPUT command (refer to chapter 6) and the FEED DJDE command (refer to chapter 8).

The syntax for the RAUX TEST parameter is as follows:

**TEST**

**Syntax**

RAUX TEST = test-exp

**Function**

Specifies the test expression for selection of paper from the auxiliary tray.

**Option(s)**

test-exp - The format and syntax of a test-exp are defined in the “Test expression definition” section. If a test-exp is satisfied, a page is selected from the auxiliary tray.

**Default**

No default

**Points to note—RAUX command**

- If a record intended to satisfy the criteria for RAUX is suspended by RSUSPEND, that record is not checked for the RAUX criteria. Note that the RAUX criteria does not take effect when printing is resumed (offline only).
- Record selection/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—RAUX command usage**

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

The data record:

```
XXXXXXXXXXXXXXXXXX
```

when processed as part of the input data stream causes the page it is a part of to be printed on a sheet of paper fed from the auxiliary tray. The Xs are depicted in the above example only to illustrate the specified offset of 20 character positions.

**RBAR command**

**Syntax**

```
RBAR TEST = (CONDITION1[OR AND]CONDITION2)
```

**Function**

The RBAR logical processing command detects the presence of bar code information on a page.

**Option(s)**

- `test-exp` - If the test condition is met, the 4635 then processes the parameters specified in the BARCODE statement.

**RFEED command**

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

**TEST**

**Syntax**

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

**Option(s)**

- `test-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.

**Default**  
No default

### Points to note— RFEED command

- Satisfaction of the RFEED criteria will cause the current page to feed from the cluster defined in the RFEED command.
- If a criteria is met, the cluster name will remain in effect until the next RFEED criteria is met or a new report is processed.
- RFEED is not available as a DJDE.
- In a multiple RFEED criteria command, the order in which the criteria are specified makes a difference in the job application. Once a criterion is satisfied, the system ignores criteria checking for the rest of the page. The criteria checking resumes when a page transition occurs. Consider this example:

```plaintext
PAPERSTK: STOCKSET ASSIGN = ('RED', red-ref),  
          ASSIGN = ('BLUE', blue-ref),  
          ASSIGN = ('GREEN', green-ref);
T1: TABLE CONSTANT = ('ABC');  
T2: TABLE CONSTANT = ('DEF');  
T3: TABLE CONSTANT = ('GHI');  
C1: CRITERIA CONSTANT = (4,3,EQ,T1);  
C2: CRITERIA CONSTANT = (4,3,EQ,T2);  
C3: CRITERIA CONSTANT = (15,3,EQ,T3);  
RFEED TEST = ((C1,'RED'), (C2,'BLUE'), (C3,'GREEN'));
```

In this example, the input task checks C1, C2, and then C3. If C2 and C3 satisfy the specified criteria, the C2 cluster has priority over the C3 cluster due to its position in the JSL.

## RSELECT and RDDELETE commands

Interspersed records within one report or file may either be deleted from or selected for printing by use of the RDDELETE and RSELECT commands. These commands can also selectively delete specialized records, for example, control records and offset records that are on the data tape but are not to be printed.

### TEST

<table>
<thead>
<tr>
<th>Syntax</th>
<th>RSELECT TEST= test-exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Define test expression for selection (RSELECT) or deletion (RDDELETE) of records for printing.</td>
</tr>
<tr>
<td>Option(s)</td>
<td>test-exp - The format and syntax of a test-exp are defined in the “Test expression definition” section. If a test-exp is satisfied, the</td>
</tr>
</tbody>
</table>
record is selected for printing. If the RDELETE test-exp is satisfied, the record is deleted from the printed output.

Default: No default

Points to note—RSELECT and RDELETE commands

- When defining TABLE CONSTANTS or CHANGE parameters, offsets to the subfields of the records are specified in bytes, relative to zero from the start of the user’s portion of the record to the beginning of the subfield.
- Record selection or deletion is performed prior to RAUX and suspend or resume, offline. If a record satisfying either the RAUX suspend or resume test criteria was previously not selected for, or deleted from printing, neither the RAUX nor suspend nor resume functions.

Example—RDELETE command usage

The following commands illustrate the use of RDELETE to process interspersed reports on a record basis (refer to figure 7-3).

T1: TABLE CONSTANT=('EFGH');
C1: CRITERIA CONSTANT=(104,4,EQ,T1);
RDELETE TEST=C1;

If the contents of a deletion control field located 104 bytes from the start of the user portion of the record are equal to the constant 'EFGH', the record is not printed. In the following example, records 2 and 6 would not be printed.

Figure 7-3 Sample RDELETE command usage

ROFFSET command

The logical processing function ROFFSET provides the capability for online and offline users to initiate a page offset in the stacker tray under control of the input data, that is, data and DJDE records. These special user-controlled offsets can be used to simplify job distribution by creating separate stacks for each distribution entity. No other special processing occurs as a result...
of the ROFFSET test being satisfied; for instance, the report is not
terminated nor are multiple copies produced at the offset
juncture.

In duplex mode, ROFFSET forces the logical page on which the
criteria are satisfied to be the first logical page of a new sheet.
The page on which the offset occurs can be determined by the
following criteria:

• If the record which satisfies the ROFFSET test is printed on a
  page of the output, that page is the offset sheet.
• If the record which satisfies the ROFFSET test is not printed
  on a page of the output, that is, it is deleted by RDELETE but
  not BDELETE, the ROFFSET function is performed for the next
  printable record. If the next printable record causes a
  transition to the next page, the next page is offset.

ROFFSET also provides the control to force an offset on either all
copies of the report or only the first copy. In conjunction with
job offset control (OFFSET parameter of the OUTPUT command),
the user can exert extensive control over the offsetting function
so as to build tailored, job-controlled stacks of output in the
stacker tray of the LPS.

**PASSES**

**Syntax**

```
ROFFSET PASSES=
```

**Option(s)**

- **FIRST** - Specifies that the satisfied criteria causes an offset only
  on the first pass of a collated print run.
- **ALL** - Specifies that the satisfied criteria causes an offset on all
  passes of a collated print run.

**Default**

Default is ALL

**TEST**

**Syntax**

```
ROFFSET TEST= test-exp
```

**Function**

Defines test expression for offsetting pages to the stacker tray.

**Option(s)**

- **test-exp** - The format and syntax of a test-exp are defined in the
  “Test expression definition” section. If a test-exp is satisfied, the
  record causes an offset in the stacker tray. If a test-exp is
  satisfied in duplex mode, the logical page on which the record
  occurs is the first logical page of a new sheet.

**Default**

No default
Points to note—ROFFSET command

- The ROFFSET feature prints the record satisfying the test expression according to the normal job parameters.
- If ROFFSET is specified for an uncollated job, an offset occurs on only the first copy of the offset page.
- If a record satisfying the ROFFSET test criteria is not selected for or deleted from printing, the offset indication is maintained and used on the next record printed.
- If OFFSET=FIRST (or ALL) and the ROFFSET criteria is satisfied on the first page of a report, the normal offset from the preceding report is nullified.
- If segments are specified, the ROFFSET command is ignored.

Example—ROFFSET command usage

A file has multiple reports without any delimiters separating the reports (as illustrated below). Each page of the report has a page number as part of the heading. Each report causes renumbering of the pages starting with Page...1. With the ROFFSET command coded below, an offset occurs for all passes of the reports.

T1: TABLE CONSTANT=('PAGE...1');
C1: CRITERIA CONSTANT=(105,8,EQ,T1);
ROFFSET TEST=C1, PASSES=ALL;

Figure 7-4 Sample ROFFSET command usage

RPAGE command

The RPAGE command allows for the repositioning of the current (or next) logical page. Refer to figures 7-5 and 7-6 for a pictorial description of the effect of the RPAGE parameters. Table 7-5 summarizes RPAGE command parameters.

Table 7-5. Summary of RPAGE command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specifies</th>
<th>Offline</th>
<th>On-line</th>
<th>DJDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIDE</td>
<td>Side of physical page on which to reposition the logical page.</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TEST</td>
<td>Test expression for detecting an RPAGE record.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>WHEN</td>
<td>Portion of the current logical page to be positioned.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
**RPAGE command parameters**

The following provides RPAGE command parameter syntax and defines the parameter options.

### SIDE

**Syntax**

```
RPAGE SIDE= { side-opt, offset-opt }
```

**Function**

Defines the side of the physical sheet on which the repositioned logical page is to appear.

**Option(s)**

- **side-opt** - Available side-opt parameters 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.

**Default**

Default is (NUFRONT, NOFFSET)

### TEST

**Syntax**

```
RPAGE TEST= test-exp
```

**Function**

Defines the test expression for detection of an RPAGE record.

**Option(s)**

- **test-exp** - The format and syntax of a test-exp are defined in the “Test expression definition” section. The logical page on which this record normally prints is called the “current logical page.”

**Default**

No default
WHEN

Syntax: \[
\text{RPAGE WHEN = } \begin{cases} 
\text{BOTTOM} \\
\text{NOW} \\
\text{TOP}
\end{cases}
\]

Function: Defines the portion of the current logical page to be positioned to the first logical page of a new location.

Option(s):
- **BOTTOM**: The logical page following the current logical page is repositioned.
- **NOW**: The current record is repositioned. This record prints at TOF. Any spacing or skipping defined to occur before printing the current record is ignored.
- **TOP**: The current logical page is repositioned.

Default: Default is **TOP**

Points to note—RPAGE command

The following points apply to offline jobs only:

- If a record intended to satisfy the criteria for RPAGE is suspended by RSUSPEND, that record is not checked for the RPAGE criteria. Note that the RPAGE criteria takes effect when printing is resumed.
- Record selection/deletion is performed prior to RPAGE processing. If a record satisfying the RPAGE test criteria was previously not selected for or deleted from printing, the RPAGE does not function.
- Since DJDE processing occurs prior to RPAGE processing, DJDE records are not checked for satisfying the RPAGE criteria.
Figure 7-5  Effects of RPAG E command SIDE parameter

* indicates action caused by execution of WHEN option.
Figure 7-6 Effects of RPAGE command WHEN parameters

- **no RPAGE option**
  - **WHEN = TOP**
    - Original Physical Side:
      - Logical page "A"
    - New Physical Side:
      - Logical page "A"
      - **RPAGE RECORD**
      - Logical page "B"
      - **RPAGE RECORD**
      - Logical page "C"

- **WHEN = BOTTOM**
  - Original Physical Side:
    - Logical page "A"
    - Logical page "B"
    - **RPAGE RECORD**
  - New Physical Side:
    - Logical page "A"
    - Logical page "C"
    - **RPAGE RECORD**

- **WHEN = NOW**
  - Original Physical Side:
    - Logical page "A"
    - Logical page "B"
    - **RPAGE RECORD**
    - (blank)
  - New Physical Side:
    - Logical page "A"
    - Logical page "B"
    - (first part)
    - Logical page "C"
USING LOGICAL PROCESSING

RSTACK command

The stacked reports feature enables the online and offline user to define a series of reports in a single file. This is accomplished by specifying an end-of-report condition in the coded logical processing commands RSTACK (online and offline) and BANNER (online use only). End-of-report is that point in processing a report when all of the pages of a copy of a report have been formatted to disk and processing has begun on the next report.

Reports are stacked in a file if more than one report is included in a single file and separated from each other logically but not physically (with tapemarks, operating system labels, and so forth). In processing stacked reports, the system checks each record for the logical end-of-report specification as defined by the TEST parameter of the RSTACK and BANNER commands. When RSTACK is used for report separation, end-of-report occurs as soon as the test criteria is satisfied. However, when BANNER is coded, the user specifies the number of consecutive banner pages which must satisfy the test criteria before end-of-report occurs.

Report separation based on banner page detection is performed with the BANNER command. This includes a test expression to detect a banner page and a count of the header or trailer pages used to separate reports. The syntax and format of the BANNER command are described in table 7-3.

RSTACK delimiter modes

There are two modes of stacked reports: delimiter and non-delimiter. In the delimiter mode, the record satisfying the TEST criteria is not part of the report following the delimiter but simply serves to separate or delimit one report from another. In the non-delimiter mode, reports are stacked one on top of each other without any special records separating the reports. The two modes are specified by the DELIMITER parameter of the RSTACK command.

If DELIMITER=YES is coded, the user may actually separate each report with multiple successive records, each of which satisfies the TEST expression on the RSTACK command. In this case, all consecutive delimiters are treated as a delimiter packet. In delimiter mode, the user has the parameter to print the delimiter (or the delimiter packet) and to select the output destination of this delimiter page—BIN, TRAY, or BOTH. The parameter BOTH delivers the page to the sample print tray and the output tray. The delimiter page, when printed, is output as part of the subsequent report. All of the printing parameters selected in the JDL remain in effect during report delimiter printing except carriage control, which is ignored; the carriage control is replaced by a print and space 1 line control. If DJDE records exist within the delimiter packet or immediately following, they are included on the delimiter page.

In non-delimiter mode (DELIMITER=NO), a single record separates one report from the next. After satisfying the TEST criteria, this record is considered part of the subsequent report. In this mode the delimiter record cannot be printed to the tray, that is, PRINT parameter not available when DELIMITER=NO. It also does not appear on the operator’s display.
**Delimiter display**

Display of the delimiter (performed only when DELIMITER=YES) is dependent on the mode (Single/Multi) in which the job is run. Single or multiple report mode is a selectable parameter on the offline START command.

If the print job is run in single-report mode, the first delimiter of the next report is displayed. This occurs after an end-of-report is encountered. To begin the next single-report job, the operator must key in another START command. If the report mode is Multi or the system is online, the delimiter is not displayed.

**Delimiter on accounting page**

The user has the option to include part of the first record of a report on the accounting page. This command is selected by coding the ACCTINFO parameter, which is normally used to print part of the first delimiter.

**Status display**

The user has the parameter to display a subfield of the first record in a report in response to the JOBS command or STATUS function key. This command is selected by coding the HRPNTA or HJOBNO parameter.

Note that HJOBNO is not an RSTACK parameter.

**Online RSTACK usage**

When using RSTACK online, consider the following:

- Detection of RSTACK criteria within a not yet recognized banner page (RSTACK record occurs prior to BANNER criteria line) results in subsequent incorrect report separation.

- Detection of RSTACK criteria in a recognized but incomplete banner page (RSTACK record occurs after BANNER criteria line) is ignored.

- Detection of RSTACK immediately following report separation is ignored. This prevents null reports. If RSTACK is specified in a selected JDE or JDL, that is, a JDE or JDL invoked in a DJDE), the following restrictions to online systems apply:
  
  - If RSTACK is not specified in the JDE or JDL of the START command, the DELIMITER=NO parameter should be specified in the selected JDE or JDL. Specification of DELIMITER=YES is overridden.
  
  - If RSTACK is specified in the JDE or JDL of the START command, the DELIMITER parameter cannot be changed in a selected JDE or JDL.
  
  - If DELIMITER=YES is specified, a record that satisfies the TEST criteria of a selected JDE or JDL results in an end of report normally. However, it prints at the start of the subsequent report unless it also satisfies the TEST (or DJDE) criteria of the starting JDE or JDL.
• If RSTACK is specified in the JDE or JDL of the START command but not in the selected JDE or JDL, RSTACK processing is suspended until end of report is triggered by BANNER page detection.

• If ACCTINFO is specified in the JDE or JDL of the START command, the ACCTINFO parameter cannot be changed or deleted in a selected JDE or JDL (even if RSTACK is suspended).

• If ACCTINFO is not specified in the JDE or JDL of the START command, it can be invoked in a selected JDE or JDL. The first record after the selected JDE takes effect is then printed as the ACCTINFO field.

RSTACK command parameters

RSTACK command parameters and parameter options are defined below.

ACCTINFO

<table>
<thead>
<tr>
<th>Syntax</th>
<th>RSTACK ACCTINFO = (offset, length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>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.</td>
</tr>
<tr>
<td>Option(s)</td>
<td>offset - Specified in bytes (relative to zero) from the start of the user's portion of the record to the subfield within the record.</td>
</tr>
<tr>
<td></td>
<td>length - The number of bytes in the subfield (1 to 64).</td>
</tr>
<tr>
<td>Default</td>
<td>No default</td>
</tr>
</tbody>
</table>

DELIMITER

| Syntax | RSTACK DELIMITER= {YES | NO} |
|--------|--------------------------------|
| Option(s) | YES - Specifies that all consecutive records satisfying the TEST criteria separate one report from another but are not part of either report. |
| | NO - Specifies that this single record separates one report from another and is actually part of the subsequent report. |
| Default | Default is NO |
HRPTNA

Syntax

\[
\text{RSTACK HRPTNA} = \begin{cases} 
(\text{offset, length}) \\
\text{NONE} 
\end{cases}
\]

Function

Specifies that a subfield of the first record is being displayed as REPORT NAME on the LPS console in response to the JOBS command or STATUS function key. If DELIMITER = YES is also coded, the subfield is from the first delimiter record of the report. For DELIMITER = NO, the subfield is from the first data record.

Option(s)

- 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 16).
- NONE - Specifies no report name is selected.

Default

Default is NONE

PRINT

Syntax

\[
\text{RSTACK PRINT} = \{\text{BIN} | \text{BOTH} | \text{TRAY} | \text{NONE}\}
\]

Function

If DELIMITER=YES is coded, the user may specify if the report delimiters are to be printed, and if so, the output destination of the printed delimiters.

Option(s)

- BIN - Specifies that report delimiters are to be printed and the output delivered to the output stacker tray.
- BOTH - Specifies that report delimiters are to be printed, and the output delivered to both the sample print tray and output stacker tray.
- TRAY - Specifies that report delimiters are to be printed, and the output delivered to the sample print tray.
- NONE - Specifies that report delimiters are not to be printed. For DELIMITER=NO, no page is printed. If the report is being printed in duplex mode, report delimiters are printed on a single-sided duplex output page. The parameter BOTH in duplex is treated as BIN.

Default

Default is NONE
TEST

Syntax  \[ \text{RSTACK TEST=} \text{test-exp} \]

Option(s)  \text{test-exp} - Defines test expression for end-of-report conditions for either change mode or constant criteria. The format and syntax of a test-exp are defined in the "Test expression definition" section. If a test-exp is satisfied, the record specifies an end-of-report condition.

Default  No default

Points to note—RSTACK command

- If the TEST expression on the RSTACK command consists solely of a change mode CRITERIA command, DELIMITER=NO must be coded.
- An RSTACK command containing a TEST expression specifying a constant mode CRITERIA command and DELIMITER=NO can be used to detect a heading of a report as a report boundary.
- A record which is an RSTACK delimiter, that is, it satisfies the TEST expression, cannot be deleted from or not selected for printing by the RSELECT or RDELETE logical processing. If the record satisfies the RSTACK test criteria but is not a delimiter, it can be deleted from or not selected for printing but still causes report separation.

Example—RSTACK command usage

\begin{verbatim}
T1: TABLE CONSTANT=('XEROX');
C1: CRITERIA CONSTANT=(1,5,EQ,T1);
    RSTACK TEST=C1,DELIMITER=YES,ACCTINFO=(14,19),HRPTNA=(20,10);
\end{verbatim}

Reports are separated by a record with the characters XEROX in bytes 1 to 5 (relative to 0). Two reports are created as illustrated in the following figure. The delimiter record is not printed with the report.
RSUSPEND and RRESUME commands

The print suppression logical processing function permits the user to delete from printing groups of records that are distinguishable at the start and end, but whose intermediate records may not be unique or distinguishable. (Print suppression or resumption is invoked by the use of two separate commands—RSUSPEND and RRESUME.) The tests for each command are independent and must be described separately. Each of the commands can specify the full range of tests as described previously for the other logical processing commands.

When specifying either the RSUSPEND or RRESUME command, the user can also specify whether suspension or resumption of printing occurs on the current or next record. This is controlled by the BEGIN parameter. This additional control provides the necessary flexibility to cope with the variability of requirements for print suppression. The BEGIN parameter is independently specifiable in both RSUSPEND and RRESUME commands.

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

Printing is resumed when a record satisfying the TEST in the RRESUME command is encountered. If BEGIN=CURRENT is coded in the RRESUME command, the record satisfying the TEST is printed. If BEGIN=NEXT is coded, printing resumes with the next record.
RSUSPEND and RRESUME command parameter syntax and parameter option definitions are shown in table 7-15.

BEGIN

Syntax

**RSUSPEND BEGIN** = \{ **CURRENT** | **NEXT** \}

**Option(s)**

**CURRENT** - Specifies whether printing stops or resumes on the current or next record for offline jobs. For RSUSPEND, if CURRENT is coded, the record satisfying test-exp does not print.

(For RRESUME, if CURRENT is coded, the record is printed.)

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

(If NEXT is coded, the record satisfying test-exp is printed, and printing is suppressed beginning with the next record.)

**Default**

Default is NEXT

TEST

Syntax

**RSUSPEND TEST** = **test-exp**

**Function**

Defines test expression for the beginning of print suppression (RSUSPEND).

**Option(s)**

**test-exp** - The format and syntax of a test-exp are defined in the “Test expression definition” section. If the test-exp is satisfied, the record is used to suspend printing.

Defines test expression for the resumption of printing (RRESUME) following print suppression. If the test-exp is satisfied, the record is used to resume printing following print suppression.

**Default**

No default

Points to note—RSUSPEND and RRESUME commands

- **RSUSPEND**
  - Make sure that if an RSUSPEND command is coded, an RRESUME command must also be present for the job. A warning is issued by the PDL compiler if one command, but not both, is invoked for a job. However, the JDE is compiled as programmed.
  - The RSUSPEND command is intended to suspend records within a report.
• **RSTACK**  
  - A record satisfying the RSTACK test can still be found and terminates the report, even if the printing of records is suspended at the time. Also, DJDE records are not processed even if record printing has been suspended. Refer to chapter 8, “Specifying dynamic job descriptor entries (DJDEs),” for further information.
  - A record satisfying the RSTACK criteria is still detected and terminates the report and record suspension, even if the printing of records was suspended at the time.
  - If the criteria for RSTACK and RSUSPEND are satisfied on the same record, that record delimits the report, and the record is suspended.

• If a data record satisfying the test expression in the RSUSPEND is encountered, printing is suspended. If no record satisfying the test expression in the RRESUME command is encountered (or no RRESUME command is present for the job), there is no output generated for records that occur after the point of suspension.

• Record selection or deletion is performed before suspend/resume processing. If a record satisfying either the suspend or resume test criteria was not selected for or deleted from printing previously, it does not cause either the suspend or resume function.

• The records just before print suspension and after resumption should have compatible printer carriage control (PCC) characters. No additional carriage control characters are inserted by the system during the print suppression.

• Use of the LINEUM parameter in the CRITERIA command is not recommended when using RRESUME. Since the carriage control characters are not processed during the print suppression, the line number used by the system is that which existed when the suppression started. This could result in the test criteria being either unexpectedly satisfied or never satisfied as a function of the line number where the print suppression started.

• DJDE records are processed, not ignored, even when detected after an RSUSPEND record and before the corresponding RRESUME record.

---

**Example— RSUSPEND and RRESUME commands usage**

```
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;
```
The printing of the input data records in the following sample data is suspended (from record //JOB to //EXEC) by the preceding RSUSPEND command. Printing resumes after the //EXEC data record (RRESUME command above).

```
//JOB .FOR001A0
//OPTION CATAL
//ASSIGN SYSRLB,3340,TEMP    Records not to be printed
   INCLUDE IMPCBBM
//EXEC FCOBOL

.                      Data to be printed
.
.
.
```

**TABLE command**

The TABLE command is used to build a table of constants for use by the logical processing commands. Each constant included in a TABLE command is examined by the system to see if it is equal in value to the input data field specified in the CONSTANT parameter of the CRITERIA command. The TABLE command must precede its reference in a CRITERIA command.

The MASK parameter of the TABLE command permits character positions of a string to be ignored or tested for type attributes.

In an unmasked comparison (using only the CONSTANT parameter of the TABLE command) characters in the corresponding positions of the constant string and the input data string are tested for absolute equality one character at a time. In a masked comparison (using the MASK parameter), before the comparison is done for each character position, that position in the constant string is checked for a mask character as specified by the user. If one is found, the corresponding data character is tested only for the type indicated by the mask character.

Any valid character in the data character set, such as ASCII and EBCDIC, may be used in the MASK parameter as a mask character. Any character that is not explicitly tested for absolute equality by that TABLE command may be defined by the MASK parameter as a mask character. MASK parameter character definitions apply only to the TABLE command in which the definition occurs and, as in unmasked comparisons, it is possible to specify multiple strings in the CONSTANT parameter.

The following shows The CONSTANT and MASK parameter syntax.
CONSTANT

Syntax

\[ \text{ac:TABLE CONSTANT=} (sc_1[,sc_2][,...]) \]

Function
Specifies the content of one or more string constant(s).

Option(s)
- \( sc \) - String constants composed of literal characters and masking characters (as defined by the MASK parameter of this command). All string constants must be of the same length in bytes. The number of bytes for all constants in the table (after conversion of any hexadecimal or octal constants) is limited to 255 bytes.

Default
No default

MASK

Syntax

\[ \text{ac:TABLE MASK=} (\text{ignore-char},\text{charspeci}_1 [,\text{charspeci}_2][,...]) \]

Function
Defines mask characters used within string constants (specified by CONSTANT parameter) and indicates which character positions have special type testing and what that testing is.

Option(s)
- \( \text{ignore-char} \) - Specifies a character to appear in those positions within a string constant which are not to be compared.
- \( \text{charspeci} \) - (i = index number) specifies a character(s) to appear 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.

Default
No default

Examples—TABLE command usage

Example 1
The following are sample TABLE commands. The identifiers T1, T2, and T3 are each referenced by a CRITERIA command.

\begin{verbatim}
T1: TABLE CONSTANT='$$//';
T2: TABLE CONSTANT=('//JOB ','//EXEC');
T3: TABLE CONSTANT=(120)'*';
\end{verbatim}
Example 2

The following example illustrates the use of the MASK parameter to determine whether a particular field in a data record contains one of several different 6-character serial number formats. Possible formats are: xx999x, x999x9, and 99x999. For this example, x represents any alphabetic character and 9 represents any numeric character.

If the input code type is EBCDIC, the VOLUME command includes TCODE=EBCDIC. The TABLE command to find the format x999x9 within the input data stream would look like:

```
T1: TABLE MASK=('?','%','@'),CONSTANT=('@ % % % @ %');
```

The characters used in the MASK parameter occupy numbered positions, beginning with 0, separated by commas. These position numbers in the MASK parameter are used as type numbers. Therefore, the following mask character to type associations are made for the above TABLE command.

<table>
<thead>
<tr>
<th>Character</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>None</td>
<td>Make no comparison</td>
</tr>
<tr>
<td>%</td>
<td>1</td>
<td>Standard default, any numeric (0-9)</td>
</tr>
<tr>
<td>@</td>
<td>2</td>
<td>Standard default, any alphabetic (A-Z, a-z)</td>
</tr>
</tbody>
</table>

The data string characters in the positions corresponding to those occupied by \ in the CONSTANT parameter are tested for type 2 (in the example, alphabetic A-Z or a-z). Characters corresponding positionally with a % are tested for type 1 (numeric, 0-9). If the string specified in the CONSTANT parameter included a ?, the corresponding data character from the input stream is considered equal without any comparison being made, because it is an ignore type mask character.

Example 3

If the TABLE command in example 2 were changed to:

```
T2: TABLE MASK=('?','%','@'),CONSTANT=('A7%%@%');
```

Then, when the first two character positions of the CONSTANT parameter are checked for a mask character (as specified in the MASK parameter), none is found. For those two positions, exact character matches between the input data string characters and the CONSTANT parameter characters (in this case A7) are required. Therefore, in this example, only data strings that begin with A7 can possibly pass the entire test.
Example 4

The TABLE command

```
T2: TABLE MASK=('*',':','!'),CONSTANT=('DATE ::::/:: ID!!!');
```

sets up the following character-to-type associations:

<table>
<thead>
<tr>
<th>Mask position</th>
<th>Mask character</th>
<th>Character type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>*</td>
<td>None</td>
<td>Make no comparison</td>
</tr>
<tr>
<td>1</td>
<td>:</td>
<td>1</td>
<td>Any numeric (0-9)</td>
</tr>
<tr>
<td>2</td>
<td>!</td>
<td>2</td>
<td>Any alphabetic (A-Z, a-z)</td>
</tr>
</tbody>
</table>

A logical test using this TABLE command examines a field of an input data record to determine if the word DATE is present, if the / is where it is supposed to be, if the date itself is numeric (type 1), if the string ID is present, and if the first three characters after ID are alphabetics (type 2).

The basic element used to describe a test for a logical function is the CRITERIA. Each CRITERIA command describes a field in either a record or block and the specific test to be performed.
8. Specifying dynamic job descriptor entries (DJDEs)

Overview

Dynamic job descriptor entries (DJDEs) are parameters embedded within the input data stream and are used to modify the printing environment established by a job descriptor entry (JDE) “on the fly.” Dynamic job descriptor entry processing enables certain JDE parameters to be changed on a page-to-page or record-to-record basis. The IDEN command coded in a JDE notifies the system that DJDE records are included in the input data stream. This chapter contains information on these topics:

- Benefits of using DJDEs
- Page- and record-oriented DJDEs
- JDL specifications: the IDEN command
- DJDE format
- DJDE record specification
- Application of DJDEs
- DJDE operator information pages
- Job parameter modification restrictions
- Duplex DJDE page printing
- Effect on multiple logical pages
- Online DJDE restrictions
- DJDE processing optimization
- DJDE parameter definitions
- PDL parameters for graphics
- DJDE FILE parameters.

Table 8-1 lists the page- and record-oriented DJDEs. Table 8-2 contains a detailed description of each of the DJDE parameters.

Benefits of using DJDEs

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

- The printing system does not stop between reports nor does it require operator intervention. The operator starts up a job on the printing system and typically returns to it only when minor operational activity is required.
- Forms may be changed on a page-to-page basis.
- Many variations on VFU channel, margin, and top- and bottom-of-form assignments may be applied to reports as they are created instead of being stored in the printing system through JDEs. Thus, the number of JDEs required for job processing is reduced.
- Unusual processing requirements may be satisfied through the use of DJDEs.
• Varying numbers of copies can be generated automatically with routing or distribution notification sent to the operator.

• Nonprint data can be downloaded for storage on the LPS system disks through the print stream.

In the following sections of this chapter, note that some DJDE parameters show a default value. This is the default value used if the parameter is not specified in either a DJDE or the current JDE or JDL. If the parameter is not specified in a DJDE but is specified in the current JDE or JDL, the JDE or JDL value remains in effect.

Page- and record-oriented DJDEs

There are two types of DJDEs: page-oriented and record-oriented.

Page-oriented

Page-oriented DJDEs change specific pages within a report and can change these pages differently in different copies. Such parameters may be placed within the report itself and take effect at the next page boundary. They may also appear at report boundaries to effect changes on all pages of a report (or report- ply). The following DJDEs are page-oriented:

Table 8-1. Page-oriented DJDEs

<table>
<thead>
<tr>
<th>DJDE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td>Specifies the location of the starting print line of a logical page for graphics.</td>
</tr>
<tr>
<td>BSKIP</td>
<td>Specifies that barcode information is processed for pages that contain barcode information.</td>
</tr>
<tr>
<td>BSIDE</td>
<td>Specifies that the system will assign sequence and presence information to either odd, even, or both sides.</td>
</tr>
<tr>
<td>BSEQ</td>
<td>Specifies whether the sequence number should be reset, continued, or set at zero.</td>
</tr>
<tr>
<td>BFORM</td>
<td>Specifies that a form be printed on the back side of a printed page.</td>
</tr>
<tr>
<td>BTEXT</td>
<td>Specifies bar code audit parameters.</td>
</tr>
<tr>
<td>COLLATE</td>
<td>Specifies whether the pages of a job are to be collated (placed into sets) or uncollated (like pages placed together).</td>
</tr>
<tr>
<td>COPIES</td>
<td>Specifies the number of copies or sets of the pages to produce.</td>
</tr>
<tr>
<td>DEPT</td>
<td>Specifies the accumulation of accounting statics for reports on a name basis.</td>
</tr>
<tr>
<td>DESTINATION</td>
<td>Specifies the output destination of printed pages.</td>
</tr>
<tr>
<td>DUPLEX</td>
<td>Specifies whether both sides of a piece of paper are printed (duplex) or only to top side (simplex).</td>
</tr>
<tr>
<td>FEED</td>
<td>Controls the stock (type of paper) on which the page is printed.</td>
</tr>
<tr>
<td>FONTINDEX</td>
<td>Specifies a particular location in the input record where an index to the specified font is stored.</td>
</tr>
</tbody>
</table>
### Table 8-1. Page-oriented DJDEs (continued)

<table>
<thead>
<tr>
<th>DJDE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>FONTS</td>
<td>Specifies the fonts used in input data or variable (CME) data.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Specifies that a new page descriptor entry (PDE) is used for formatting control.</td>
</tr>
<tr>
<td>FORMS</td>
<td>Specifies the form to be merged on the printed pages.</td>
</tr>
<tr>
<td>INVERT</td>
<td>Specifies that an image is to be inverted.</td>
</tr>
<tr>
<td>ITEXT</td>
<td>Specifies a text message to be displayed to operators during the input process.</td>
</tr>
<tr>
<td>JDE</td>
<td>Specifies the JDE to be used within the selected JDL at the next page boundary.</td>
</tr>
<tr>
<td>JDL</td>
<td>Specifies the name of the JDL to be invoked at the next page boundary.</td>
</tr>
<tr>
<td>MAP</td>
<td>References a previously created font mapping file.</td>
</tr>
<tr>
<td>MARGIN</td>
<td>Specifies the left printing margin within each logical page.</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Specifies the CME to be used on the page to override an existing CME.</td>
</tr>
<tr>
<td>NUMBER</td>
<td>Specifies page numbering control.</td>
</tr>
<tr>
<td>OTEXT</td>
<td>Provides a text message to be displayed to the operator during job printing.</td>
</tr>
<tr>
<td>PMODE</td>
<td>Specifies the printing mode for each page (either landscape or portrait).</td>
</tr>
<tr>
<td>RFORM</td>
<td>Specifies whether a form is printed on all RTEXT pages.-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RTEXT</td>
<td>Specifies text to be printed on a separate page preceding a report.</td>
</tr>
<tr>
<td>SEFMAP</td>
<td>Specifies a font mapping table different than the one specified in the JDL.</td>
</tr>
<tr>
<td>SEPARATORS</td>
<td>Specifies that the next data page in a report be used to generate segment separator sheets.</td>
</tr>
<tr>
<td>SHIFT</td>
<td>Specifies a shift of image on the page for binding purposes.</td>
</tr>
<tr>
<td>SF1FUNCTION</td>
<td>Specifies sheet finisher function number 1 through the DFA channel C6.</td>
</tr>
<tr>
<td>SF2FUNCTION</td>
<td>Specifies sheet finisher function number 2 through the DFA channel C7.</td>
</tr>
</tbody>
</table>
Table 8-1. Page-oriented DJDEs (continued)

<table>
<thead>
<tr>
<th>DJDE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIDE</td>
<td>Specifies the repositioning of the new logical page to the first logical page of the given side of a physical sheet of paper</td>
</tr>
<tr>
<td>SNUMBER</td>
<td>Prints the segment number on generated segment separator sheets</td>
</tr>
<tr>
<td>SPLIT</td>
<td>Defines the conditions for starting and ending segments</td>
</tr>
<tr>
<td>SRECOVER</td>
<td>Specifies the recovery mode for sheets of paper in the paper path</td>
</tr>
<tr>
<td>STIMING</td>
<td>Specifies timing parameters. Performs same function as the Finishing Configuration Utility (FCG) but is ignored by the system</td>
</tr>
<tr>
<td>STOCKS</td>
<td>Redefines the parameters of a STOCKSET command (the types of stocks to use) for the page</td>
</tr>
<tr>
<td>TMODE</td>
<td>Specifies process direction measurement for determining default pitch modes</td>
</tr>
<tr>
<td>TRANS</td>
<td>Specifies whether transparencies are used in the print job.</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>Allows shifting in the x-direction for short-edge binding.</td>
</tr>
</tbody>
</table>

Record-oriented

Record-oriented DJDEs take effect immediately at the next record following the last record of the set of DJDE records of which they are a part; that is, after an END parameter, a compiled record-oriented DJDE takes effect immediately. The exception is the OVERPRINT DJDE, which takes effect at the next logical page boundary if FONTINDEX has been invoked in a JDE or DJDE. These DJDEs also may appear at report boundaries to change all pages in a report (or report-ply). The following DJDEs are record oriented:
Table 8-2. **Record-oriented DJDEs**

<table>
<thead>
<tr>
<th>DJDE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>Assigns a VFU channel to a page line number or set of line number</td>
</tr>
<tr>
<td>BOF</td>
<td>Specifies the bottom-of-form line number</td>
</tr>
<tr>
<td>C text</td>
<td>Allows comment text in the DJDE record</td>
</tr>
<tr>
<td>DATA</td>
<td>Specifies the location and length of printable data within an input record</td>
</tr>
<tr>
<td>END</td>
<td>Specifies the end of DJDE information</td>
</tr>
<tr>
<td>FILE</td>
<td>Enables files to load to the system disks while a print job is in progress</td>
</tr>
<tr>
<td>OVERPRINT</td>
<td>Provides instructions to the system when overprint lines (print lines whose carriage control specifies printing with no line spacing since the last printed line)</td>
</tr>
<tr>
<td>TOF</td>
<td>Specifies the top-of-form line number</td>
</tr>
</tbody>
</table>

The following graphic DJDEs are record-oriented:

<table>
<thead>
<tr>
<th>DJDE</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>Specifies the new imaging parameters for graphics</td>
</tr>
<tr>
<td>BATCH</td>
<td>Permits normal processing on online banner pages for batch mode jobs</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Cancels automatic reimaging (hold) of IMAGE or GRAPHICS parameters</td>
</tr>
<tr>
<td>GRAPHIC</td>
<td>Specifies that the DJDE is a graphic sentinel</td>
</tr>
<tr>
<td>IMAGE</td>
<td>In batch mode, defines the new imaging parameters for the graphic</td>
</tr>
<tr>
<td>SAVE</td>
<td>Specifies IMG files updated the current report which are not to be purged when the report's output processing is completed</td>
</tr>
</tbody>
</table>
### IDEN command

To invoke DJDE processing, an IDEN command must be coded in the JDL. This command notifies the system that DJDE records may be part of the input data stream. It also describes the search criteria for locating and identifying DJDE records. The DJDE record (or records) are interspersed among the data records in the input data stream. Each DJDE record contains an identification field (which matches the search criteria specified in the active JDE) and a series of parameters that describe the actual JDE changes to be applied to the report.

The IDEN command (coded in a user's JDE) notifies the system that a DJDE record (or records) may be part of the input data stream. The command also describes the characteristics of a DJDE record so the system can identify and locate any DJDE records. The IDEN command syntax and interpretation are described in table 8-3. An example of a coded IDEN command is

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

The PREFIX parameter defines the character string which must appear in the identification part of each DJDE record in the input data stream. When the system recognizes this character string (in the example, the EBCDIC string 'RTEST'), it examines the rest of the record for DJDE parameters. The OFFSET and SKIP parameters define the starting positions of the prefix string and DJDE parameters within the record.

Example:

```plaintext
IDEN PREFIX=E'DJDE', SKIP=7, OFFSET=2, OPRINFO=YES;
```

### OFFSET

<table>
<thead>
<tr>
<th>Syntax</th>
<th>OFFSET= value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>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 prefix string constant of the DJDE record. It may be a negative number.</td>
</tr>
<tr>
<td>Default</td>
<td>Default is 0</td>
</tr>
</tbody>
</table>

### OPRINFO

<table>
<thead>
<tr>
<th>Syntax</th>
<th>OPRINFO= YES, NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>YES specifies that the DJDE record is to be printed and delivered to the tray. NO specifies that the DJDE record will not be printed.</td>
</tr>
</tbody>
</table>

---

**SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)**

**8-6 XEROX 4635 LASER PRINTING SYSTEM PDL/DJDE REFERENCE**
All non-Interpress system-generated pages, such as accounting sheet, error summary, OPRINFO, recovery marker, and report separator are printed in the same xerographic mode as the previous page to prevent unnecessary xerographic mode switching (XMS). The exception to the rule is when the default ink is a primary color and the previous page's xerographic mode is BLACK. Accounting sheets always print using the system default 'BLACK' ink. The ink used to print all other non-Interpress system generated pages depends on the xerographic mode of the previous page and the default ink. Refer to table 9-4.

**Default**

Default is NO

---

### PREFIX

**Syntax**

```
PREFIX= sc
```

**Function**

An sc is the search criterion for recognizing DJDE records. It is a byte string of up to 255 characters represented as a hexadecimal, octal, BCD, ASCII, or EBCDIC character constant (as described in the “Right-part constants” section of chapter 1). Any records within the data stream which contain an identification field equal to the specified prefix string sc are recognized and processed as DJDE records.

**Default**

No default

---

### SKIP

**Syntax**

```
SKIP= value
```

**Function**

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.

**Default**

Default is 1

---

**Point to note—IDEN command**

When OPRINFO=YES is selected, the length of the DJDE record that prints on the sheet is whatever amount will fit on the width of the paper. With portrait orientation, using OPRINFO=YES may cause data or page exceeds page size errors if the DJDE record is too long.
DJDE record specification

DJDE records are created by the user as part of the data stream. The system looks for them only if a DJDE prefix has been specified by an IDEN command within the JDL used to process the job. DJDE information is contained in one or more records, each of which may be up to the maximum record length specified for the job input data. For each DJDE record, the prefix (the identification field) must begin in the same location in the record. DJDEs are terminated by an END parameter in the last DJDE record. All specified DJDE information is applied at the next record or page boundary after the END parameter is encountered. There may be multiple DJDE sequences in a job. Each set modifies only the specific parameters mentioned within the DJDE. DJDE parameters are defined in table 8-4.

The following should be considered when preparing DJDE records as part of the input data stream. Examples of DJDE records are illustrated in figures 8-1 and 8-2.

- The DJDE record may contain more than one parameter. Each parameter within a record must be separated from the next parameter by a comma.
- Lowercase alphabetic characters are not accepted in the DJDE parameter line. If they are used, a DJDE syntax error message is displayed.
- The end of a DJDE record parameter that is split and continued on the next DJDE record should be terminated with a ; (comma, semicolon).
- The prefix in the DJDE record may appear after the DJDE parameter(s) as long as it is consistently located in all DJDE records.
- DJDE records need not be consecutive since none of the parameters are applied until END; is encountered. However, it is recommended that DJDE records be consecutive, especially when used with delimiter records. To optimize input processing speed, DJDE parameters should be grouped into as few records as possible. A group of DJDE records terminated by an END parameter is called a packet.
- Only one page-oriented packet can be deferred to the next page. If a page-oriented DJDE packet is received after the start of page A so that it is deferred to page A+1, any separate DJDE packets following it on page A may cause unpredictable results and may possibly be printed as data.
- A page-oriented DJDE positioned prior to any print parameters on a page takes effect on the current page. A page-oriented DJDE positioned after the first print parameter on a page takes effect on the next page, unless end-of-report is encountered.
- For delimiter mode stacked report processing, DJDE parameters may be placed in the report body. They may also be placed within or after (but not before) the delimiter record packet.
For change mode stacked reports, the DJDE parameters may be placed within the body of the stacked report, where the DJDE records contain the same change field contents as the report to which the DJDE applies.

After a set of DJDEs is terminated by an END parameter, there must be one or more data records before the next DJDE packet. DJDE records directly following a previously terminated DJDE parameter set are ignored.

If the file containing the DJDE is variable-blocked, the program that blocks the file may strip off trailing blanks. Thus, if the comment record contains no actual comments, the blank following the C may be eliminated. If the C is the last character of the record, the parameter record is accepted. However, any character other than a blank following the C causes the record to be processed as a legitimate DJDE record and not as a comment. Refer to the C text parameter in table 8-2.

Print-and-skip carriage control parameters associated with online DJDE records are treated as skip immediate carriage control parameters. If the LPS is not at the channel specified, a skip occurs; but if it is already there, no skip occurs unless the last carriage control parameter received was a print without spacing parameter. Print-and-space carriage control parameters (as well as the print-without-spacing carriage control parameter) associated with DJDE records are ignored.

The printer carriage control (PCC) byte associated with offline DJDE records is ignored.

If a DJDE is always created because of coding procedures, but there is no need to set any of the DJDE parameters for a particular report, a null DJDE may be created with only an END parameter and no other parameters specified, which will not modify any of the existing job setup parameters.

FORMS, FORMAT, and BFORM DJDE parameters can appear only once in the DJDE packet.

Application of DJDEs

The following occurrences should be noted when using DJDE records:

The specific parameters included in a DJDE packet are the only ones modified when the DJDE is applied. The only exceptions are the JDE=DJDE and JDL=DJDE, which cause all processing parameters contained in the JDE to be updated except for those listed in the "Job parameter modification restrictions" section. The modified parameter remains in effect until that parameter is encountered in another DJDE packet or until the end-of-report is reached. At the start of the next report, either of the following occurs:

If in multireport mode or operating online, the next report begins with all of the parameters specified in the original JDE invoked by the operator’s START command.
• If in single-report mode, the system stops, waits for the operator to enter a new START command and then processes the next report with all processing parameters defined in the JDE invoked by the new START command.

Figure 8-1  Single record DJDE and IDEN command

Figure 8-2  Multiple record DJDE and IDEN command

Using the JDE=DJDE to reassign the top-of-page channel assignment (generally channel 1), when the DJDE is a data record, may incorrectly position the first page after the page transition occurs. Page-oriented DJDEs, like the JDE=DJDE, are implemented when the LPS detects the switch from one physical page to the next. The normal page transition is caused by a skip from the bottom of a page to the top of the next, generally channel 1. The LPS knows a page transition has occurred only because the channel 1 assignment is on a line number less than the present line number of the current page, and therefore a page transition must have been created. Unfortunately, positioning to the old channel 1 line number has already occurred. The new JDE parameters are implemented after the page transition. There are several possible solutions to this problem.

• You may add an ASSIGN=DJDE to the DJDE packet to change the applicable channel, such as channel 1, to the new line number. Since the ASSIGN DJDE is a line-oriented DJDE, it is implemented on the next line and thus affects the next channel skip.
• You may insert a record after the DJDE packet to cause the page transition and then allow a second skip to channel 1 to position to the correct line. This causes a blank page to be printed if the new channel 1 is less than the old but will be totally transparent if the new channel 1 assignment is greater (further down the page) than the old.

• In the online environment only, the carriage control on the DJDE record itself is honored if it is a SKIP parameter. If the DJDE uses the ASSIGN parameter to specify a new channel 1, a skip-to-channel-1 carriage control on the DJDE record causes a skip to the new channel 1. But if the DJDE uses JDE= to specify a new channel 1, the skip-to-channel-1 carriage control on the DJDE record causes a skip to the old channel 1, until a physical page break occurs.

**DJDE operator information pages**

The changes to the JDE, specified in the DJDE, are incorporated when END parameter is encountered. The changes begin on the next record or page following the last DJDE record.

The specification OPRINFO=YES in the JDE ensures that the DJDE records are printed and sent to the tray at the next page transition after an END parameter. The DJDE records are printed on a separate page from the report data. However, no page formatting occurs and DJDEs on one long record are truncated at the end of the physical page boundary. They are printed and delivered to the tray regardless of the OPRINFO parameter if the DJDE contains an error. Comments in the DJDE may be used for operator notification or output routing instructions in conjunction with the OPRINFO parameter.

In duplex processing, DJDE records are printed as duplex pages with a blank page on the back (and are counted on the accounting sheet). If a DJDE is applied at the transition to the back side of a duplex page and OPRINFO = YES, the DJDEs are printed following the back side page. A row of asterisks (******) appears on the DJDE page following the parameters whose printing is so delayed.

The message:

***MISSING END COMMAND OR MISSING PAGE BOUNDARY

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

• No END parameter is found in the DJDE.

• No complete page boundary was found before the end of the report, that is, at the end of report there were some DJDEs that had not yet been applied.

**Job parameter modification restrictions**

To process a DJDE, the system has already been required to process data up to the recognition and interpretation of the DJDE itself. This requires that the basic description of the input source must have been correct prior to the application of the
DJDE. Thus, the description of the input source cannot be changed through DJDE, either by accident or by intent. The system does not allow these basic parameters to be changed through the DJDE parameter JDE (as defined in table 8-4). The JDE specified by JDE=DJDE is referred to as a selected JDE. The parameters which cannot be changed through a selected JDE are listed in figure 8-3.

Figure 8-3 Parameters which cannot be changed through a JDE

- VOLUME
- HOST, LABEL, UNPACK
- OUTPUT
- OFFSET
- IDEN
- All parameters
- BLOCK
- RECORD
- BANNER

Changing the online BANNER offset criteria or VOLUME command OPTIMIZE parameter in a selected JDE causes unpredictable results and should not be attempted.

The selected JDE should specify all JDE parameters which stay the same as the original JDE as well as the changes. JDE parameters which can be changed are as follows:

Figure 8-4 JDE parameters which can be changed

- LINE
- DATA, FONTINDEX, MARGIN, OVERPRINT, PCCTYPE, VFU
- OUTPUT
- All parameters except OFFSET
- VOLUME
- TCODE, CODE
- ACCT
- EXPORT
- MESSAGE
- PDE
- ROUTE
- ROFFSET
- RO Aux
- RPAGE

- RSTACK
- TEST (online only)

If a front cover is invoked in a selected JDE which takes effect after the first page of the report, a cover is pulled from the AUX tray at the point where the DJDE is applied.

Note: CODE changes to data may occur and will take effect on the page boundary following the DJDE record. If a translation code change is involved in a selected JDE, the parameter portion of any subsequent DJDE must be in the new code. The prefix used by the IDEN must, however, retain the original hexadecimal value. Thus, if the CODE changes from EBCDIC to ASCII, the
prefix must remain in EBCDIC even though the parameter portion changes to ASCII.

**Duplex DJDE page printing**

When certain DJDEs (listed below) are invoked between an odd-numbered page (front side) and an even-numbered page (back side), the following printing sequence occurs. The odd-numbered page is printed as usual, but with a blank back side. What would have been printed on the back side (the even-numbered page) is printed on the front of the next sheet. This occurs for the following DJDEs:

- COPIES
- BFORM
- OTEXT
- RTEXT
- JDE
- JDL
- DUPLEX=NO
- FEED

**Effect on multiple logical pages**

If multiple BEGINs are being used to define multiple logical pages on one physical page (as described in the “Multiple logical pages on physical page” section in chapter 6) the following points should be considered:

- With the exception of DEPT and SHIFT, all page-oriented DJDEs are applied at a logical page boundary.
- The following DJDEs, when applied, cause the current page position to be set to the first logical page of the front side of a new physical sheet:
  - DUPLEX
  - COPIES
  - JDL
  - JDE
  - BFORM
  - OTEXT
  - RTEXT
  - FEED
- The following DJDEs, when applied, cause the current page position to be set to the first logical page of the next physical side:
  - FORMS
  - FORMAT
  - BEGIN
  - PMODE
  - FONTS
COPIES=processing

If the system is approaching disk saturation, the output task may have already released disk space for a report that increases its copy count from an initial count of one to a higher number. To avoid this, do one of the following:

• Set the number of copies of the first page to 2 or higher
• Include a copy-sensitive CME
• Use the COPIES parameter in the START command.

Online DJDE restrictions

If an online job has header pages only (no trailers), DJDEs in the first header page of a report are recognized but not applied. Also, if headers only are defined, DJDEs invoked within the current report are applied to the first header page of the following report. These conditions result from the fact that the end-of-report processing for the first report is not detected until the first header page of the following report is received and processed.

DJDE processing optimization

A slight optimization in the processing of the FORMS, FORMAT, MODIFY, ALTER, CANCEL, GRAPHIC, and IMAGE DJDEs may be realized if the following conditions are met:

• The FORMS, FORMAT, and MODIFY DJDE parameters are contained within a single record with no other parameters.
• The ALTER, CANCEL, GRAPHIC, and IMAGE DJDE parameters each are contained within a single record with no other parameters.
• The first DJDE starts at the SKIP position, and there are no blanks until the END parameter.
• Each DJDE parameter is followed by a comma, and the last comma is followed by the END parameter.
• The FORMS, FORMAT, and MODIFY DJDE parameters are limited to the following syntax:

  FORMS=form-id
  FORMAT=pde-id
  MODIFY=cme-id
**DJDE parameter definitions**

The following lists each DJDE and its options.

### ALTER

**Syntax**

```
ALTER = (un cm in dots xdots, hpos un cm in dots xdots, [n[/d]])
```

**Function**

Specifies new imaging parameters for graphics previously referenced using the hold (automatic reimaging) parameter.

**Option(s)**

- **name** - Identifies a previously called out graphic (refer to DJDE IMAGE parameter) which is currently subject to the hold parameter of its calling command. ALTER maintains the hold parameter in effect but also specifies a new set of imaging parameters for the current and and subsequent pages.

- **vpos** - Specifies the vertical position of the top edge of the graphic, relative to 0,0 on the current physical page in centimeters (cm), inches (in), (specified as a decimal number with up to three digits to the right of the decimal point), dots, xdots, or user-defined units (un). If un is specified, the user-defined positioning unit must previously be defined by the UNITS parameter of the OUTPUT command of the current JDE or JDL (refer to chapter 6). If no units are specified, inches are assumed.

- **hpos** - Specifies the horizontal position of the left edge of the graphic relative to 0,0 on the current physical page. The form of the specification (un, cm, in, dots, or xdots) is the same as for vpos.

- **n/d** - Specifies the reference scale factor. Each parameter, n and d, must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied.

**Examples:**

```
ALTER=(BUS,5,5,1);
ALTER=(IMG 1,3,5), END;
```

### ASSIGN

**Syntax**

```
ASSIGN = (channo,lineno)
```

**Function**

Specifies an assignment of a VFU channel number to a page line number or set of line numbers. It is possible to have multiple
ASSIGN parameters within a DJDE. They take effect at the next record following a DJDE END parameter. DJDE assignments affect only those channel assignments specified. Other assignments remain the same. The ASSIGN parameters are the same as for the VFU command ASSIGN parameter described in chapter 6.

**Option(s)**

- **channo** - The number of the channel being assigned. It is an integer in the range 0 to 15. The user may end the VFU command with a semicolon and start another VFU command without an id field to continue specification for the same channel or a different channel.

- **lineno** - 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.

---

**BSEQ**

**Syntax**

```
{ [BSEQ=RESET| NO RESET| NO SEQ] }
```

**Function**

Allows processing of barcode data streams with the ability to cross-check the three components of the system, namely, the PC-UI, the ESS and the IOT. By cross-checking, the 4635 ensures that any barcode sheets printed has been accounted for by each component of the software.

**Options**

- **RESET** - Indicates that the sequence number should be reset to a value of one at the beginning of each report. This is the default.

- **NO RESET** - Indicates that the sequence number will be continued across report boundaries.

- **NO SEQ** - Indicates that for pages matching the odd-even-both specification, the presence bit will be set, and a sequence number of zero will be specified. This indicates that the IOT should simply report the value of any barcodes encountered on these pages. No error is generated when no barcode is read on a page where one is expected.

**Default**

Default is RESET
BSKIP

Syntax \{ [BSKIP=READ| SKIP] \}

Function Allows processing of barcode data streams with the ability to cross-check the three components of the system, namely, the PC-UI, the ESS and the IOT. By cross-checking, the 4635 ensures that any barcode sheets printed has been accounted for by each component of the software.

Option(s) READ - Indicates that all presented sides of all sheets will be scanned for barcode information, regardless of the presence information for that sheet. Barcodes encountered when none are expected generate an error condition.

SKIP - Denotes that barcode information is processed only for pages with presence information indicating that barcodes are expected. In other words, barcodes encountered when none are expected are ignored. This is the default.

Default Default is SKIP

BSIDE

Syntax \{ [BSIDE=ODD| EVEN| BOTH] \}

Function Allows processing of barcode data streams with the ability to cross-check the three components of the system, namely, the PC-UI, the ESS and the IOT. By cross-checking, the 4635 ensures that any barcode sheets printed has been accounted for by each component of the software.

Option(s) ODD - Indicates that the system will assign sequence and presence information assuming that only odd sides are to be considered. This is the default.

EVEN - Indicates that only even sides will be considered when assigning sequence and presence information. If EVEN is coded for a simplex job, the job will contain no bar code sequence or presence information.

BOTH - Indicates that add sides of simplex sheets and both sides of duplex sheets will be considered in sequence and presence processing.

Default Default is ODD
Points to note—BARCODE command

- Sequence numbers can range from 0 to 65535, but 0 and 65535 are reserved for system use (refer to the BAR ALI command in the 4635 LPS Operator Guide and BARCODE NO SEQ commands in this chapter).

- At the start of a job, the sequence number is always reset to 1. It can also be reset when a user-specified ABORT I or an INPUT processing forced abort was initiated.

- The READ parameter is not intended to physically enable (or disable) the barcode reader. It only directs the course of action taken when a barcode is read when none is expected.

Use BSEQ to either reset the sequence number at the start of a report (RESET), not reset the sequence number (NO RESET), or recognize the presence of bar codes without specifying sequence numbers (NO SEQ). The 4635 LPS makes the determination to perform RESET/NO RESET/NO SEQ processing at the start of every report. The start of report for BSEQ processing includes any DJDE packets specified prior to any data in the report. Any BSEQ DJDEs specified thereafter are ignored, including any JDE/JDL DJDEs.

BATCH

Syntax

```
BATCH = [START
        END]
```

Function

Supported for online batch mode jobs only. Permits normal processing of online banner pages for batch mode jobs by delimiting the batch mode graphic data. Must be used precisely to mark the beginning and ending of online batch mode data. The first graphic (or IMAGE DJDE) must be preceded immediately by a BATCH=START DJDE, and the last graphic must be followed immediately by a BATCH=END DJDE. This parameter, when specified, must be the only parameter in the DJDE packet.

BEGIN

Syntax

```
BEGIN = (vpos ,hpos)
```

Function

Specifies the location of the starting print line of a logical page for graphics. The vpos parameter specifies the vertical position of the first character of the first print line on the logical page. It may be specified in inches (IN) or centimeters (CM). The default is inches.
Option(s) vpos - Specifies a decimal number with up to three digits to the right of the decimal point, for example, 0.563IN and 2.35CM are all legal specifications.

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.

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

Syntax BFORM = \{form-id, \{form-id, [init, [copies]]\}\}

Function Specifies that a form be printed on the back side of a user page (must have DUPLEX=YES). It takes effect as soon as the data is read into the system. BFORM parameters are the same as for the OUTPUT command.

Option(s) form-id - Specifies a file name which exists on disk. This file is created by compiling an FSL source file with the FDL system task. form-id may be the keyword NONE, which means that no form is to be added to the associated report page of variable data.

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.

NONE - Specifies that no form is to be added to the associated report page of variable data.

Example:
BFORM=(FORM 1, 2), END;

BOF

Syntax BOF = value

Function Specifies the bottom-of-form line number. It takes effect at the next record following a DJDE END parameter. A value is defined the same as for VFU BOF in chapter 6.

Considerations If bottom-of-form (BOF) is altered by DJDE to a line number smaller than the current line from which the DJDE was processed,
input effects an immediate page transition. If carriage control action to perform at BOF is specified as OVR (the default), input spaces from TOF a number of lines equal to the difference between its location at DJDE and the new BOF. Refer to the PCC ASSIGN command in chapter 5 regarding alternate BOF parameters.

**BTEXT**

**Syntax**

BTEXT = 'RNA=xxx],CJN=sss],[DPT=xxx],[LDT=*
W|BW|SM|M|Q|0..365|yy-mm-dd]
[RSQ=Y|N],[DDE=Y|N],[DAR=SUM
-WMGT|SUM-OLOG|SUM-BO TH|DET
-WMGT|DET-OLOG|DET-BO TH],[RRA=nnn]
[NSE=sss],[PRD=yy-mm-dd],[SEQ=sss]
[RRA=nnn],[TXT=xx..x]'

= NONE

**Function**
The BTEXT command is implemented primarily to support the PC-UI report audit logging feature which creates an audit log on the UI hard disk. The audit log is initiated and terminated by including enabling data in the job stream. BTEXT options syntax checks are performed by SYS-UI. BTEXT report definition parameters are, however, useful without the installation of a bar code reader. Page definition parameters, on the other hand, are meaningless without barcode text or a bar code reader.

The BTEXT command contains the following parameters for both report definition and page detail.

**Option(s)**

RNA - the report name with maximum of 16 characters in length. This keyword is required in the first BTEXT of each report to be audited. It may only occur once per report. For clarity in relating the audit reports to SFS reports, this should be the same value for job accounting. This parameter must be present to audit a job or report.

CJN - the customer job number with a maximum of six digits. For clarity in relating the audit reports to SFS reports, this should be the same value used for job accounting.

DPT - the department name with a maximum of 31 characters. For clarity in relating the audit reports to SFS reports, this should be the same value used for job accounting.

LDT - the log stale date indicator. Once this date has passed, the audit log may be deleted or overwritten without warning to the operator. Options include:
(U)nique specifies that report is to be run only once, ever. Operator intervention at PC UI is required to re-run the report or delete the audit log.

(W)eekly
(B)i-(W)eekly
(S)emi(M)onthly
(M)onthly
(Q)uarterly

* (Asterisk). Specifies that operator is not notified of re-runs but the report is renamed with a unique version as if the operator had been notified and accepted the re-run. Reports re-run with this indicator stay in the Audit Log directory until deleted by the operator using a PC UI window provided.

0..365 is the number of days where zero days specified means the report can be run as frequently as desired, and will not be checked for accidental re-runs.

yy-mm-dd y: year, mm: month, dd: day

RSQ - report sequence checking enablement. If (Y)es, the audit report flags out of sequence sheets delivered. This option must be set to (N)o if out of sequence sheets are delivered. For out-of-order deliveries, the job must be run in NOSEQ or BAR ALI mode so that the IOT will not catch and purge the mis-sequenced pages.

DDE - duplicate detection enablement. If (Y)es, the audit report flags duplicate sheets delivered. This option must be set to (N)o if multiple copies of a report or pages within the report are printed. For out-of-order deliveries, the job must be run in NOSEQ or BAR ALI mode so that the IOT will not catch and purge the mis-sequenced pages.

DAR - the normal audit report option. DAR parameters are a combination of a summary/detail option and an exception data reporting option. For example, SUM-WMGT specifies a summary report with waste management data shown chronologically.

**Summary/detail options:**

(SUM)mary displays the audit report headings, the total sheets delivered to each destination, including scrap, expected sheets if specified, and reconciles reports amount if specified.

(DET)ail displays the audit report headings, detail information about each sheet expected and delivered, the total sheets delivered to each destination, including scrap, expected sheets if specified, and reconciles reports amount if specified.
Exception data reporting options:

(W M G T) specifies waste management data shown chronologically.

(O P L O G) specifies system events, operator commands, and responses that may relate to the production of the report shown chronologically. Examples would be a job abort, ESS boot, etc.

(BOTH) specifies both W M G T and O P L O G

RRA - the report reconciliation amount, the total monetary value of the report expected. Maximum of 11 digits plus a decimal point may be specified. Commas are not allowed. This option is required to print an audit report reconciliation on amount.

NSE - the number of sheets expected for the report. Up to five digits, for a maximum value of 99999, may be specified. This option is required in order to print an audit report reconciliation on number of sheets consumed. The maximum value of 99999 takes into account non-barcode pages in the report.

PRD - the process date. If specified, it prints in the audit report header: yy: year, mm: month, dd: day.

Page Detail parameter

Page detail parameters are used in a BTEXT command for every page for which detailed auditing is desired. These parameters are meaningless without barcode text printed on the page, or a bar code reader.

SEQ - Sequence number. Up to 5 digits, for a maximum value of 65534. This is required to identify non-delivered, out of sequence, or duplicate pages. The value of this parameter must match the barcode text to be printed on the page.

Note: This parameter is required if the PRA or TXT parameters are present.

PRA - Page reconciliation amount. The total value of the page. Up to 11 digits plus a decimal point may be specified. Commas are not allowed. This option is required to print an audit report reconciliation on amount.

Note: This parameter is dependent upon the SEQ parameter.

TXT - Identifying text. Maximum of 64 characters. If present, prints as detail information for each page expected or delivered as reported on the audit report.

Note: This parameter is dependent upon the SEQ parameter.

Default value:

NONE - Specifies no BTEXT. If BTEXT is not specified, defaults to NONE.
Parameter rules

- All parameters have the form
  Keyword=value.
- values must be specified as shown in the BTEXT command syntax. A value may be a keyword as shown in the BTEXT command syntax or text described as follows:
  a) xxx Alphanumeric text, including spaces. Case sensitivity is retained.
  b) nnn Numeric text, including one optional decimal point. Maximum of two digits after the decimal point.
  c) sss Numeric text, no decimal points.
  d) yy-mm-dd Numeric text describing year, month, and day. Each pair of characters is separated by a dash. Eight characters in all are required.
- Parameters are separated by a comma.
- values cannot have commas embedded.
- White space within a parameter is ignored if it immediately precedes or follows the "=" sign.
- White space following a value is ignored.
- White space preceding or following a comma is ignored.
- A comma following the last parameter is ignored.
- Parameter keyword text is case insensitive.

Points to note—BTEXT command

- If RNA (report name) is not specified, no audit logging is performed.
- Bar code text that prints on the pages is placed in the job stream by the application to enable barcode features.
- Although the BTEXT command can be placed in the JSL, it will usually be placed in a DJDE as most of the parameters will be unique to a job. When placed in the JSL, it is specified under the PDL MESSAGE command.
  Example: MESSAGE BTEXT='RNA=CK0409';
- Unlike the PDL OUTPUT command, where all parameters are treated independently from one another, BTEXT parameters are treated as one entity. This implies that parameters specified at the job level do not override those specified at the system level on an individual basis. However, as a whole, the job level BTEXT does override the system level BTEXT. For example, if the system level contains the RNA and DPT parameters, but the job level only contains the RNA parameter, INPUT Processing does not use the DPT from the system level and the RNA from the job.
level. Instead, INPUT Processing only uses the RNA (with no DPT) from the job level and sends that information to the PC UI.

**Caution**

A word of caution must be mentioned as to when BTEXT information is sent to the PC UI. For every report, INPUT Processing reads the JDE/JDL specified in the START command at report set-up time. In addition, it also performs the BTEXT JDE to JDL overrides. Although BTEXT is processed at this time, the information is not immediately sent to the PC UI until after the first record of the report is read and processed. If the first record is a DJDE BTEXT, the DJDE BTEXT overrides the JDE/JDL BTEXT. Again, no combining is done. If the JSL has RNA and DPT parameters and the DJDE BTEXT has just an RNA parameter, only the RNA parameter from the DJDE BTEXT is useful for auditing.

Once received by the PC UI, a set of PC UI BTEXT parameter override rules must be observed. At the start of every report, the PC UI enables receiving BTEXT with a new RNA parameter. When this parameter is received, the PC UI ignores any further report detail parameters within the same report. This implies that overriding occurs with report detail parameters received after the start of the report but before the RNA parameter. After the RNA parameter is received, if report and page detail parameters are intermixed within the same BTEXT command for future pages, only the page detail parameters are processed. If there are multiple values for a BTEXT parameter within the same packet sent to the PC UI, only the last value is used. For example, if there are multiple DPT parameters specified in a DJDE packet which also contain the RNA parameter, the last DPT parameter is used.

- If a BTEXT is placed in the JSL and a DJDE BTEXT is specified as the first record in the job, the DJDE BTEXT overwrites the BTEXT specified in the JSL.
- When a DJDE BTEXT is used, it must precede all data for the page to ensure it will be forwarded to the PC UI before the page is processed. The ESS software does not check for this. It is up to the user to code the application correctly.
- Multiple DJDE BTEXT commands on a page must appear in the same DJDE packet because only one page-oriented DJDE packet can take effect per page.
- The BTEXT report parameter(s) must precede the first BTEXT page detail parameter. The ESS software does not check for this. It is up to the user to code the application correctly.
- Multiple values for BTEXT parameter(s) are accepted within the same DJDE packet, but only the last value in the data stream is used.
- Report parameters can only be specified at the beginning of the report. If they are specified again at some later point in the report, they are ignored.
- Reports with multiple copies are not expected to be audited in the normal course of operations, as in the case of check printing, this implies multiple copies of checks.
• To accommodate PDL record length constraints, BTEXT commands that do not fit in one PDL record can span multiple records by grouping parameters under separate BTEXT commands.

Example:

```
MESSAGE BTEXT='RNA=CK0409,CJN=HR0525',
    BTEXT='RRA=153249.37,NSE=2490',
    BTEXT='LDT=03-07-94,RSQ=Y';
```

Note, however, that BTEXT parameters cannot span PDL records. Every BTEXT parameter must start and end in the same PDL record.

• To accommodate short record lengths in the data stream, the information to be communicated in a BTEXT command can be placed into two or more consecutive DJDE records within a DJDE packet.

Example:

```
$DJDE BTEXT='RNA=CK0409,CJN=HR0525,
    RRA=153249.37,NSE=2490',END;
```

• The maximum length of all BTEXT strings within a DJDE packet or in the JSL MESSAGE command is determined by the following:

```
508-N where N is the number of BTEXT commands in the JSL MESSAGE command or DJDE packet.
```

Due to string length limitations, it is imperative that BTEXT strings not be padded with unnecessary blank characters.

Example 1:

```
$DJDE BTEXT='RNA=CK0409,CJN=HR0525';
$DJDE BTEXT='RRA=1533249.37,NSE=2490',END;
```

In Example 1, the maximum length allowed for the BTEXT string is: 508 -2 = 506 (N is 2 as there are two BTEXT commands).

Example 2:

```
MESSAGE BTEXT='RNA=CK0409,CJN=HR0525',
    BTEXT='RRA=153249.37,NSE=2490',
    BTEXT='LDT=03-07-94,RSQ=Y';
```

In Example 2, the maximum length allowed for the BTEXT strings is: 508 - 3 = 505 (N is 3 as there are three BTEXT commands).

• The maximum length of each DJDE BTEXT string is 255.

• Values for required BTEXT parameters not supplied in a DJDE or JSL will be supplied defaults by the PC UI if defaults exist for the parameter.

• BTEXT commands are treated as text strings by the PDL/DJDE compiler and forwarded to the PC UI to process. Therefore, it is possible that JSLs containing BTEXT will compile successfully even though errors exist in the BTEXT command string. It is up to the PC UI to process the BTEXT commands and flag any errors found. The same goes for BTEXT DJDEs.
Printing does not stop and Auditing will not take place if any BTEXT errors are found.

- If BTEXT is specified and you are not running in PC UI mode, INPUT issues a warning message and allow the operator to continue or abort. If the operator chooses to continue, the BTEXT commands are ignored. Otherwise, the job will be aborted.

- Although the BTEXT parameters can all be specified under one command, it is recommended that you group the report parameters and the page detail parameters into separate BTEXT commands.

Example 1  Sample minimal data to audit a report with monetary and sheet reconciliation:

$DJDE BTEXT='RNA=CK0409,RRA=74698.32,NSE=4985',;
$DJDE BTEXT='SEQ=1,PRA=153.65,TXT=901 R. Adams',END;

Example 2  DJDE to print a report with sheet reconciliation only:

$DJDE BTEXT='RNA=Personnel Reviews, CJN=HR052,NSE=249',;
$DJDE BTEXT='SEQ=1,TXT=326769 Harold R. Adams',END;

Example 3  Sample DJDE to audit a report with sheet reconciliation only, and without page details:

$DJDE BTEXT='RNA=Admin 31,NSE=4230',END;

Example 4  Sample JSL for a test run. The same report name will be used each time, and may be run as frequently as desired:

MESSAGE BTEXT='RNA=Test01,LDT=0';

C

Syntax  

C text

Function  The C text parameter allows for comment text to be included in the DJDE record. The C must be followed by a space (not by an equal sign), and must begin at the skip position specified in the IDEN command. Any text up to a semicolon, comma (unless semicolon or comma is enclosed in parentheses), or end-of-record is treated as commentary.

CANCEL

Syntax  \n
CANCEL= \{(name1 [,name2][,...])

Function  Cancels, effective on the current page, the hold (automatic reimaging) specified by the H parameter of an IMAGE or GRAPHIC parameter.
**SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)**

**Option(s)**
- **name** - Identifies the graphic(s) being held.
- **ALL** - Specifies that all graphics being held are to be cancelled.

### COLLATE

**Syntax**

```
COLLATE= [YES] [NO]
```

**Function**

Specifies whether collated or uncollated mode is to be used. It takes effect at the next page boundary but must occur prior to the first data record of the report. In duplex processing, `COLLATE=NO` is ignored.

**Option(s)**
- **YES** - Specifies that report copies are to be printed in collated mode.
- **NO** - Specifies that report copies are to be printed in uncollated mode.

### COPIES

**Syntax**

```
COPIES= number
```

**Function**

Specifies the number of copies of a report to produce at the next page boundary. In duplex processing, if `COPIES` appears before the back side of a duplex page, a blank back sheet is output, and the print data continues on the next front page. If number is set to zero, this copy count remains in effect until a new copy count or end of report is encountered. DJDEs encountered while number is zero is in effect are processed normally.

The number of copies specified in the DJDE overrides the number of copies in the OUTPUT command. You may encounter problems when the copy count at the start of the report is 1 (either by virtue of the JDE or the last DJDE encountered at the start of report), and the copy count is later increased. There is no problem if the `COPIES=DJDE` is in the delimiter at the start of report and the value of the `COPIES=DJDE` is at least 2. The problem is caused when output processing releases space as soon as it believes that it no longer needs it.

The following procedures are suggested to avoid the problem:

- Set the number of copies of the first page to 2 or higher.
- Include a copy-sensitive CME.
- The `COPIES` parameter in the START command can be used. COPIES parameter in the START command overrides the DJDE parameter, COPIES=; in this case, the DJDE COPIES= parameter is not applied. Therefore, it is possible to run the same job with the COPIES parameter in the START command.
and one without the COPIES parameter in the START command and get two different outputs.

**DATA**

**Syntax**

\[ \text{DATA} = (\text{pdo, length}) \]

**Function**

Specifies the location and length of printable data within an input record. It takes effect at the next record following a DJDE END parameter. DATA parameters are defined in the LINE command in chapter 6.

**DEPT**

**Syntax**

\[ \text{DEPT} = \text{sc} \]

**Function**

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. The DEPT parameter is defined in the ACCT command in chapter 6.

**DESTINATION**

**Syntax**

\[ \text{DESTINATION} = \begin{cases} \text{BIN} \\ \text{TRAY} \\ \text{EXPORT} \end{cases} \]

**Function**

Determines the delivery destination of paper.

**Option(s)**

- BIN - Specifies delivery to the currently active bin.
- TRAY - Specifies delivery to the sample tray. Transparencies should only be sent to the sample tray. Use caution when selecting the sample tray because of its limited capacity.
- EXPORT - Specifies that sheets be sent to the bypass transport. Export is honored only if SELECT=AUTO is specified by the operator.

**DUPLEX**

**Syntax**

\[ \text{DUPLEX} = \begin{cases} \text{YES} \\ \text{NO} \end{cases} \]

**Function**

Specifies whether printing is to occur on both sides of a page (duplex mode) or a single side (simplex mode). It takes effect on the next page boundary. The DUPLEX parameters are defined in the OUTPUT command in chapter 6.
END

Function Specifications 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.

Default No default

FEED

Syntax FEED = {stock-reference
stock-name
MAIN
AUX
OPR}

Function Controls the stock on which the page is printed. FEED=stock-reference references the stock assigned to a stock-name by the STOCKSET command in effect at the time the page is printed.

Option(s) stock-reference - Allows users to change stocks associated with a job without altering the stock references in the data application itself.

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.)

MAIN, AUX, and OPR - Provide compatibility to existing applications. FEED=OPR is equivalent to FEED=MAIN.

Note that the terms cluster-name and stock-name are synonymous.

FILE

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

Function Provides a capability of loading files to the LPS disks while a print job is in progress. The print job may be invoked solely for the purpose of downloading permanent files. Although intended primarily for use with online systems, this parameter is not restricted to online use.

Option(s) file-name - is a 1 to 6 character name used to identify the disk file.

file-type - is a 3 character file type symbol.
**SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)**

- **f** is one of the following file input format parameters:
  - C = card-image format (default)
  - L = LPS-labeled tape format
- **s** is one of the following file storage parameters:
  - D = delete after report printed (default)
  - P = permanent
- **n** is the maximum number of card-images (default=120).

For LPS-labeled files, the DJDE FILE parameter may be specified as follows:

```
FILE=( )
```

In this case, the file name and file type are obtained from the label record that precedes the file data.

**Considerations**

The input task handles the creation and deletion of files which are transmitted from a host. The output task marks files for deletion at the end of the report which created them. The input task checks for files to be deleted at the start of the job, at report setup, and before exiting. Note that when the output task marks the files for deletion, the input task has long since finished with the report that created the files and is either waiting for data or actively processing another report. If the input task is processing a report, the files are deleted after end-of-report processing for the current report. After the files for the previous report have been deleted, the input task then completes report setup for the report that follows.

(Refer to the “DJDE FILE command” later in this chapter.)

---

**FONTINDEX**

**Syntax**

```
FONTINDEX= (offset [,init-val] [,bit-opt])
```

**Function**

Allows the user to specify a certain location in the input record where an index to the font to be used is stored. It takes effect on the next logical page boundary. The FONTINDEX parameters are the same as in the LINE command in chapter 6.

---

**FONTS**

**Syntax**

```
FONTS= (f1 [,f2] [,...])
```
**Function**

Specifies the fonts to be used in printing variable input data or CME data. The FONTS parameter within a DJDE takes effect on the next page boundary.

**Option(s)**

$f_1$ - Each font index ($f_1$) specifies a 1 to 6 alphanumeric character identifier (consisting of at least one letter) corresponding to a font cataloged on the system disk (up to 128 fonts can be used with the font indexing capability and up to 94 fonts on a single page). If the number of fonts to be invoked exceeds the size of one DJDE record, multiple FONTS parameters must be used.

$s_1$ - Each spacing index ($s_1$) value specifies an optional override line spacing value specifying either lines per inch or dots per line to be associated with the font; maximum value for $s_1$ is 30 LPI; minimum value is 10 XDOTS or dots. If a unit (LPI, XDOTS, or DOTS) is not specified, LPI is used as the default.

If an override line spacing value is specified, lines printed using the font cause the indicated line spacing to be performed after the line using the font. If different fonts are used on the same print line, the line spacing value specified for the font of the largest character in the line is used to determine the position of the next print line. (Refer to the “Points to note” section.)

---

**Points to note—FONTINDEX and FONTS**

- An xdot is 1/600 unit of measurement. A form specifying xdots, however, can be created, edited, and compiled on any LPS with version 3 software, but the form, which is 600 spots per inch (spi), will not print on your 4850, 4890, 4050, 4090, or 4635 which accepts only 300 spi jobs.

- Since there is ambiguity between the FONTINDEX and FONTS parameters, if either is abbreviated to the first three letters, the parameter defaults to FONTINDEX.

- In DJDE processing, if the input task (DJD) encounters more than 127 fonts in a packet, the remaining fonts in the packet are ignored or truncated. Because the DJDE packet in this case is truncated, the outcome of the DJDE is unpredictable. Thus, the DJD task sends the error message:

  0S6700 SYNTAX ERROR IN DJDE.

---

**FORMAT**

**Syntax**

FORMAT= pde-id

**Function**

Specifies an entirely new page descriptor entry (PDE) to be used for formatting control. It takes effect on the next page boundary.

**Option(s)**

pde-id - Refers to a separately cataloged file in the PDE library on disk. Parameters of the selected PDE, such as BEGIN, FONTS, and PMODE, may also be modified on an individual basis. Further information is contained in the “Page descriptors—PDE command” section in chapter 6.
FORMS

Syntax

\[
\text{FORMS=} \begin{cases} \text{form-id} \\ \text{(form-id[,init[,copies]])} \\ \text{NONE} \end{cases}
\]

Function

Specifies the form to be merged on the printed pages. It takes effect on the next page boundary. The form specified is invoked beginning with the first page of the specified starting copy number. Multiple FORMS parameters may be used to associate different forms with different copy plies.

Option(s)

- **form-id** - Specifies a file name which exists on disk. This file is created by compiling an FSL source file with the FDL system task. form-id may be the keyword NONE, which means that no form is to be added to the associated report page of variable data.
- **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.

Example:

\[
\text{FORMS=(FORM3, 3, 2), END;}
\]

Points to note—FORMAT and FORMS

- Since there is ambiguity between the FORMAT and FORMS parameters, if either is abbreviated to the first three letters, the parameter defaults to FORMAT.
- Paper size can only be changed in a job stream by using a JDE to call out the desired paper size, that is, through the PAPERSIZE parameter of the OUTPUT command. It is important to change the image size when the paper size is smaller than the current image so that all data is printed on the page.

GRAPHICS

Syntax

\[
\text{GRAPHICS=} \begin{cases} \text{name,vpos} \\ \text{,hpos} \\ \text{,UNCM]IN\[DOTS\[XDOTS\[H\[n[/d]]]} \end{cases}
\]

Function

Specifies that the DJDE is a graphic sentinel. That is, it immediately precedes and identifies a graphic in the report data.
SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)

stream. This parameter, when specified, must be the only parameter in a DJDE packet.

If no textual data has been previously encountered, the graphic is document interleaved and copied to an IMG type disk file.

If textual data has been previously encountered, the graphic is page interleaved (block mode) and the last previous text record should have caused a page transition. If not, the occurrence of this DJDE parameter forces an immediate page termination, and the next text record encountered is processed relative to TOF on the next physical page.

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

Option(s)

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

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 in units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS. An XDOT is 1/600 unit of measurement, which is used for 600 spots per inch (spi) processing. Jobs with XDOTS can be created, edited, and compiled on any LPS with version 3 software, but will not print on your 4050, 4090, 4635, or 4850 which is 300 spi.

hpos - Specifies the horizontal position in units (UN), centimeters (CM), inches (IN), DOTS, or XDOTS.

H - Specifies that the same graphic is to be held and imaged without further callouts at the same position and with the same scale factor at all subsequent pages until changed by a DJDE CANCEL or ALTER parameter.

n/d - Specifies the reference scale factor. Each parameter, n and d, must be an integer in the range of 1 to 8, thereby allowing a reference scale factor in the range of 1/8 to 8. The reference scale factor is multiplied.

IMAGE

Syntax

Function

In batch mode, this OVERPRINT = parameter defines new imaging parameters for subsequent graphics. In all other modes, it defines imaging parameters for the named graphic.
### SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>img-name, vpos UN CM IN DOTS, hpos UN CM IN DOTS [T][H][n/d]</th>
</tr>
</thead>
</table>

**Option(s)**

**img-name** - 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 that would be specified by a PDE BEGIN of 0,0). This parameter is specified as a decimal number with up to three digits to the right of the decimal point and may be in DOTS, centimeter (CM), inches (IN), or user-defined units (UN). If UN is specified, the user-defined positioning unit must previously be defined by the UNITS parameter of the OUTPUT command of the current JDE or JDL (refer to chapter 7). If no units are specified, inches are assumed.

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

**T** - Specifies that the referenced graphic is page interleaved and follows the current page data in the input stream.

**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.

**n/d** - Specifies the reference scale factor. Refer to the “Scaling” section in your 4635 LPS Operations Reference for more information.

The parameters T, H, and n/d are not order dependent. They may be coded as needed without extra position holding commas.

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

**Example:**

```
IMAGE=(IMG1, 3, 2), END;
```

### INVERT

**Syntax**

```
INVERT= {FRONT BACK BOTH NONE}
```

**Function**

Specifies that the following pages are to be inverted. When specified in a DJDE data stream (page oriented), this feature
allows the 4635 to invert the image on a physical page by 180 degrees.

**FRONT**  Specifies page inversion on front sides
**BACK**   Specifies page inversion on back sides
**BOTH**   Specifies page inversion on front and back sides
**NONE**   Specifies turning page inversion off

### ITEXT

**Syntax**  

\[
\text{ITEXT=}\left\{ \begin{array}{l}
\text{sc} \\
\text{(sc,passnum)} \\
\text{NONE}
\end{array}\right.
\]

**Function**  Specifies a text message to be output during input processing. Takes effect on the next page boundary. The parameters are the same as for the MESSAGE ITEXT parameter defined in chapter 6.

### JDE

**Syntax**  

\[
\text{JDE=} \quad \text{jde-id}
\]

**Function**  The jde-id specifies the JDE to be used within the selected JDL at the next page boundary. In a JDE packet containing a JDE, other record- or page-oriented DJDE parameters included in the packet override parameters specified by this selected JDE. (Refer to the "Job parameter modification restrictions" section in this chapter.)

### JDL

**Syntax**  

\[
\text{JDL=} \quad \text{jdl-id}
\]

**Function**  The jdl-id specifies the name of the JDL to be invoked at the next page boundary. The jdl-id must exist on disk in the JDL directory. If only JDL is specified (not JDE) in the DJDE, the JDE to be used is the one named in the START command. In a DJDE packet containing a JDL parameter, other record- or page-oriented DJDE parameters override parameters specified by the JDL DJDE.

### MAP

**Syntax**  

\[
\text{MAP=} \quad \text{file-name}
\]

**Function**  References a previously created font mapping file (file-name) from a [label:] SEFFNT command.
MARGIN

Syntax

\[
\text{MARGIN} = \begin{cases} 
\text{value} \\
\text{(value, value-type)} 
\end{cases} 
\]

Function

Specifies the left printing margin within each logical page. It takes effect at the logical page following a DJDE END parameter. The parameters are the same as in the LINE command defined in chapter 6.

MODIFY

Syntax

\[
\text{MODIFY} = \begin{cases} 
\text{cme-id} \\
\text{(cme-id,init [,copies])} \\
\text{NONE} 
\end{cases} 
\]

Function

Specifies the copy modification entry (CME) to be used in report processing. It takes effect at the next logical physical page boundary. A cme-id refers to a file which is separately cataloged in the CME directory. Refer to the “CME” section in chapter 6 for further information. The parameters are the same as those in the OUTPUT command defined in chapter 6.

NUMBER

Syntax

\[
\text{NUMBER} = \begin{cases} 
\text{(pnum,lnum,cnum [,findex])} \\
\text{NO} 
\end{cases} 
\]

Function

Specifies page numbering control. It takes effect on the next logical page boundary. The parameters are the same as those for OUTPUT command.

Option(s)

- **pnum** - Page number: specifies the starting number (an integer) for page numbering. The beginning page number may be non-positive. The number is incremented at page transitions, but it is not printed until it goes positive. The maximum page number that can be printed is 2,147,483,647.
- **lnum** - Line number: specifies an integer line number on which the page number is to be placed.
- **cnum** - Column number: specifies an integer ending column number for the page number sequence.
- **findex** - Font index: the list of fonts specified in the FONTS parameter of the PDE command. If not specified, the first font in the list is used.
- **NO** - Specifies that no page numbering is to be performed. This is the default.
Example:
NUMBER=(1, 1, 40, 1),END;

**Default**
Default is NO

**OTEXT**

**Syntax**

\[
\text{OTEXT} = \begin{cases} 
\text{sc}, & \text{[passnum]}, \text{[WAIT]} \\
\text{sc}, & \text{[END]}, \text{[WAIT]} \\
\text{NONE} 
\end{cases}
\]

**Function**
Specifies a text message to be output to the operator during job printing. Takes effect on the next page boundary. The parameters are the same as for the MESSAGE command defined in chapter 6.

**Considerations**
Use the OTEXT message to inform the operator how to align the bar code reader for a given job.

1. Code an OTEXT message (with a “WAIT” statement in the data stream) that indicates the location and orientation that the barcode reader needs to be set. The WAIT statement in the OTEXT message stops output until the operator completes the alignment task and presses the CONTINUE key.
2. Lift the X8 top printer cover.
3. Ensure that the bar code reader matches the orientation specified in the OTEXT message. If necessary, turn the bar code reader 90 degrees in either direction.
4. Turn the green shaft knob to locate the bar code reader in the position specified in the OTEXT command.
5. Ensure that none of the fingers cross the path between the bar code reader and the paper path. If necessary, lift the finger handle on the lower front bracket and move the fingers forward or backward.
6. Close the X8 top cover.
7. Press the CONTINUE key.

Example OTEXT message:
“Open the X8 cover and position the bar code reader to 7.00” in the landscape orientation now. When you are finished, close the cover and press the CONTINUE key.”

**OVERPRINT**

**Syntax**

\[
\text{OVERPRINT} = \text{over-opt, disp}
\]
**Function:** Specifies overprint control. Overprint lines are print lines whose carriage control specifies printing with no line spacing after the last printed line. It takes effect on the next record following a DJDE END; parameter. If FONTINDEX has been invoked in a JDE or DJDE, OVERPRINT takes effect at the next logical page boundary. OVERPRINT parameters are the same as for the LINE command:

**Option(s):**

- **over-opt** - 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. (See the “Points to note” section which follows.)
  - **PRINT2** - Specifies that up to two consecutive lines are printed per line—one line and one overprint. Other overprints for the line are ignored.

- **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.

---

**PMODE**

**Syntax:**

```
PMODE=  \{ LANSDCAPE  \\
        PO R T A I T  \}
```

**Function:** Specifies the printing mode for each page. LANSDCAPE indicates that printing is to be parallel to the long edge of paper. POR T A I T indicates that printing is to be parallel to the narrow edge of paper. If PMODE is changed through a DJDE command, the BEGIN command should be respecified (even if the same horizontal and vertical position has been specified in the previous BEGIN parameter).

---

**RFORM**

**Syntax:**

```
RFORM=  \{ form-id  \\
         NONE  \}
```

**Function:** Specifies a form to be printed with all RTEXT pages. It takes effect on the next RTEXT page. The parameters are the same as for the ROUTE command defined in chapter 6.

Example:

```
RFORM=GBAR, END;
```
**RTEXT**

**Syntax**

\[
\text{RTEXT=} \begin{cases} 
\text{sc (sc, [passnum | ALL], [line], [col], [fontindex])} \\
\text{rtext-id NONE}
\end{cases}
\]

**Function**

Specifies text to be printed on a separate page preceding a report. It takes effect on the next page boundary. The parameters are the same as for the ROUTE command defined in chapter 6. RTEXT parameters must be in ascending passnum sequence.

Example:

\[
\text{RTEXT='TEST', ALL, 15, 5, 2; END;}
\]

**SAVE**

**Syntax**

\[
\text{SAVE=} \begin{cases} 
\text{(file-name[...])} \\
\text{ALL}
\end{cases}
\]

**Function**

Must occur prior to or at the end of the report and specifies IMG files updated during the current report which are not to be purged when the report's output processing is completed. This DJDE parameter is used to override, for specific files, automatic file deletions specified by PURGE=YES in the JDE's OUTPUT command. The file-name is either the name of an IMG file or a file name mask whose form and interpretation are as described for the operator SAMPLE parameter (refer to “SAMPLE parameter”). ALL specifies that all document interleaved files created by this report's processing are to be saved, overriding an explicit (or default) specification in the JDE's OUTPUT command.

**SEFMAP**

**Syntax**

\[
\text{SEFMAP=} \begin{cases} 
((\text{font1, font2}), [\text{fontm, fontn}]) [, \text{UPD | REP}] \\
\text{NONE}
\end{cases}
\]

**Function**

SEFMAP specifies a mapping table different than the one defined in the PDL command on a page-by-page basis.

**Option(s)**

font1, font2 - Font pair in which font2 is the SEF mapping for font1.
**font<sub>m</sub>, font<sub>n</sub>** - Font pair in which font<sub>n</sub> is the SEF mapping for font<sub>m</sub>.

**UPD** - (Update) indicates that the specified font pairs change existing mappings or append new mappings to the current font mapping table.

**REP** - (Replace) indicates that the specified font pairs create a new mapping table to replace the current one. REP is the default.

- With UPD or REP, only memory-resident font mapping tables (not disk-resident tables) are affected.

**NONE** - Disables short-edge feed (SEF) and font mapping (stops a previous SEFFNT).

**Considerations**
Refer to the SEFMAP examples at the end of this section.

### SEPARATORS

**Syntax**

SEPARATORS = \{FIRST \[(LAST [,D])\]\}(BOTH [,D])\} \NONE

**Function**
Specifies that the printer software save an internal copy of the next data page in the report and use it to generate segment separator sheets. If separators are already in effect when this DJDE is encountered, it replaces the prior specification and subsequent separators are generated using the new specification.

**Option(s)**

- **FIRST** - Specifies that subsequent segments should have a copy of the saved data page inserted as the first sheet of the segment (on a separate sheet of paper) as a hardcopy segment separator.

- **LAST** - Specifies that at least one copy of the saved data page should be generated (on a separate sheet of paper) as a hardcopy segment separator and counted as the final sheet in the segment.

**Note:** In an on-line environment, a true banner trailer page, if encountered, will be used to image any LAST separators of the final segment in place of the saved data page.

- **BOTH** - Specifies both FIRST and LAST, as above.

- **D** - Specifies that on a duplex system separator sheet, information is printed on the back side of any LAST sheets, even in simplex reports. If a separator is printed on the back side of a simplex report, the system processes the separator as a duplex page. The accounting page reflects this information.

- **NONE** - Specifies that no segment separator sheets are generated. However, the system continually updates segment numbering even if no separator sheets are printed. If SEPARATORS or SNUMBER are specified again in a DJDE, the updated number is used.
**SF1FUNCTION**

Syntax  

\[
\text{SF1FUNCTION} = \begin{cases} 
\text{YES} \\
\text{NO} 
\end{cases}
\]

Function  

Specifies to the finisher whether (YES) or not (NO) to begin a particular operation on the current sheet. SF1FUNCTION invokes the C6 signal of the electronic interface between the LPS and a third party finishing device. Refer to your vendor's third party device documentation for detailed information.

**SF2FUNCTION**

Syntax  

\[
\text{SF2FUNCTION} = \begin{cases} 
\text{YES} \\
\text{NO} 
\end{cases}
\]

Function  

Specifies to the finisher whether (YES) or not (NO) to begin a particular operation on the current sheet. SF2FUNCTION invokes the C7 signal of the electronic interface between the LPS and a third party finishing device. Refer to your vendor's third party device documentation for detailed information.

**SHIFT**

Syntax  

\[
\text{SHIFT=} \begin{cases} 
(v_1,v_2) \\
\text{YES} \\
\text{NO} 
\end{cases}
\]

Function  

Specifies a shift of the image on the page for purposes of binding. It takes effect on a physical page boundary. The parameters are the same as for the OUTPUT command which is defined in table 6-1.

Option(s)  

- \(v_1\) - An integer value in dots for the amount of shift on the simplex page or the odd side (front) of the duplex page. (Each dot is 1/300 of an inch.)
- \(v_2\) - The value in dots for the amount of the shift on the even (back) side of the duplex page. For example, if \(\text{SHIFT}=75\), \(-5\) is specified, the amount of the shift is 1/4. It is important to ensure that the data does not pass the margin and go off the page.

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

No - Specifies that no shift occurs \((v_1=0 \text{ and } v_2=0)\).

Default  

Default is NO
**SIDE**

**Syntax**

\[
\text{SIDE=} (\text{side-opt, offset-opt})
\]

**Function**

Specifies repositioning of the new logical page to the first logical page of the given side of a physical sheet. It takes effect at a logical page boundary. The parameters are the same as for the RPAGE command as defined in table 8-8. The SIDE DJDE triggers repositioning for only a single logical page and does not otherwise override the SIDE parameter of the RPAGE command. It is intended to be used instead of the RPAGE command, not with the RPAGE command. The SIDE DJDE overrides both the RPAGE SIDE= parameter and the ROFFSET parameter.

**Considerations**

It is important to ensure that the data does not pass the margin and go off the page.

---

**SNUMBER**

**Syntax**

\[
\text{SNUMBER=} (\text{lnum, cnum ,}[\text{findex}])
\]

**Function**

 Specifies that the printer merge the current segment sequence number into any subsequent segment separator sheets produced. SNUMBER can be modified as a DJDE under the following conditions:

- Both SEPARATORS and SNUMBER are previously defined in either a JDE or JDL or as DJDEs.
- If the SEPARATORS parameter is not specified in the EXPORT command, it must be specified as a DJDE, along with the SNUMBER DJDE.

**Option(s)**

- **lnum** - Specifies the integer line number on which the segment sequence number string is to start.
- **cnum** - Specifies the integer column number on which the segment sequence number string is to end.
- **findex** - Specifies fontindex of the font to be used for printing the segment sequence number string. If SNUMBER is specified without SEPARATORS (under allowable conditions), the font index refers to the font list used at the time SEPARATORS was specified.

**Considerations**

lnum and cnum are interpreted using the metrics of the font specified by findex (the same as with page numbering). Refer to the “Points to note” section for additional information.
### SPLIT

**Syntax**

```
SPLIT = { NOW (min, max) OFF }
```

**Function**

Specifies the use of segment management. When segment management is in effect, an end of segment condition occurs whenever max sheets (including separators) are generated or an end-of-copy or end-of-report condition occurs.

**Option(s)**

- **NOW** - Specifies that the current or upcoming end-of-page (backside if duplex) will mark the last data page of the segment.
- **(min, max)** - Specifies the minimum and maximum numbers of sheets allowed in a segment. The minimum number of sheets allowed is 2 and the maximum is 65,535.
  
  If a segment end has not occurred by the time max sheets are generated, then the printer declares end-of-segment. If end-of-segment occurs before min sheets are generated, then the printer generates padding of either blank or multiple LAST sheets (if the LAST option of the SEPARATORS parameter has been specified). If specified with segment management already in effect, then the current segment is tested against the new max and if greater, an end-of-segment is immediately declared.

- **OFF** - Specifies that printer segment management not be provided, and if currently in effect, be immediately terminated.

**Considerations**

When SPLIT=OFF is specified, the segment number (nnn of part nnn on separator sheets) is reset from its current value and begins at 001 again when segment management is turned on again.

### SRECOVER

**Syntax**

```
SRECOVER = { PAGE SEGMENT ASK DEVICE }
```

**Function**

Specifies the recovery mode. When used as a DJDE, SRECOVER must be specified prior to the first data record in the report or it is ignored.

**Option(s)**

- **PAGE** - Specifies standard jam recovery. The printer will resume with the first sheet not delivered.
- **SEGMENT** - Specifies the reprinting of the entire segment that was being printed at the time of the jam.
ASK - Specifies that the operator be requested through the user interface to select PAGE or SEGMENT recovery.

DEVICE - Places recovery mode selection responsibility with a support third party finishing device through connection with the 4635 LPS bypass transport option.

STIMING

Syntax

STIMING = ([INTERVAL, sec][,DELAY, sec])

Function

STIMING is provided for possible compatibility with other Xerox software supporting third party finishing devices and is ignored by the Xerox 4635 LPS. The STIMING DJDE, although ignored by the 4635 LPS, specifies that the printer accommodate timing constraints needed by finishing equipment interfaced with the Programmable Bypass Transport.

STOCKS

Syntax

STOCKS = stockset-name

Function

This parameter defines the STOCKSET and its associated stock(s) to be used in a report. If the stockset-name does not refer to a STOCKSET command coded earlier in the same JDL, it is assumed that the STOCKSET exists globally as an STK file, which is read at print time. In this case, PDL inserts a message in the JSL file listing, indicating that an STK file will be used at print time.

Whenever a new stockset is chosen, that is, at start of report or through a DJDE JDE or JDL switch, each stock is checked to determine that the stock exists and can be made active. This provides an automatic method of changing stocks in the LPS as required by the data stream.

TMODE

Syntax

TMODE = (width [width unit])

Function

Specifies process direction measurement for determining default pitch modes.

Option(s)

- width - The process direction measurement that determines the default pitch mode
- width unit - Can be IN, CM, DOTS, or XDOTS. The default is IN.

Width values can range from the 7-inch equivalent to the 17-inch equivalent.
TOF

Syntax

TOF= \textit{value}

Function
Specifies the top-of-form line number. It takes effect at the next record following a DJDE END. The value parameter is defined the same as in the VFU command as defined in chapter 6.

TRANS

Syntax

TRANS= \begin{cases} \text{YES} \\ \text{NO} \end{cases}

Function
Specifies whether transparencies are used in the print job (YES) or not used (NO) in the print job.

XSHIFT

Syntax

XSHIFT= \begin{cases} \text{NO} | \text{value} \end{cases}

Function
Specifies whether the image of the form and data on a page is to be shifted, and if so, by what values. This parameter is used to shift the image of the data for two-hole paper, binding, finishing, and edgemarking. XSHIFT allows shifting in the x-direction for short-edge binding.

Options

- NO - Specifies that no shift occurs.
- value - An integer that specifies a shift amount on the front and back side. The shift is in the x-direction and each dot is 1/300 of an inch. The allowable range is 75 to -75.

Points to note—DJDE parameters

- TOF
The DJDE top-of-form (TOF) command affects the first data record following the DJDE record. Depending on the time of a page transition, this may not take effect on the desired page. In other words, even though a DJDE TOF is processed during the processing of the next record, a page transition may have occurred before the DJDE TOF value could take effect. For example, if the initial TOF and BOF values are 2 for every job, every time a data record is processed, a page transition occurs since the TOF and BOF have the same value. If a DJDE TOF with value 1 is processed and the page transition has already occurred, this DJDE TOF value is too late and does not take effect until the next page transition.
• **Short-edge feed (SEF)**
  If you want to use 11 x 17 inch paper (or any size that requires SEF) be aware that FSLs/FRMs, LGOs, and IMGs are not automatically scaled or rotated. Refer to your 4635 LPS Operations Reference for detailed information on scaling.

• **FONTINDEX**
  When multiple fonts of various sizes are used on a single page, the line spacing of any particular line is determined by the largest font in use in the previous line. The current line can also be adjusted downward by the difference in height between the first and the largest font in the current line. The line spacing of the first line in a report is determined by the first font in the PDE, provided the FONTINDEX parameter has not been specified. When FONTINDEX is specified, the line spacing of the first line of a report is determined by the font indicated by the font index value in the first line of the report. If there is no font index value in the record and FONTINDEX is specified, the line spacing for the first line of the report is determined by the first font in the PDE. When the FONTINDEX or CME parameters are specified, the line spacing of the first line of subsequent pages is determined by the line spacing of the font called out by the last font index of the previous page into the current PDL font list. If a PDE DJDE has occurred prior to the page transition, the font used for the line spacing comes from this list.

• **FONTINDEX, OVERPRINT=MERGE**
  — FONTINDEX, used in conjunction with overprinting, is handled as follows: for OVERPRINT=PRINT, the LPS overprints records analogously to an impact printer if the fonts are the same size. If the fonts differ, records are overprinted without regard to character spacing. For OVERPRINT=MERGE, the LPS replaces characters in the previous record which are blank. Character spacing values are adjusted; thus proportionally spaced or different size fonts may be used, and the LPS performs the character placement.
  — If FONTINDEX and OVERPRINT=MERGE are specified and proportional fonts are being used, data lines are not properly aligned if printing in landscape mode, and the data switches to a larger font in the middle of the line.

• **SNUMBER**
  — If the system encounters a valid SNUMBER dynamic job descriptor entry after an incorrect EXPORT command SNUMBER parameter had been specified, unpredictable results may occur.
  — SNUMBER as a DJDE may only be used to change current line or column numbers and fontindex values that are specified in the SNUMBER parameter of the EXPORT command used in the JDL. The DJDE SNUMBER may only enable segment sequence numbering if it is invoked with the SEPARATORS DJDE.

• **Ending a segment**
  A segment may be ended by a DJDE under any of these conditions:
  — SEPARATORS is specified in a DJDE
The system encounters a SPLIT=NOW dynamic job descriptor entry
The system encounters a SPLIT=OFF dynamic job descriptor entry.

• **Using a new separator page**
  You may use a new separator page by specifying it as a DJDE.

• **Resetting the segment number**
  You may set a new segment number after the current one is terminated. This is accomplished when the system encounters a series of two DJDEs: SPLIT=OFF followed by SPLIT=(min,max).

**Examples—SEFMAP usage**

**Example 1**
Before the DJDE short-edge feed parameter, the contents of the memory font mapping table where:

((font1,font2),(font3,font4))

DJDE SEFMAP parameter:

SEFMAP=((font1,font7),(font5,font6),UPD);

After this DJDE SEFMAP parameter is processed, the contents of the memory font mapping table are:

((font1,font7),(font3,font4),(font5,font6))

**Example 2**
Before the DJDE short-edge feed parameter, the contents of the memory font mapping table where:

((font1,font2),(font3,font4))

DJDE SEFMAP parameter:

SEFMAP=((font1,font7),(font5,font6),REP);

After this DJDE SEFMAP parameter is processed, the contents of the memory font mapping table are:

((font1,font7),(font5,font6))

**DJDE FILE parameter**

This DJDE provides a capability of loading card-image or LPS-labeled files to the LPS disk while a printing job is in progress. The print job may be invoked solely for the purpose of downloading files; in other situations, the files may be interleaved with variable text data so that they are available when referenced by input or output processing. In either case, a DJDE packet must immediately precede the data for a file (or batch of files) that is to be copied to the LPS system disk. The DJDE packet must include a FILE parameter used to signal the start of the file data.

The DJDE FILE parameter is record oriented and is therefore applied immediately. It may be included with other record-
oriented or page-oriented DJDE parameters in a DJDE packet. The FILE parameter is specified as follows:

\[
\text{FILE} = ([\text{file-name}],[[\text{file-type}],[[f],[s],[n]])
\]

where

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file-name</td>
<td>is a 1 to 6 character name used to identify the disk file.</td>
</tr>
<tr>
<td>file-type</td>
<td>is a 3 character file type symbol.</td>
</tr>
</tbody>
</table>
| f | is one of the following file input format parameters:  
  C = card-image format  
  L = LPS-labeled tape format (default) |
| s | is one of the following file storage parameters:  
  D = delete after report printed  
  P = permanent (default) |
| n | is the maximum number of card-images (default=120). |

For LPS-labeled files, the DJDE FILE parameter may be specified as follows:

\[
\text{FILE} = ()
\]

In this case, the file name and file type are obtained from the label record that precedes the file data.

**Points to note**

- If the file name or file type specified in the DJDE FILE parameter differ from those in the label record, the file is renamed as specified in the label record. This renaming feature for LPS-labeled files allows five restricted types of files to be downloaded as different file types.
- If a disk file already exists with the specified file name, it is replaced. Refer to the “File replacement and deletion” section for details and restrictions.
- For card-image data, the FILE parameter may be specified as follows to indicate the end of card-image data:
  \[
  \text{FILE} = (,,C) \text{ or } \text{FILE} = (\text{END},,C)
  \]
- The single characters C, L, D, and P are file storage parameter keywords for the DJDE FILE=parameter and cannot be specified as the file-name in a FILE DJDE.
- The DJDE DELETE parameter, DJDE FILE, allows the removal of the following types of files only: IMG, FRM, LGO, FNT, CME, JDL, PDE, TST, and XCS.

**File effectiveness**

Files may be used by input or output as soon as they are written to the system disk. However, they become effective only when they are referenced or invoked by a DJDE. For example, a font or form file becomes effective at the next page boundary when a DJDE changes the FORMAT or FORMS parameter to invoke the new font or form file. If a file on the disk is being replaced, the new file is not to be used until it is reinvoked by an appropriate DJDE.
File replacement and deletion

Files that may be used by input or output processing during the course of a print job can be replaced or deleted only when they are no longer required. If a file can be used by output processing, for example, FNT, FRM, IMG, and LGO, the file replacement transaction entails the temporary retention of the replaced file until the current report is printed. A file that is used only by input processing, such as CME and PDE, is replaced immediately. Thus, the files marked for deletion after the current report is printed include those that the user had specified for deletion and the superseded files that may have been required by output. Files which are transferred with a delete option are available for deletion as soon as the report is printed and therefore must not be referenced in a subsequent report.

Note: Unlike the hostcopy procedure, the DJDE FILE parameter does not allow the file replacement or deletion transaction to occur if the file to be replaced or deleted is secured and the operator's logon level is not 5.

Card-image file processing

The FILE parameter is checked for valid parameters, and an appropriate amount of disk space is made available for the file.

- If the destination file type is not acceptable for a card-image file, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE PROCESSING ERROR - ILLEGAL DESTINATION FILE TYPE.

  Input then proceeds to read and discard the extent of the file.

- If the file name is improperly specified, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE PROCESSING ERROR - INVALID DESTINATION FILE NAME.

  Input then proceeds to process the data as variable text data.

- If more card-image records are read than are predicted by the parameter specifying the maximum number of card images, the extra records are read and discarded, and the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE xxx.yyy OVERFLOW - EXCESS CARD-IMAGES LOST.

- If the file cannot be created due to insufficient space on the disk, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  NO SPACE ON DISK FOR FILE xxx.yyy.

  Input then proceeds to read and discard the extent of the file.

When a card-image file record is read, the PCC byte is ignored. Block and record delimiters (if any) are used to determine record length. Records with a length greater than 80 bytes are truncated to 80 bytes; records with a length less than 80 bytes
SPECIFYING DYNAMIC JOB DESCRIPTOR ENTRIES (DJDES)

are padded with ASCII blanks to 80 bytes. Character translation may be performed, depending on the JDE in effect.

Sequence numbers are not inserted into the records prior to storing, so the file must be retrieved by MERGE prior to editing in order to insert valid sequence numbers.

Any DJDE record serves to terminate card-image record processing for the file being downloaded.

- If a DJDE record is encountered and one or more card-image records have been written in the file, the file is truncated to an appropriate size and closed. If OPRINFO=YES, the message:

  FILE xxx.yyy CREATED

  is listed on an OPRINFO page.

- If a DJDE record is encountered before any card-image records have been filed, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE xxx.yyy ERROR - NO CARD IMAGE DATA

  and the null file is deleted.

File processing mode is concluded, and print mode is resumed with the DJDE record unless the DJDE record contains another FILE parameter.

LPS-labeled file processing

The FILE parameter is checked for valid parameters, and the LPS label record at the beginning of the file is read. If the destination file type is acceptable, the file size field is obtained from the label record, and the extent of the file is calculated.

- If the very first label record does not meet the requirements of a label record, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE PROCESSING ERROR - NO LABEL RECORD.

  Input then proceeds to process the data as variable text data.

- If the destination file type is OSD, SAF, SYS, SYS$, or TSK, or if the file type specification in the DJDE parameter is not one of the file types included in the LPS file directory, the preceding DJDE packet is printed on an OPRINFO page together with the message:

  FILE PROCESSING ERROR - ILLEGAL DESTINATION FILE TYPE.

  Input then proceeds to read and discard the extent of the file.

- If the file cannot be created due to insufficient space on the disk, one of two courses of action is taken. If file space is freed up after a previous report is completely printed by output, input displays the message:

  DISK FULL - INPUT WAITING FOR OUTPUT TO COMPLETE

  and waits for such a report to be completed before trying again to create the file. If there is no possibility of disk file space becoming available, the preceding DJDE packet is printed on an OPRINFO page together with the message:
NO SPACE ON DISK FOR FILE xxx.yyy.

Input then proceeds to read and discard the extent of the file.

The file's data records are read and stored into the appropriately named file until the total extent of the file is spanned. PCC bytes and block and record delimiters are ignored while processing the file data.

• If the DJDE parameter FILE=() signaled the start of LPS-labeled data, input presumes that more than one LPS-labeled file may be batched together and checks for another label record when it completes processing a file.

• If OPRINFO=YES and the file transfers are successfully concluded, the file name and file type of each file transferred are listed on the OPRINFO page with a brief indication of whether the file was newly created or replaced an existing file.

Delimited records for LPS-labeled files

To prevent loss of data through trailing blank suppression procedures on the host spooler, data records for LPS-labeled files must be delimited by appending an extra non-blank character at the end of each record. The same non-blank character must be used with all records processed in the file processing mode.

• If the non-blank character that is appended at the end of the records is not consistently present throughout the extent of a file, the preceding DJDE packet is printed on an OPRINFO page together with the message:

   FILE xxx.yyy PROCESSING ABORTED DUE TO DATA ERROR.

   Input then deletes the partial file that was created and proceeds to process the data as variable text data.

The Advanced Image Subsystem (AIS) enables a laser printing system to image encoded digitized graphic images in addition to other text and forms.

With RIP, Interpress, and graphic capabilities, more than 16 images on a page are allowed.

LPS-resident graphic software supports input from:

• Magnetic tapes with fixed or variable record/block structures, that is, RECORD STRUCTURE = F, FB, V, or VB only, in an unpacked format, which are compatible with use on the LPS.

• IBM online host systems equipped with the XPPI software package or equivalent.
9. Using graphics

Graphics considerations

There are several PDL parameters available for graphics handling. These include four JDE OUTPUT command parameters and six DJDEs.

Tables 9-1 and 9-2 summarize these parameters. The OUTPUT command parameters are described in the “OUTPUT command” section in chapter 6. The graphics DJDE parameters were discussed previously in this chapter 8.

Graphics DJDEs are record oriented because they take effect immediately. For example, ALTER, CANCEL, and IMAGE may appear anywhere on the page and are applied to the page on which they occur. Positioning GRAPHIC, BATCH, and SAVE DJDEs is discussed in table 9-2, “DJDE parameter definitions.” These also take effect immediately.

Page-oriented DJDEs to be applied to the next page must not precede batch mode or page interleaved graphics to be applied to the current page.

Note that your Xerox 4635 LPS can handle single-color graphics from a Xerox 4850 or 4890 HighLight Color LPS, but cannot process 2-color graphics.

Input for graphics

Named graphics may be input to the LPS through magnetic tape, the online channel, or Ethernet. They may be input either separately from or interleaved with textual data. If input separately, they are recorded as permanent disk files. If interleaved with textual data, they may or may not be recorded as permanent disk files. (Refer to the “Processing modes” section for a discussion of this operation.)

A graphic is typically incorporated into a document by a DJDE record occurring with the data targeted for the same simplex or duplex page on which the graphic appears. Additionally, a graphic may be referenced in a form, so that use of the form automatically includes the graphic.
Processing modes

A graphic is handled by the software in one of the following four modes:

- Batch
- Block
- Move
- Random

The user selects the mode depending upon how the graphic is made available for report processing, as well as user performance requirements and disk resources.

Random mode

In random mode, a graphic is obtained from a named disk file. This occurs if the file already exists at the start of report processing or is read from the report's input data stream before the first record of the report text is processed. In the latter case, each graphic is written to the disk as a permanent IMG file. In random mode, the output software reads the various graphics for a page from their individual files into graphics memory.

Move mode

In move mode, a graphic is obtained from a named disk file (just as in random mode), but the graphic is copied into the print file and used as in block mode. This feature may be invoked by the GRAPHICS command of the OUTPUT command. (Refer to chapter 6, “Specifying output parameters.”)

Note: Under some circumstances, move mode graphics are overridden and treated as random mode. In particular, if for any page there is exactly one graphic and that graphic is on a disk drive other than the print file's disk drive, random mode is invoked.

Block mode

In block mode, a graphic is obtained from the input data stream immediately following the textual data to be imaged on the same page (refer to the “Page interleaved” section of this chapter). The graphic is copied directly to the print file adjacent to the page's formatted textual information. Separate graphic disk files are not created. In block mode, the output software reads graphics from the print file into graphics memory. Block mode graphics exist on disk only in the print file and only while the report is being processed. No disk file is created.

A given page of a report which includes block mode graphics may also include either random mode graphics or move mode graphics.
**Batch mode**

In batch mode, the input stream contains graphics which are to be imaged, one per page, with either default or optional imaging specifications. Batch mode graphics are copied to the print file and processed the same as block mode graphics. In batch mode, only one graphic is imaged on each page, (for example, forms may not be invoked in batch mode). This feature may be invoked by the GRAPHICS parameter of the OUTPUT command.

**Tape formats**

**Noninterleaved**

To transfer IMG files from tape to an LPS system disk independent of printing, the COPY TAPE LABEL file-name command may be used.

**Document interleaved**

In this format, a report’s data stream begins with a set of one or more graphics prior to the first record of text. These graphics are copied one at a time to named IMG disk files. When the report has finished printing, the IMG files created are automatically deleted, unless the user has explicitly specified otherwise.

The rules for formatting a document interleaved graphic report tape file are:

- The tape format may be any legal fixed or variable length block, or record format which is otherwise supported by the LPS, that is, RECORD STRUCTURE = F, FB, V or VB. It may not be an undefined block or record format, that is, RECORD STRUCTURE = U or UB. Packed data formats are not supported.

- Use of block and record constants in an appropriately structured file (type F, FB, V, or VB) containing interleaved graphics is supported. However, the actual delimiter searching is suspended while an IMG file is being read.

  - For record delimiters, this means that delimiter searching is suspended from the first record of graphic data until the entire graphic has been read. Extranous bytes at the end of the last record are ignored. Record delimiter processing resumes with the next record.

  - For block delimiters, this means that the first byte of a graphic must be the first data byte of a block and that the first byte of data following the graphic must also be the first byte of a block. Extranous data in the last block following the last byte specified by the IMG byte count are ignored.

- Each graphic must be preceded immediately by a GRAPHIC DJDE.
• Improved tape-to-disk processing time is obtained if the graphic data portion of each record is an integer multiple of 512 bytes and begins at an even byte offset from the beginning of the block. Processing of each graphic begins in this mode. If a record is not a multiple of 512 bytes or does not begin at an even byte offset from the beginning of the block, the mode is terminated.

Page interleaved

In this format, one or more graphics imaged on a physical page are included in the data file immediately following the textual data for that page. These graphics, if referenced, are processed in block mode, that is, copied one at a time directly to the print file adjacent to the page’s formatted textual data. If not referenced, a page interleaved graphic is skipped. The rules for page interleaved graphics are as follows:

• The graphics must occur on the tape immediately following the last text to appear on the same physical page.

• Rules 1 through 4 for document interleaved graphics apply (refer to the “Document interleaved” section).

Batch mode

In this mode, a report contains only graphics imaged one per page and an optional IMAGE DJDE preceding each graphic. These graphics are copied to the print file and processed as block mode graphics.

Rules 1 and 4 for document interleaved graphics apply (refer to the “Document interleaved” section).

Note: Block or record delimiters are not permitted.

Except for the graphics themselves, the input stream may contain only optional DJDE IMAGE records. When a DJDE IMAGE record is encountered, it defines the scaling and positioning parameters for subsequent graphics.

Online formats

Noninterleaved

HOSTCOPY is the mechanism for transferring noninterleaved graphics through the online channel to disk. The HOSTCOPY utility supports IMG files in Xerox system tape format.

Document and page interleaved

Online report processing, which includes document and page interleaved graphics, functions as tape input processing functions. The rules for online interleaved graphics are:

• Each graphic must be preceded immediately by a DJDE GRAPHIC record.
• There is no change in the maximum record length when processing interleaved graphic records. However, every record of an online interleaved graphic must have an appended terminating character which is the same for all records of the graphic. If an interleaved graphic record is encountered which does not have the same terminating character as the first record of the graphic, the user has the option of aborting or continuing the report. If the report is continued, the graphic is unprintable and the results may be unpredictable.

• Document-interleaved graphics must begin immediately following the banner pages.

**Batch mode**

Online processing of batch mode reports, except for media-imposed differences, is the same as for magnetic tape. The input data stream format must comply with the following rules:

• Even though a JDE specifying batch mode is in effect, the first graphic must be preceded by a DJDE specifying BATCH=START. The last graphic must be followed immediately by a DJDE specifying BATCH=END. These batch stream delimiters are necessary to distinguish banner page data (header and trailer) from graphic data.

• Except for the graphics themselves, the input stream may contain only optional DJDE IMAGE records. When a DJDE IMAGE record is encountered, it defines scaling and positioning parameters for subsequent graphics. The actual batch mode data stream is defined by the BATCH=START and BATCH=END DJDEs.

• There is no change in the maximum record length when processing batch mode graphic records. However, every record within the graphic must be terminated by a byte containing an appended terminating character which is the same for all records of the graphic. If a batch mode graphic record is encountered which does not have the same terminating character as the first record of the graphic, the user has the option of aborting or continuing the report. If the report is continued, the graphic is unprintable and the results may be unpredictable.

**Document interleaved graphic file transfers**

The document interleaved format copies graphics down the channel from the host to LPS .IMG files. This is done by running a document interleaved report containing a trivial amount of text using a JDE whose OUTPUT command specifies graphics with PURGE=NO. This function is available for offline use also.

**Management of image files**

To produce print files consisting of interleaved textual and image data, the LPS file id must be saved in ASCII in the LPS 128-byte file label.
PDL command and DJDE options for graphics

Several PDL commands are available for graphic handling. These include four job descriptor entry (JDE) OUTPUT commands and six dynamic job descriptor entries (DJDEs).

Tables 9-1 and 9-2 summarize these commands.

The DJDEs are record oriented because they take effect immediately. For example, the ALTER, CANCEL, and IMAGE commands may appear anywhere on the page and are applied to the page on which they occur. The GRAPHIC, BATCH, and SAVE DJDE commands also take effect immediately.

Page-oriented DJDEs applied to the next page must not precede page interleaved graphics to be applied to the current page.

Table 9-1. OUTPUT command parameters for graphics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAPHICS</td>
<td>Specifies graphics may be used in current job.</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Specifies initial scaling and positioning parameters for batch mode.</td>
</tr>
<tr>
<td>PURGE</td>
<td>Specifies default disposition of document interleaved graphics.</td>
</tr>
<tr>
<td>UNITS</td>
<td>Specifies user-selected units for positioning graphics.</td>
</tr>
</tbody>
</table>

Table 9-2. DJDE definitions for graphics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>Specifies new imaging parameters for graphics previously referenced using HOLD statement.</td>
</tr>
<tr>
<td>BATCH</td>
<td>Marks beginning and end of online batch mode graphic data.</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Terminates option of the GRAPHICS DJDE.</td>
</tr>
<tr>
<td>GRAPHIC</td>
<td>Precedes interleaved graphic in data stream.</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Specifies name, scaling, and positioning of a graphic printed.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Identifies document interleaved graphics not to be deleted or purged.</td>
</tr>
</tbody>
</table>

Refer to chapter 8 for more information on DJDEs.
Performance

Random mode

Random mode requires a separate disk access by the output processing task for every graphic on a page. The more graphics on a page and the larger the graphics, the more likely it is that paper path holes are created.

Online

It should be noted that sending interleaved graphics across the online interface impacts system performance. The time to process graphic information across the channel, block it, and write it to disk is significant. As the average density of graphic information decreases, throughput should increase.

Document interleaved file creation

The time required to create (or replace) and subsequently purge document-interleaved graphics is significant.

Restrictions

This section summarizes graphic restrictions, some of which have been mentioned elsewhere in this document. These restrictions are grouped according to whether they affect graphic features.

Note: The restrictions which follow are applicable without RIP and Interpress.

Graphic feature restrictions

- Tape formats supported by interleaved graphic processing are restricted to LPS compatible fixed or variable record structures, that is, RECORD STRUCTURE = F, FB, V, or VB only. Undefined length record structures, such as types U and UB, are not supported. Packed data formats are not supported.
- The maximum number of graphics that may be imaged on one physical page is 16. This includes graphics associated with forms.
- Graphic images may be overlapped on a page with each other and with text.
- There is no allowance or “tolerance” available for the paper size in a JDE for a graphic to be off the page.
• No online banner page (neither header nor trailer) may include graphic references.

• A JDE which is selected through the DJDE option JDE= or JDL= may not specify GRAPHICS=NO if the current report has already processed at least one graphic.

• In online batch mode, PCC commands are ignored beginning with the BATCH=START DJDE until the BATCH=END DJDE.

• Graphics processing is not copy sensitive. A copy-sensitive form may not reference graphics.

• If RPAGE is used to move a logical page from one physical page to another, graphic references are handled as follows: Graphic references encountered after the record which satisfied the RPAGE statement are imaged on the physical page to which the logical page is moved. Any other image references are applied to the physical page being formed at the time the RPAGE statement was satisfied. For example, if RPAGE WHEN=TOP is specified, it is possible for a graphic to appear on physical page n even though text which may have preceded its reference in the input data stream appears on physical page n+1.

Use caution when using RPAGE with page interleaved graphics.

• A document-interleaved graphic replaces an existing graphic with the same name. This is true even if the existing graphic is used in a report which is being printed.

• Forms are not supported for batch mode processing.
A. PDL command and DJDE summary

This appendix may be used as a quick reference for information about PDL commands and command parameters. The following definitions describe the conventions used in this appendix.

Conventions

1. Symbols used in the command syntax are as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac</td>
<td>Command identifier. Consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be alpha.</td>
</tr>
<tr>
<td>dd</td>
<td>Command identifier. Same as ac except restriction of having one alpha character is removed. This identifier applies only to SYSTEM (or JDL) and JOB (or JDE) command sets.</td>
</tr>
<tr>
<td>id</td>
<td>Reference to a command identified by an ac command identifier.</td>
</tr>
<tr>
<td>sc</td>
<td>String constant. A hexadecimal, octal, ASCII, EBCDIC, or character constant.</td>
</tr>
<tr>
<td>value</td>
<td>A decimal constant.</td>
</tr>
<tr>
<td>s</td>
<td>The “s” at the end of some PDL commands and DJDEs is optional.</td>
</tr>
</tbody>
</table>

2. The LPS column indicates the laser printing system to which the command or parameter applies. ALL means that the command is applicable to all laser printing systems operating with version 3 software or higher.

3. The “Online” and “Offline” columns specify the input data stream modes of operation for the LPS. A Y for a particular parameter means the parameter is available for that processing mode. An N means the parameter is not available for that mode.

4. The DJDE column heading indicates whether the corresponding PDL command may also be used as a DJDE. If the column contains a Y, the DJDE parameter is available for both online and offline modes. An N means the parameter is not available as a DJDE.
5. The P or R column heading indicates whether the DJDE parameter is page oriented (P in the column) or record oriented (R in the column).
Table A-2. Summary of PDL commands and DJDE parameters

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNORMAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
<td>All</td>
<td>STOP</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>ISMATCH</td>
<td>4850</td>
<td>STOP</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>ISSTITUTE</td>
<td>4850</td>
<td>ANY</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>OTEXT</td>
<td>All</td>
<td>NOWAIT</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>SECURITY</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>ACCT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEPT</td>
<td>All</td>
<td>jdl-name</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>USER</td>
<td>All</td>
<td>BIN</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>BANNER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HCOUNT</td>
<td>All</td>
<td>0</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>HJOBNO</td>
<td>All</td>
<td>NONE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>HRPTNA</td>
<td>All</td>
<td>NONE</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>TCOUNT</td>
<td>All</td>
<td>0</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
<td>All</td>
<td>BANNER</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>BARCODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BSEQ</td>
<td>4635</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>BSIDE</td>
<td>4635</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>BSKIP</td>
<td>4635</td>
<td>BSKIP</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>BDELETE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BLOCK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ADJUST</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>CONSTANT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>FORMAT</td>
<td>All</td>
<td>BIN</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>LENGTH</td>
<td>All</td>
<td>1330</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>LMULT</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>LTHFLD</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>OFFSET</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>POSTAMBLE</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>PREAMBLE</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>ZERO</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>BSELECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Table A-2. **Summary of PDL commands and DJDE parameters (continued)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac:CATALOG</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPORT</td>
<td>SEPARATORS</td>
<td>4635</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NUMBER</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>SPLIT</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>SRECOVER</td>
<td>4635</td>
<td>PAGE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>STIMING</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>ac:CME</td>
<td>CONSTANT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FONTS</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INKS</td>
<td>4850</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINE</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POSITION</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac:CODE</td>
<td>ASSIGN</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>All</td>
<td>EBCDIC</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac:CRITERIA</td>
<td>CHANGE</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSTANT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINENUM</td>
<td>All</td>
<td>all lines</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>All</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td>All</td>
<td>no default</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>IDEN</td>
<td>OFFSET</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OPRINFO</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PREFIX</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SKIP</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Parameter</td>
<td>LPS</td>
<td>Default</td>
<td>Off-line</td>
<td>On-line</td>
<td>DJDE</td>
<td>Page- or record-oriented</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----------</td>
<td>----------</td>
<td>---------</td>
<td>------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>IDR (4850 only)</td>
<td>ICATALOG</td>
<td>4850</td>
<td>SDLT$ (system default ink)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>ILIST</td>
<td>4850</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>PALETTES</td>
<td>4850</td>
<td>DEFAULT</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>dd:jDE</td>
<td>INCLUDE</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>dd:jDL</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>dd:jOB</td>
<td>INCLUDE</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LINE</td>
<td>DATA</td>
<td>All</td>
<td>offline=1,132 online=0,150</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>FCB</td>
<td>All</td>
<td>PROCESS</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FONTINDEX</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>INDEX</td>
<td>4850</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>MARGIN</td>
<td>All</td>
<td>(1,POS)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>OVERPRINT</td>
<td>All</td>
<td>(PRINT,NODISP)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>PCC</td>
<td>All</td>
<td>(0,NOTRAN)</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCCTYPE</td>
<td>All</td>
<td>offline=ANSI online=IBM 3211 or IBM 4245</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCSB</td>
<td>All</td>
<td>PROCESS</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VFU</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>MESSAGE</td>
<td>ITEXT</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>OTEXT</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>BTEXT</td>
<td>4635</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>BFORMS</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>COLLATE</td>
<td>All</td>
<td>YES</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>COPIES</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>COVERS</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CYCLEFORMS</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
Table A-2. Summary of PDL commands and DJDE parameters (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSITY</td>
<td>All</td>
<td>DEFAULT</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESTINATION</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>DUPLEX</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>FACEUP</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEED</td>
<td>All</td>
<td>OPR</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>FORMATS</td>
<td>All</td>
<td>FMT1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>FORMS</td>
<td>All</td>
<td>• NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• init defaults to first or next copy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• FOR defaults to FORMAT if FORMS not fully keyed in</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAPHICS</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDFAULT</td>
<td>4850</td>
<td>First ink in ILIST parameter</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>IDR</td>
<td>4850</td>
<td>DFIDR.IDR</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>IMAGES</td>
<td>All</td>
<td>inches, top left corner scaled at 1/1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>INVERT</td>
<td>4635</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>IRESULT</td>
<td>4850</td>
<td>Sysgened default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>MODIFY</td>
<td>All</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>NTO1</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>OFFSET</td>
<td>All</td>
<td>ALL</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSTK</td>
<td>4635</td>
<td>RES</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOD and POS default=1</td>
<td>SIZE=.5 inch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAPERSIZES</td>
<td>All</td>
<td>sysgened default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PURGE</td>
<td>All</td>
<td>YES</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOLUTION</td>
<td>All</td>
<td>sysgened default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHIFT</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>SF1FUNCTION</td>
<td>4635</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SF2FUNCTION</td>
<td>4635</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZING</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAPLE</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOCKS</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Parameter</td>
<td>LPS</td>
<td>Default</td>
<td>Offline</td>
<td>Online</td>
<td>DJDE</td>
<td>Page- or record-oriented</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>--------</td>
<td>------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><strong>OUTPUT</strong> (continued)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSPPR</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMODE</td>
<td>4635</td>
<td>depends on papersize, default value is inches</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>TRANS</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>UNITS</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XMP</td>
<td>4850</td>
<td>DEFAULT</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>XSHIFT</td>
<td>4635</td>
<td>NONE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td><strong>ac:PCC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADVTAPE</td>
<td>All</td>
<td>YES</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFAULT</td>
<td>All</td>
<td>DEFAULT</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITIAL</td>
<td>All</td>
<td>TOF</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MASK</td>
<td>All</td>
<td>X'FF'</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ac:PDE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEGIN</td>
<td>All</td>
<td>(.18 IN, .66 IN)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>FONTS</td>
<td>All</td>
<td>L0112B</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>PMODE</td>
<td>All</td>
<td>LANDSCAPE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td><strong>RAUX</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RDELETE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RBAR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>4635</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RECORD</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJUST</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTANT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAT</td>
<td>All</td>
<td>BIN</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>All</td>
<td>offline=133 online=150</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMULT</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTHFLD</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFFSET</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSTAMBLE</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREAMBLE</td>
<td>All</td>
<td>0</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRUCTURE</td>
<td>All</td>
<td>FB</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A-2. **Summary of PDL commands and DJDE parameters (continued)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFEED (4635 only)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>4635</td>
<td>no default for clu-def</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROFFSET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASSES</td>
<td>ALL</td>
<td>ALL</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac: ROUTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFORM</td>
<td>All</td>
<td>NONE</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>RTEXT</td>
<td>All</td>
<td>NONE</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>RPAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIDE</td>
<td>All</td>
<td>(NUFRONT, NOFFSET)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>WHEN</td>
<td>All</td>
<td>TOP</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>RRESUME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEGIN</td>
<td>All</td>
<td>NEXT</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>RSELECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>RSTACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCTINFO</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DELIMITER</td>
<td>All</td>
<td>NO</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>HRPTNA</td>
<td>All</td>
<td>NONE</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PRINT</td>
<td>All</td>
<td>NONE</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>RSUSPEND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEGIN</td>
<td>All</td>
<td>NEXT</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>All</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac: SEFFNT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEFMAP</td>
<td>4635</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>SEFFNT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAP</td>
<td>4635</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
<tr>
<td>SEFMAP</td>
<td>4635</td>
<td>no default</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
</tr>
</tbody>
</table>
Table A-2. Summary of PDL commands and DJDE parameters (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac:STOCKSET</td>
<td>ASSIGN</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INFEED</td>
<td>All</td>
<td>first stock-name</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSPAGE</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>dd:SYSTEM</td>
<td></td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac:TABLE</td>
<td>CONSTANT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MASK</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac:TCODE</td>
<td>DEFAULT</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TASSIGN</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRESET</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ac:VFU</td>
<td>ASSIGN</td>
<td>All</td>
<td>no default</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>BOF</td>
<td>All</td>
<td>66</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>TOF</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>R</td>
</tr>
<tr>
<td>VOLUME</td>
<td>BMULT</td>
<td>All</td>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CODE</td>
<td>All</td>
<td>EBCDIC</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EOV</td>
<td>All</td>
<td>(NOPAUSE, NO_EOF)</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOST</td>
<td>All</td>
<td>IBMOS</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INTERPRESS</td>
<td>All</td>
<td>YES NO_CHECKSUM NO_BREAKPAGE SIMPLE INTEGRAL PERFORMANCE RIP</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LABEL</td>
<td>All</td>
<td>STANDARD</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCODE</td>
<td>All</td>
<td>EBCDIC</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LPACK</td>
<td>All</td>
<td>NO</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAXLAB</td>
<td>All</td>
<td>81</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MINLAB</td>
<td>All</td>
<td>80</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OPTIMIZE</td>
<td>All</td>
<td>NONE</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>
### Table A-2. Summary of PDL commands and DJDE parameters (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>LPS</th>
<th>Default</th>
<th>Off-line</th>
<th>On-line</th>
<th>DJDE</th>
<th>Page- or record-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLUME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(continued)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSCHN</td>
<td>All</td>
<td>9</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSHDP</td>
<td>All</td>
<td>0</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSTLP</td>
<td>All</td>
<td>0</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLABEL</td>
<td>All</td>
<td>NO</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMULT</td>
<td>All</td>
<td>1</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSAT</td>
<td>All</td>
<td>REMOUNT</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCODE</td>
<td>All</td>
<td>EBCDIC</td>
<td>Y Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNPACK</td>
<td>All</td>
<td>NONE</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
B. PDL command quick reference

This appendix may be used as a quick reference for information on all PDL command parameter and parameter option syntax. This appendix applies to all Xerox laser printing systems; LPS-specific commands, parameters, and options are indicated in parentheses. Refer to your specific laser printing system PDL/DJDE reference for information on commands not applicable to your 4635 LPS. Table B-1 describes the conventions used in this appendix.

Conventions

Table B-1. Symbols used in command syntax

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac</td>
<td>Command identifier. Consists of 1 to 6 alphanumeric characters (A through Z and 0 through 9). One of the characters must be alpha.</td>
</tr>
<tr>
<td>dd</td>
<td>Command identifier. Same as ac except restriction of having one alpha character is removed. This identifier applies only to SYSTEM (or JDL) and JOB (or JDE) command sets.</td>
</tr>
<tr>
<td>id</td>
<td>Reference to a command identified by an ac command identifier.</td>
</tr>
<tr>
<td>sc</td>
<td>String constant. A hexadecimal, octal, ASCII, EBCDIC, or character constant.</td>
</tr>
<tr>
<td>value</td>
<td>A decimal constant.</td>
</tr>
<tr>
<td>italics</td>
<td>Indicate variable names or numbers.</td>
</tr>
<tr>
<td>underlining</td>
<td>Indicates the default parameter or parameter option.</td>
</tr>
<tr>
<td>braces</td>
<td>( ) Indicate required options.</td>
</tr>
<tr>
<td>brackets</td>
<td>([ ]) Indicate optional choices.</td>
</tr>
<tr>
<td>parenthesis</td>
<td>( ) Used when more than one parameter option is listed after a parameter.</td>
</tr>
<tr>
<td>. . .</td>
<td>(Ellipsis) Indicates repetition of a parameter option or a series of parameter options.</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>CONTENT</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>ABNORMAL ERROR =</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>ABNORMAL IMISMATCH =</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>ABNORMAL ISUBSTITUTE =</td>
<td>ANY</td>
</tr>
<tr>
<td>ABNORMAL IMISMATCH =</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>ABNORMAL OTEXT =</td>
<td>WAIT</td>
</tr>
<tr>
<td>ABNORMAL SECURITY =</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCT DEPT = sc</td>
<td>jdl name</td>
<td></td>
</tr>
<tr>
<td>ACCT USER =</td>
<td>BIN</td>
<td>BIN</td>
</tr>
<tr>
<td></td>
<td>TRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOTH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANNER HCOUNT = value</td>
<td>test-exp</td>
<td></td>
</tr>
<tr>
<td>BANNER HJOBNO =</td>
<td>(offset, length)</td>
<td>NONE</td>
</tr>
</tbody>
</table>
### Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BANNER continued</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| BANNER HRPTNA = { (offset, length) 
  NONE } | NONE |               |
| BANNER TCOUNT = value | 0 |               |
| BANNER TEST = test-ex | No default |               |
| BANNER TYPE = { DATE 
  BANNER } | BANNER |               |
| **BARCODE** |                                                                         |               |
| BARCODE = [ BSKIP = READ / SKIP, 
  BSDIE = ODD / EVEN / BOTH, 
  BSEQ = RESET / NO RESET / NO SEQ ] | BSKIP |               |
| **BDELETE** |                                                                         |               |
| BDELETE TEST = test-exp | No default |               |
| **BLOCK** |                                                                         |               |
| BLOCK ADJUST = value | 0 |               |
| BLOCK CONSTANT = sc | No default |               |
| BLOCK FORMAT = type | BIN |               |
| BLOCK LENGTH = value | 1330 |               |
| BLOCK LMULT = value | 1 |               |
| BLOCK LTHFLD = value | 0 |               |
| BLOCK OFFSET = value | 0 |               |
Table B-2. **PDL command quick reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLOCK continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLOCK POSTAMBLE=value</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>BLOCK ZERO = YES</td>
<td>NO</td>
</tr>
<tr>
<td><strong>BESELECT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BESELECT TEST=test-exp</td>
<td>No default</td>
</tr>
<tr>
<td><strong>ac:CATALOG</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>catalog-name: CATALOG;</td>
<td>No default</td>
</tr>
<tr>
<td><strong>ac:CME</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ac:CME CONSTANT=sc</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME FONT=value</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME LINE= { n }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME LINE= { (n,m) }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME LINE= { (n,-) }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME POSITION = n</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CME INK = { inkindex }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>CME INK = { 0 }</td>
<td>No default</td>
</tr>
<tr>
<td><strong>4850 command only</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CME INK = { inkindex }</td>
<td>No default</td>
</tr>
<tr>
<td><strong>ac:CODE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ac:CODE ASSIGN = { (input, output) }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>ac:CODE ASSIGN = { (input, (output,...,output)) }</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>ac:CODE DEFAULT = code-type</td>
<td>EBCDIC</td>
</tr>
<tr>
<td><strong>ac:CRITERIA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ac:CRITERIA CHANGE = (offset, length, NE, LAST)</td>
<td>No default</td>
</tr>
</tbody>
</table>
### Table B-2. **PDL command quick reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ac:CRITERIA cont.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTANT=</td>
<td>(offset, length, {EQ}, tab-id)</td>
<td>No default</td>
</tr>
<tr>
<td>ac:CODE LINENUM</td>
<td>(int, count)</td>
<td>All lines</td>
</tr>
<tr>
<td><strong>END</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END;</td>
<td></td>
<td>No default</td>
</tr>
<tr>
<td><strong>EXPORT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEPARATORS=</td>
<td>{FIRST / (LAST [,D]) / (BOTH [,D]) / NONE}</td>
<td>NONE</td>
</tr>
<tr>
<td>SN UMBER=</td>
<td>(lnum, cnum[,findex])</td>
<td>No default</td>
</tr>
<tr>
<td>SPLIT=</td>
<td>{NOW / (min, max) / OFF}</td>
<td>No default</td>
</tr>
<tr>
<td>SRECOVER=</td>
<td>{PAGE / SEGMENT / ASK / DEVICE}</td>
<td>PAGE</td>
</tr>
<tr>
<td>STIM ING=</td>
<td>{INTERNAL, sec) / (DELAY, sec) / (INTERVAL, sec, DELAY, sec)\</td>
<td>No default</td>
</tr>
<tr>
<td><strong>IDEN</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDEN OFFSET=</td>
<td>value</td>
<td>0</td>
</tr>
<tr>
<td>IDEN OPRINFO=</td>
<td>{YES }</td>
<td>NO</td>
</tr>
<tr>
<td>IDEN PREFIX=</td>
<td>sc</td>
<td>No default</td>
</tr>
<tr>
<td>IDEN SKIP=</td>
<td>value</td>
<td>1</td>
</tr>
<tr>
<td><strong>IDR (4850 LPS only)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDR ICATALOG=</td>
<td>ink-catalog-name</td>
<td>SDLT$ (system default ink)</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
<td>Default</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td><strong>IDR (4850 LPS only) continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDR ILIST=</td>
<td>'ink-name' {'ink-name',[['ink-name'],[...]]}</td>
<td>No default</td>
</tr>
<tr>
<td>IDR PALETTE=</td>
<td>'palette-name'</td>
<td>DEFAULT</td>
</tr>
<tr>
<td><strong>dd:jDE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[INCLUDE=</td>
<td>(catalog-id_1[,catalog-id_2][,...])]</td>
<td>No default</td>
</tr>
<tr>
<td><strong>dd:jDL</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDL=</td>
<td>jdl-id</td>
<td>No default</td>
</tr>
<tr>
<td><strong>dd:jOB</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[INCLUDE=</td>
<td>(catalog-id_1[,catalog-id_2][,...])]</td>
<td>No default</td>
</tr>
<tr>
<td><strong>LINE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE DATA=</td>
<td>(pdo, length)</td>
<td>Offline: 1,134</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online: 0, 150</td>
</tr>
<tr>
<td>LINE FCB=</td>
<td>IGNORE {PROCESS}</td>
<td>PROCESS</td>
</tr>
<tr>
<td>LINE FONTINDEX=</td>
<td>{offset ([,init-val [,bit-opt]])}</td>
<td>NONE</td>
</tr>
<tr>
<td>LINE INKINDEX=</td>
<td>(offset [,init-val [,bit-opt]])</td>
<td>NONE</td>
</tr>
<tr>
<td>LINE MARGIN=</td>
<td>{value {value,value-type}}</td>
<td>(1,POS)</td>
</tr>
<tr>
<td>LINE OVERPRINT=</td>
<td>(over-opt, disp)</td>
<td>(PRINT, NODISP)</td>
</tr>
</tbody>
</table>
### Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LINE continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINE PCC=</td>
<td>(offset, trans-type)</td>
<td>(0, NOTRAN)</td>
</tr>
</tbody>
</table>
| LINE PCCTYPE= | pcc-name | Offline: ANSI  
Online: IBM 3211 |
| LINE UCSB= | \{IGNORE \} \{PROCESS \} | PROCESS |
| LINE VFU= | \{vfu-id \} \{NONE \} | NONE |
| **MESSAGE** | | |
| MESSAGE ITEXT= | \{sc \} \{sc,passnum \} \{NONE \} | NONE |
| MESSAGE OTEXT= | \{sc \} \{sc[,passnum][,WAIT] \} \{sc[,END][,WAIT] \} \{NONE \} | NONE |
| MESSAGE BTEXT= | '\[RNA=xxx][,cjin=sss][,dpt=xxx] \[,LDT=/U/W/BW/SM/M/O/0..365/yy-mm-dd][,RSQ=Y/N][,DDE=Y/N][,DAR=SUM-WMGT/SUM-OPLOG/SUM-BOTH/DET-WMGT/DET-OPLOG/DET-BOTH][,RRA=nnn][,NSE=sss][,PRD=yy-mm-dd][,SEQ=sss][,PRA=nnn][,TXT=xx ..x]' | NONE |
| **OUTPUT** | | |
| OUTPUT BFORM= | \{form-id \} \{form-id[,init [,copies]][,(INK[S],inkref[,inkref][,...]) \} | NONE |
| OUTPUT COLLATE= | \{YES \} \{NO \} | YES |
| OUTPUT COPIES= | number | 1 |
Table B-2. **PDL command quick reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OUTPUT continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT COVER=</td>
<td>cover-opt</td>
<td>NONE</td>
</tr>
<tr>
<td>OUTPUT CYCLEFORMS=</td>
<td>(form-id[,form-id,...), (form-id[(INK[S],inkref[,inkref],...)]), (form-id[,INK[S],inkref[,inkref]...]])</td>
<td>NONE</td>
</tr>
<tr>
<td>OUTPUT DENSITY=</td>
<td>{FIX, NO,FIX, DEFAULT}</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>OUTPUT DESTINATION=</td>
<td>{BIN, TRAY, EXPORT}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT DUPLEX=</td>
<td>{YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT FACEUP=</td>
<td>{YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT FEED=</td>
<td>{stock-reference, stock-name, MAIN, AUX, OPR}</td>
<td>OPR</td>
</tr>
<tr>
<td>OUTPUT FORMAT=</td>
<td>pde-id</td>
<td>FMT1</td>
</tr>
<tr>
<td>OUTPUT FORMS=</td>
<td>form-id ([init [,copies]](INK[S], inktref [,inkref]...) )</td>
<td>• NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• init defaults to first or next copy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• FOR defaults to FORMAT if FORM[S] is not fully keyed in.</td>
</tr>
</tbody>
</table>
Table B-2. **PDL command quick reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OUTPUT continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT GRAPHICS=</td>
<td><img src="image" alt="OUTPUT GRAPHICS= Syntax" /></td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT IDFAULT=</td>
<td><code>inkref</code></td>
<td>First ink in ILIST parameter.</td>
</tr>
<tr>
<td>OUTPUT IDR=</td>
<td><code>idr-name</code></td>
<td><code>DFIDR.IDR</code></td>
</tr>
<tr>
<td>OUTPUT IMAGE</td>
<td><img src="image" alt="OUTPUT IMAGE Syntax" /></td>
<td>Inches, top left corner scaled at 1/1.</td>
</tr>
<tr>
<td>OUTPUT INVERT=</td>
<td><img src="image" alt="OUTPUT INVERT= Syntax" /></td>
<td></td>
</tr>
<tr>
<td>OUTPUT IRESULT=</td>
<td><img src="image" alt="OUTPUT IRESULT= Syntax" /></td>
<td>Sysgened default</td>
</tr>
<tr>
<td>OUTPUT MODIFY=</td>
<td><code>cme-id</code></td>
<td>NONE</td>
</tr>
<tr>
<td>OUTPUT NTO1=</td>
<td><img src="image" alt="OUTPUT NTO1= Syntax" /></td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT NUMBER=</td>
<td><img src="image" alt="OUTPUT NUMBER= Syntax" /></td>
<td>NO</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
<td>Default</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>OUTPUT continued</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT O STK=</td>
<td>(clu-def,[mod],[pos],[RES/NORES], [TAB],[size DOTS/IN/CM],[L/P])</td>
<td></td>
</tr>
<tr>
<td>OUTPUT OFFSET=</td>
<td>{ALL, FIRST, NONE}</td>
<td>ALL</td>
</tr>
<tr>
<td>OUTPUT PAPERSIZE=</td>
<td>{A3, A4, B4, B5, USLEGAL, USLETTER, (x,y)}</td>
<td>Sysgened default</td>
</tr>
<tr>
<td>OUTPUT PURGE=</td>
<td>{NO, YES}</td>
<td>YES</td>
</tr>
<tr>
<td>OUTPUT RESOLUTION</td>
<td>= [300 \text{ or } 600]</td>
<td>Sysgened default</td>
</tr>
<tr>
<td>OUTPUT SHIFT=</td>
<td>{(v1, v2), YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT SF1FUNCTION=</td>
<td>{YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT SF2FUNCTION=</td>
<td>{YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT SIZING=</td>
<td>{SEMAUTO, BEST, EXACT}</td>
<td>SEMIAUTO</td>
</tr>
<tr>
<td>OUTPUT STOCKS=</td>
<td>stockset-name</td>
<td>No default</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>OUTPUT continued</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT SYSPR= U</td>
<td>USLEG AL A3 A4 B4 B5 USLETTER (x,y)</td>
<td>Papersize of the last sheet that is before the system page.</td>
</tr>
<tr>
<td>OUTPUT TMODE=</td>
<td>width (width[,width until])</td>
<td></td>
</tr>
<tr>
<td>OUTPUT TRANS=</td>
<td>YES NO</td>
<td>NO</td>
</tr>
<tr>
<td>OUTPUT UNITS=</td>
<td>size in dots</td>
<td>No default</td>
</tr>
<tr>
<td>OUTPUT XMP=</td>
<td>DEFAULT REPORT</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>OUTPUT XSHIFT=</td>
<td>NO value</td>
<td>NO</td>
</tr>
<tr>
<td>ac:PCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ac:PCC ADVTAPE=</td>
<td>YES NO</td>
<td>YES</td>
</tr>
<tr>
<td>ac:PCC ASSIGN=</td>
<td>(byte,ccln) (byte, ccln[,ccln][,...])</td>
<td>No default</td>
</tr>
<tr>
<td>ac:PCC DEFAULT=</td>
<td>ccIn pctype</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
<td>Default</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>ac:PCC continued</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ac:PCC INITIAL=</td>
<td>BOF, TOF</td>
<td>TOF</td>
</tr>
<tr>
<td>ac:PCC MASK=</td>
<td>value</td>
<td>X'FF'</td>
</tr>
<tr>
<td>ac:PDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ac:PDE BEGIN=</td>
<td>(vpos, hpos)</td>
<td>(.18 IN, .66 IN)</td>
</tr>
<tr>
<td>ac:PDE FONTS=</td>
<td>(f1, [f2][...])</td>
<td>L0112B</td>
</tr>
<tr>
<td></td>
<td>((f1,s1), [f2,s2])[,....])</td>
<td></td>
</tr>
<tr>
<td>ac:PDE PMODE=</td>
<td>LANDSCAPE, PORTRAIT</td>
<td>LANDSCAPE</td>
</tr>
<tr>
<td>RAUX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAUX TEST=</td>
<td>test-exp</td>
<td>No default</td>
</tr>
<tr>
<td>RBAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBAR TEST=</td>
<td>(CONDITION1, [OR/AND, CONDITION2])</td>
<td>No default</td>
</tr>
<tr>
<td>RDDELETE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDDELETE TEST=</td>
<td>test-exp</td>
<td>No default</td>
</tr>
<tr>
<td>RECORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECORD ADJUST=</td>
<td>value</td>
<td>0</td>
</tr>
<tr>
<td>RECORD CONSTANT=</td>
<td>sc</td>
<td>No default</td>
</tr>
<tr>
<td>RECORD FORMAT=</td>
<td>type</td>
<td>BIN</td>
</tr>
</tbody>
</table>
### Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD</td>
<td>RECORD LENGTH = value</td>
<td>Offline=133</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online=150</td>
</tr>
<tr>
<td></td>
<td>RECORD L MULT = value</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>RECORD L TH FLD = size</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RECORD OFFSET = value</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RECORD POSTAMBLE = length</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RECORD PREAMBLE = length</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>RECORD STRUCTURE = structure-type</td>
<td>FB</td>
</tr>
<tr>
<td>RFEED</td>
<td>RFEED TEST = (test-exp,clu-def)</td>
<td>No default</td>
</tr>
<tr>
<td>ROFFSET</td>
<td>ROFFSET PASSES = {FIRST, ALL}</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>ROFFSET TEST = test-exp</td>
<td>No default</td>
</tr>
<tr>
<td>ac:ROUTE</td>
<td>ac:ROUTE RFORM = {form-id, NONE}</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td>ac:ROUTE RTEXT = {sc,{passnum/ALL,\line,\col,\fontindex}}</td>
<td>NONE</td>
</tr>
</tbody>
</table>
Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RPAGE</strong></td>
<td><strong>RPAGE SIDE=</strong> {side-opt, offset-opt}</td>
<td>(NUFRONT, NO OFFSET)</td>
</tr>
<tr>
<td></td>
<td><strong>RPAGE TEST=</strong> test-exp</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td><strong>RPAGE WHEN=</strong> {BOTTOM, NOW, TOP}</td>
<td>TOP</td>
</tr>
<tr>
<td><strong>RRESUME</strong></td>
<td><strong>RRESUME BEGIN=</strong> {CURRENT, NEXT}</td>
<td>NEXT</td>
</tr>
<tr>
<td></td>
<td><strong>RRESUME TEST=</strong> test-exp</td>
<td>No default</td>
</tr>
<tr>
<td><strong>RESELECT</strong></td>
<td><strong>RSELECT TEST=</strong> test-exp</td>
<td>No default</td>
</tr>
<tr>
<td><strong>RSTACK</strong></td>
<td><strong>RSTACK ACCINFO=</strong> (offset, length)</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td><strong>RSTACK DELIMETER=</strong> {YES, NO}</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td><strong>RSTACK HRPTNA=</strong> {offset, length}</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td><strong>RSTACK PRINT=</strong> {BIN, BOTH, TRAY, NONE}</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td><strong>RSTACK TEST=</strong> test-exp</td>
<td>No default</td>
</tr>
<tr>
<td>Command</td>
<td>Syntax</td>
<td>Default</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>RSUSPEND</td>
<td>RSUSPEND BEGIN { CURRENT \ NEX T }</td>
<td>NEXT</td>
</tr>
<tr>
<td></td>
<td>RSUSPEND TEST= test-exp</td>
<td>No default</td>
</tr>
<tr>
<td>ac:SEFFNT</td>
<td>SEFMAP= ((font\textsubscript{1}, font\textsubscript{2},), [(font\textsubscript{m},font\textsubscript{n})])</td>
<td>No default</td>
</tr>
<tr>
<td>SEFFNT</td>
<td>SEFMAP= ((font\textsubscript{1}, font\textsubscript{2},), [(font\textsubscript{m},font\textsubscript{n})]) \ \text{NONE}</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>MAP file-name</td>
<td>No default</td>
</tr>
<tr>
<td>ac:STOCKSET</td>
<td>ASSIGN= (stock-descriptor\textsubscript{1[,stock-descriptor\textsubscript{2[,...,]}})</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>ac:STOCKSET INIFEED= stock-name stock-reference</td>
<td>first stock name</td>
</tr>
<tr>
<td></td>
<td>ac:STOCKSET SYSPAGE= stock-name stock-reference \ MAIN \ AUX</td>
<td>No default</td>
</tr>
<tr>
<td>dd:SYSTEM</td>
<td>Missing syntax?</td>
<td>No default</td>
</tr>
<tr>
<td>ac:TABLE</td>
<td>ac:TABLE CONSTANT= (sc\textsubscript{1[,sc\textsubscript{2[,...,]}})</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>MASK= (ignore-char([,charspec\textsubscript{1}, charspec\textsubscript{2}][,...]))</td>
<td>No default</td>
</tr>
</tbody>
</table>
Table B-2. **PDL command quick reference**

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TCODE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFAULT=</td>
<td>tcode-type</td>
<td>No default</td>
</tr>
<tr>
<td>TASSIGN=</td>
<td>(typespec,</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>'inputspec'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'{inputspec'2[,...]}</td>
<td></td>
</tr>
<tr>
<td>TRESET=</td>
<td>(typespec,</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>'inputspec'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'{inputspec'1[,...]}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td><strong>ac:VFU</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VFU ASSIGN=</td>
<td>(channo, lineno)</td>
<td>No default</td>
</tr>
<tr>
<td></td>
<td>channo, lineno[,lineno][,...])</td>
<td></td>
</tr>
<tr>
<td>VFU BOF=</td>
<td>value</td>
<td>66</td>
</tr>
<tr>
<td>VFU TOF=</td>
<td>value</td>
<td>1</td>
</tr>
<tr>
<td><strong>VOLUME</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLUME BMULT=</td>
<td>value</td>
<td>1</td>
</tr>
<tr>
<td>VOLUME CODE=</td>
<td>keyword</td>
<td>EBCDIC</td>
</tr>
<tr>
<td></td>
<td>id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>VOLUME EOF=</td>
<td>(PAUSE, EOF)</td>
<td>NO PAUSE,</td>
</tr>
<tr>
<td></td>
<td>(N O PAUSE, NO EOF)</td>
<td>NO EOF</td>
</tr>
<tr>
<td>VOLUME HOST=</td>
<td>host-type</td>
<td>IBMOS</td>
</tr>
<tr>
<td>VOLUME INTERPRESS=</td>
<td>(YES)[CHECKSUM/NO CHECKSUM][,BREAKPAGE/NO BREAKPAGE][,ROBUST/SIMPLE][,INTEGRAL/FRACTION][,PERFORMANCE/C O M PlIA NCE][,RIP/NORIP])</td>
<td></td>
</tr>
</tbody>
</table>

B–16 XEROX 4635 LASER PRINTING SYSTEM PDL/DJDE REFERENCE
### Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VOLUME continued</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLUME LABEL=</td>
<td>label-type</td>
<td>STANDARD</td>
</tr>
<tr>
<td>VOLUME LCODE=</td>
<td>{ keyword [ id ] }</td>
<td>EBCDIC</td>
</tr>
<tr>
<td>VOLUME LPACK=</td>
<td>{ YES [ NO ] }</td>
<td>NO</td>
</tr>
<tr>
<td>VOLUME MAXLAB=</td>
<td>value</td>
<td>81</td>
</tr>
<tr>
<td>VOLUME MINLAB=</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>VOLUME OPTIMIZE=</td>
<td>{ keyword [ keyword[,...] ] }</td>
<td>NONE</td>
</tr>
<tr>
<td>VOLUME O SCHN=</td>
<td>value</td>
<td>9</td>
</tr>
<tr>
<td>VOLUME O SHDP=</td>
<td>value</td>
<td>0</td>
</tr>
<tr>
<td>VOLUME O STLP=</td>
<td>value</td>
<td>0</td>
</tr>
<tr>
<td>VOLUME PLABEL=</td>
<td>{ YES [ NO ] }</td>
<td>NO</td>
</tr>
<tr>
<td>VOLUME RMULT=</td>
<td>value</td>
<td>1</td>
</tr>
<tr>
<td>VOLUME RSAT=</td>
<td>{ SPLIT [ REMOUNT ] }</td>
<td>REMOUNT</td>
</tr>
<tr>
<td>VOLUME TCODE=</td>
<td>{ tcode-type [ tcode-id ] }</td>
<td>EBCDIC</td>
</tr>
</tbody>
</table>
### Table B-2. PDL command quick reference

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VOLUME continued</strong></td>
<td><strong>VOLUME UNPACK</strong> = { T4X3, T4X3H2, UNIVAC, NONE }</td>
<td><strong>NONE</strong></td>
</tr>
</tbody>
</table>
C. Character code assignment tables

IBM BCD (6-bit representation)

Figure C-1. IBM BCD (6-bit representation)

<table>
<thead>
<tr>
<th>Octal (columns)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rows)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>Least Significant Bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>000</td>
<td>0</td>
<td>9</td>
<td>/</td>
<td>Z</td>
<td>J</td>
<td>R</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>K</td>
<td>B</td>
<td>&amp;</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>K</td>
<td>B</td>
<td>&amp;</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>3</td>
<td>#</td>
<td>T</td>
<td>,</td>
<td>L</td>
<td>S</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>4</td>
<td>@</td>
<td>U</td>
<td>%</td>
<td>M</td>
<td>*</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>5</td>
<td>'</td>
<td>V</td>
<td>_</td>
<td>N</td>
<td>)</td>
<td>E</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>6</td>
<td>=</td>
<td>W</td>
<td>&gt;</td>
<td>O</td>
<td>;</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>7</td>
<td>X</td>
<td>?</td>
<td>P</td>
<td>G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: 0'20' is the official blank character.

*Corresponds to BCD code set used by IBM users and defined by PDL command CODE = IBMBCD.
Honeywell 200/2000 BCD (6-bit representation)

Figure C-2. Honeywell 200/2000 BCD (6-bit representation)

<table>
<thead>
<tr>
<th>Octal (rows)</th>
<th>Most Significant Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Binary (columns)</td>
<td>000 001 010 011 100 101 110 111</td>
</tr>
<tr>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
</tr>
</tbody>
</table>

Notes: 0'15' is the official blank character; 0'77' is the padding character. *Corresponds to BCD code set used by Honeywell 200/2000 users and defined by PDL command CODE = H2BCD.
Honeywell 6000 BCD (6-bit representation)

Figure C-3.  Honeywell 6000 BCD (6-bit representation)

<table>
<thead>
<tr>
<th>Octal (columns)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>Least Significant Bits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>000</td>
<td>0</td>
<td>8</td>
<td>space</td>
<td>H</td>
<td>^</td>
<td>2</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>1</td>
<td>9</td>
<td>A</td>
<td>I</td>
<td>J</td>
<td>R</td>
<td>/</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>2</td>
<td>[</td>
<td>B</td>
<td>&amp;</td>
<td>K</td>
<td>-</td>
<td>S</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>3</td>
<td>#</td>
<td>C</td>
<td>.</td>
<td>L</td>
<td>$</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>4</td>
<td>@</td>
<td>D</td>
<td>]</td>
<td>M</td>
<td>*</td>
<td>U</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>5</td>
<td>:</td>
<td>E</td>
<td>(</td>
<td>N</td>
<td>)</td>
<td>V</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>6</td>
<td>&gt;</td>
<td>F</td>
<td>&lt;</td>
<td>O</td>
<td>;</td>
<td>W</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>7</td>
<td>?</td>
<td>G</td>
<td>\</td>
<td>P</td>
<td>'</td>
<td>X</td>
</tr>
</tbody>
</table>

Notes:
1. Usual BCD Character
2. Printer Character

Corresponds to BCD code set used by Honeywell 600/6000 series SSM tapes and defined by PDL command CODE = BCD or CODE = H6BCD.
### Fielddata translation

Figure C-4. Fielddata translation

<table>
<thead>
<tr>
<th>Octal (columns)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>0</td>
<td>@</td>
<td>C</td>
<td>K</td>
<td>S</td>
<td>)</td>
<td>*</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>[</td>
<td>D</td>
<td>L</td>
<td>T</td>
<td>-</td>
<td>(</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>]</td>
<td>E</td>
<td>M</td>
<td>U</td>
<td>+</td>
<td>%</td>
<td>2</td>
<td>'</td>
</tr>
<tr>
<td>3</td>
<td>#</td>
<td>F</td>
<td>N</td>
<td>V</td>
<td>&lt;</td>
<td>:</td>
<td>3</td>
<td>;</td>
</tr>
<tr>
<td>4</td>
<td>^</td>
<td>G</td>
<td>O</td>
<td>W</td>
<td>=</td>
<td>?</td>
<td>4</td>
<td>/</td>
</tr>
<tr>
<td>5</td>
<td>(blank)</td>
<td>H</td>
<td>P</td>
<td>X</td>
<td>&gt;</td>
<td>!</td>
<td>5</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>I</td>
<td>Q</td>
<td>Y</td>
<td>&amp;</td>
<td>,</td>
<td>6</td>
<td>”</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
<td>J</td>
<td>R</td>
<td>Z</td>
<td>$</td>
<td>\</td>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>
## UNIVAC ASCII character set

Figure C-5. **UNIVAC ASCII character set**

<table>
<thead>
<tr>
<th>Octal</th>
<th>Character</th>
<th>Octal</th>
<th>Character</th>
<th>Octal</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>040</td>
<td>(blank)</td>
<td>100</td>
<td>０</td>
<td>140</td>
<td>\</td>
</tr>
<tr>
<td>041</td>
<td>１</td>
<td>101</td>
<td>A</td>
<td>141</td>
<td>a</td>
</tr>
<tr>
<td>042</td>
<td>&quot;</td>
<td>102</td>
<td>B</td>
<td>142</td>
<td>b</td>
</tr>
<tr>
<td>043</td>
<td>#</td>
<td>103</td>
<td>C</td>
<td>143</td>
<td>c</td>
</tr>
<tr>
<td>044</td>
<td>$</td>
<td>104</td>
<td>D</td>
<td>144</td>
<td>d</td>
</tr>
<tr>
<td>045</td>
<td>%</td>
<td>105</td>
<td>E</td>
<td>145</td>
<td>e</td>
</tr>
<tr>
<td>046</td>
<td>&amp;</td>
<td>106</td>
<td>F</td>
<td>146</td>
<td>f</td>
</tr>
<tr>
<td>047</td>
<td>/</td>
<td>107</td>
<td>G</td>
<td>147</td>
<td>g</td>
</tr>
<tr>
<td>050</td>
<td>(</td>
<td>110</td>
<td>H</td>
<td>150</td>
<td>h</td>
</tr>
<tr>
<td>051</td>
<td>)</td>
<td>111</td>
<td>I</td>
<td>151</td>
<td>i</td>
</tr>
<tr>
<td>052</td>
<td>*</td>
<td>112</td>
<td>J</td>
<td>152</td>
<td>j</td>
</tr>
<tr>
<td>053</td>
<td>+</td>
<td>113</td>
<td>K</td>
<td>153</td>
<td>k</td>
</tr>
<tr>
<td>054</td>
<td>,</td>
<td>114</td>
<td>L</td>
<td>154</td>
<td>l</td>
</tr>
<tr>
<td>055</td>
<td>–</td>
<td>115</td>
<td>M</td>
<td>155</td>
<td>m</td>
</tr>
<tr>
<td>056</td>
<td>.</td>
<td>116</td>
<td>N</td>
<td>156</td>
<td>n</td>
</tr>
<tr>
<td>057</td>
<td>/</td>
<td>117</td>
<td>O</td>
<td>157</td>
<td>o</td>
</tr>
<tr>
<td>060</td>
<td>0</td>
<td>120</td>
<td>P</td>
<td>160</td>
<td>p</td>
</tr>
<tr>
<td>061</td>
<td>1</td>
<td>121</td>
<td>Q</td>
<td>161</td>
<td>q</td>
</tr>
<tr>
<td>062</td>
<td>2</td>
<td>122</td>
<td>R</td>
<td>162</td>
<td>r</td>
</tr>
<tr>
<td>063</td>
<td>3</td>
<td>123</td>
<td>S</td>
<td>163</td>
<td>s</td>
</tr>
<tr>
<td>064</td>
<td>4</td>
<td>124</td>
<td>T</td>
<td>164</td>
<td>t</td>
</tr>
<tr>
<td>065</td>
<td>5</td>
<td>125</td>
<td>U</td>
<td>165</td>
<td>u</td>
</tr>
<tr>
<td>068</td>
<td>6</td>
<td>126</td>
<td>V</td>
<td>166</td>
<td>v</td>
</tr>
<tr>
<td>067</td>
<td>7</td>
<td>127</td>
<td>W</td>
<td>167</td>
<td>w</td>
</tr>
<tr>
<td>070</td>
<td>8</td>
<td>130</td>
<td>X</td>
<td>170</td>
<td>x</td>
</tr>
<tr>
<td>071</td>
<td>9</td>
<td>131</td>
<td>Y</td>
<td>171</td>
<td>u</td>
</tr>
<tr>
<td>072</td>
<td>:</td>
<td>132</td>
<td>Z</td>
<td>172</td>
<td>z</td>
</tr>
<tr>
<td>073</td>
<td>;</td>
<td>133</td>
<td></td>
<td>173</td>
<td></td>
</tr>
<tr>
<td>074</td>
<td>&lt;</td>
<td>134</td>
<td></td>
<td>174</td>
<td></td>
</tr>
<tr>
<td>075</td>
<td>=</td>
<td>135</td>
<td></td>
<td>175</td>
<td></td>
</tr>
<tr>
<td>076</td>
<td>&gt;</td>
<td>136</td>
<td></td>
<td>176</td>
<td>~</td>
</tr>
<tr>
<td>077</td>
<td>?</td>
<td>137</td>
<td></td>
<td>177</td>
<td>(null)</td>
</tr>
</tbody>
</table>
### Standard ASCII character set

#### Figure C-6. Standard ASCII character set

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>MOST SIGNIFICANT BITS</th>
<th>LEAST SIGNIFICANT BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0000</td>
<td>0 @ P c p</td>
<td>Reserved for Xerox use only</td>
</tr>
<tr>
<td>1 0001</td>
<td>1 A Q a q</td>
<td></td>
</tr>
<tr>
<td>2 0010</td>
<td>&quot; 2 B R b r</td>
<td></td>
</tr>
<tr>
<td>3 0011</td>
<td># 3 C S c s</td>
<td></td>
</tr>
<tr>
<td>4 0100</td>
<td>$ 4 D T d t</td>
<td></td>
</tr>
<tr>
<td>5 0101</td>
<td>% 5 E U e u</td>
<td></td>
</tr>
<tr>
<td>6 0110</td>
<td>&amp; 6 F V f v</td>
<td></td>
</tr>
<tr>
<td>7 0111</td>
<td>' 7 G W g w</td>
<td></td>
</tr>
<tr>
<td>8 1000</td>
<td>( 8 H X h x</td>
<td></td>
</tr>
<tr>
<td>9 1001</td>
<td>) 9 I Y i y</td>
<td></td>
</tr>
<tr>
<td>A 1010</td>
<td></td>
<td>* : J Z j z</td>
</tr>
<tr>
<td>B 1011</td>
<td></td>
<td>+ : K I k</td>
</tr>
<tr>
<td>C 1100</td>
<td></td>
<td>, &lt; L \ I</td>
</tr>
<tr>
<td>D 1101</td>
<td></td>
<td>~ = M J m</td>
</tr>
<tr>
<td>E 1110</td>
<td></td>
<td>¼ . &gt; N ^ n</td>
</tr>
<tr>
<td>F 1111</td>
<td></td>
<td>÷ / ? O _ o</td>
</tr>
</tbody>
</table>
### Standard EBCDIC character set

**Figure C-7. Standard EBCDIC character set**

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>MOST SIGNIFICANT BITS</th>
<th>LEAST SIGNIFICANT BITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0000</td>
<td>&amp; - ¼ ½ {} \ 0</td>
<td></td>
</tr>
<tr>
<td>1 0001</td>
<td>/ a j A J 1</td>
<td></td>
</tr>
<tr>
<td>2 0010</td>
<td>b k s B K S 2</td>
<td></td>
</tr>
<tr>
<td>3 0011</td>
<td>c l t C L T 3</td>
<td></td>
</tr>
<tr>
<td>4 0100</td>
<td>d m u D M U 4</td>
<td></td>
</tr>
<tr>
<td>5 0101</td>
<td>e n v E N V 5</td>
<td></td>
</tr>
<tr>
<td>6 0110</td>
<td>f o w F O W 6</td>
<td></td>
</tr>
<tr>
<td>7 0111</td>
<td>g p x G P X 7</td>
<td></td>
</tr>
<tr>
<td>8 1000</td>
<td>h q y H Q Y 8</td>
<td></td>
</tr>
<tr>
<td>9 1001</td>
<td>i r z I R Z 9</td>
<td></td>
</tr>
<tr>
<td>A 1010</td>
<td>e ! ∧ : !</td>
<td></td>
</tr>
<tr>
<td>B 1011</td>
<td>. $ , # { }</td>
<td></td>
</tr>
<tr>
<td>C 1100</td>
<td>&lt; * % @</td>
<td></td>
</tr>
<tr>
<td>D 1101</td>
<td>( ) ' [ \ ]</td>
<td></td>
</tr>
<tr>
<td>E 1110</td>
<td>+ ; &gt; =</td>
<td></td>
</tr>
<tr>
<td>F 1111</td>
<td>! ~ ? &quot;</td>
<td></td>
</tr>
</tbody>
</table>
### Xerox EBCDIC to extended ASCII hexadecimal translation values

Figure C-8.

<table>
<thead>
<tr>
<th>Hexadecimal</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>DA</td>
<td>2D</td>
<td>26</td>
<td>2D</td>
<td>80</td>
<td>1E</td>
<td>1F</td>
<td>8C</td>
<td>8D</td>
<td>1A</td>
<td>1C</td>
<td>5C</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>FA</td>
<td>9A</td>
<td>D9</td>
<td>8A</td>
<td>9F</td>
<td>61</td>
<td>98</td>
<td>81</td>
<td>8A</td>
<td>41</td>
<td>4F</td>
<td>9F</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>A</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>C</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>D</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>E</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>F</td>
<td>20</td>
<td>FA</td>
<td>EA</td>
<td>8B</td>
<td>8B</td>
<td>AE</td>
<td>62</td>
<td>68</td>
<td>73</td>
<td>82</td>
<td>42</td>
<td>53</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>

Values on perimeter are EBCDIC. Values within blocked rectangle are ASCII. Example: EBCDIC value of AB is equivalent to an ASCII value of 7D.
**D. Offline specifications**

### Input unpacking examples

**Figure D-1. Input unpacking examples**

<table>
<thead>
<tr>
<th>BCD Character</th>
<th>Binary Representation</th>
<th>Represented on 9T Tape by</th>
<th>After Unpacking by OSS</th>
<th>After Translation to ASCII by OSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0100010000001001001</td>
<td>A 1 5 *</td>
<td>00010001 00000001 00000001 00000101 00101100</td>
<td>00100001 00110001 00110101 00110010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T4X3 Unpacking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BCD Character (HZBCD)</th>
<th>Binary Representation (HZBCD)</th>
<th>Represented on 9-Track Tape by</th>
<th>After Unpacking by OSS</th>
<th>After Translation to ASCII by OSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0100010000001001001</td>
<td>A 1 5 *</td>
<td>00010001 00000001 00000001 00000101 00101100</td>
<td>00100001 00110001 00110101 00110010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T4X3H2 Unpacking</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Valid host computer/label specification table

Table D-1. Valid host computer/label specifications

<table>
<thead>
<tr>
<th>Host type</th>
<th>Label specification</th>
<th>Unlabeled</th>
<th>ANSI</th>
<th>Standard</th>
<th>System print</th>
<th>COBOL</th>
<th>Undef</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMOS</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBMMDOS</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRASP</td>
<td>Label specification is ignored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWER</td>
<td>Label specification is ignored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OWSWR</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US70</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XEROX</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2500</td>
<td>Burroughs medium systems</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2700</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3500</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3700</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4700</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B6700</td>
<td>Burroughs large system</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2000</td>
<td>Honeywell 200/2000 series</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>H6000</td>
<td>Honeywell 600/6000 series</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>Label specification is ignored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCTDUMP</td>
<td>Label specification is ignored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIVAC</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANSI</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCR</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSX11</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICL2900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLDUMP</td>
<td>Label specification is ignored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undef</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Each shaded area of this table shows the label that is substituted by PDL when an invalid host/label pair is specified. Each X indicates a valid label specification parameter for each type of host.
## Host system JDLs on the system software tape

Job description library (JDL) source files supplied on the operating system software tape are summarized below.

<table>
<thead>
<tr>
<th>JDL</th>
<th>JDEs provided for:</th>
</tr>
</thead>
</table>
| **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: |
| 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 VM E/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 |
| **IBM RCA** | IBM OS and DOS standard-labeled tapes  
IBM ANSI-labeled and OS Writer tapes  
US70 (RCA)-labeled tapes |
| **ICL** | ICL 2900-labeled tapes |
| **NCR** | NCR-labeled tapes |
| **OLDUMP** | online dumps of host transmissions |
| **ONLINE** | online with/without banner or trailer pages |
| **POWERP** | IBM POWER VS, POWER 4.0  
POWER 4.1/4.2; IBM DOS GRASP (including Interpress tapes) |
| **RSX** | RSX-11 (FLX) ASCII tapes (including Interpress tapes) |
| **UNIVAC** | Univac SDF-formatted tapes |
| **XEROX** | ANSI-labeled tapes  
Unlabeled tapes  
Xerox ANSI-labeled tapes  
JDEs to print unknown tapes  
Printing tapes without attempting to deblock the tape properly |
After an LPS software system is created by the user, the source for these JDLs is resident in the JSL directory. Use the Editor PRINT parameter or the PDL compiler to obtain a printout of the JDL(s) desired.

Interpress data from magnetic tape

A limited set of Xerox Network Systems (XNS) transactions is supported for magnetic tape input to receive Interpress data from tape. Supported transactions are:

- Courier-encoded routing packets
- Courier-encoded procedure call message packets. (The procedure call supported is “Xerox NS version 3.0 Printing.Print.”)
- Bulk data packets. The bulk data is Interpress Commercial set 3.0 encoded data. Interpress Publication set is supported as well with the raster image processor (RIP) parameter. XNS protocols are described further in the LPS Operations Reference.
## Tape format

Figure D-2. **Tape format**

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spooler’s Preamble</td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>Sub-version</td>
</tr>
<tr>
<td>Connection control</td>
<td>Data stream type</td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Source socket number</td>
<td></td>
</tr>
<tr>
<td>Destination socket number</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Data bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible garbage byte</td>
<td></td>
</tr>
<tr>
<td>Potential padding</td>
<td></td>
</tr>
<tr>
<td>Spooler’s postamble</td>
<td></td>
</tr>
</tbody>
</table>

Xerox Simplified Transport Protocol logical block

[block diagram]
Interpress tape input must be formatted as follows:

1. Information is encapsulated on one tape in logical blocks which consist of a header (described in detail in the Xerox Simple Transport Protocol Standard) followed by data. A logical block is defined here as equivalent to one user record.

2. The header consists of:
   - 2 bytes Checksum (per the Internet Datagram Protocol)
   - 2 bytes Protocol version number (per the Sequenced Packet Protocol)
   - 1 byte Connection control (per the Sequenced Packet Protocol)
   - 1 byte Data stream type (per the Sequenced Packet Protocol)
   - 2 bytes Length (per the Internet Datagram Protocol)
   - 2 bytes Source socket number (per the Internet Datagram Protocol)
   - 2 bytes Destination socket number (per the Internet Datagram Protocol)
   - 4 bytes Reserve for future use

Points to note

- Version 0.1, (version 0 in the high-order byte; subversion 1, in the low order byte) is the only version currently supported.
- Currently, the end-of-message bit is defined for the connection control field. It is set in the header of any logical block containing the last byte of a data stream. Bit 3 (10 hexadecimal) is the assigned bit for the End-of-Message flag, as defined by the Sequenced Packet Protocol.
- The four data stream type values assigned are as follows:
  - 0 = Courier-encoded print message
  - 1 = Bulk data, such as Interpress
  - 2 = Courier-encoded routing data
  - 254 = End
- The data stream types and the connection control fields are planned to be compatible with the Sequenced Packet Protocol.

3. The logical block must be an integral number of 16-bit words, while the data contained in it is not so constrained, implying that there may be a garbage byte at the end of the logical block. This garbage byte is not included in the length field but is included in the checksum.

4. Instead of using addresses as defined in the Internet Datagram Protocol, routing information (destination and source) is represented by Courier-encoded clearinghouse three-part names, that is, Name:Domain:Organization. The Courier encoding of a clearinghouse name is a RECORD of three STRINGS. If the names are unknown/nonexistent, they are represented by strings of zero (0) length.
5. A file is defined as all the information between the start-of-file and end-of-file label groups. If the tape format supports it, a file may span volumes. A file fully contains one or more documents. (A document may not span file boundaries.) Information is stored on a file in the following order: 1 routing logical block 1 or more Courier-encoded print procedure message logical blocks 0 or more bulk logical blocks Optional repetitions of b- followed by c-type logical blocks An end logical block

6. The recipient may choose not to verify the checksum as the default. However, some means must be provided for enabling checksum verification if the sender specifically requests the service. The presence of valid checksums (as opposed to the null value, all bits, that is, FFFF hexadecimal) does not guarantee that checksums are verified by the recipient.

Usage requirements

To input Interpress data through magnetic tape:

1. The tape's format must comply with the Xerox Magnetic Tape Format Standard and the Xerox Simple Transport Protocol Standard.

2. Interpress must be specified from magnetic tape in the job START parameter. The START command has the following fields:

START [jde],[jdl],[S or M],[copies],[REPORTS:r1,r2,...],INTERPRESS

(Note that the keyword INTERPRESS must be included; fields in brackets are optional.)

3. Interpress must be specified in the job descriptor library (JDL) by using the print description language (PDL) parameter command:

VOLUME INTERPRESS=YES.

4. The parameter to enable error checker for Interpress tapes is provided, on a job basis, through the PDL parameter command:

VOLUME INTERPRESS=CHECKSUM (or NOCHECKSUM).

(Note that this software checking is in addition to any other error checking mechanism supplied by the tape driver's hardware.)

5. An appropriate JDL must be set up so that the data extraction processor can skip over the host-specific printer carriage control bytes and other control bytes normally associated with the line printer, record-orientated data. (Refer to the LPS Tape Formats Manual for details.)

Notes and restrictions

1. A logical block may not span a physical block, and a physical block may not contain more than one logical block. A logical block is defined as equivalent to one user data record. (Refer to “Xerox Simplified Transport Protocol logical block” example earlier in this appendix.)
2. The data portion of the logical block is restricted to a minimum length of 1024 bytes. The restrictions are placed to optimize performance. The restriction on the maximum logical block length is placed so the Interpress processing does not have a degrading impact on the available user space in input. It is best to create logical blocks of precisely 1040 bytes in length. This block length is the sum of 16 XNS control bytes and 1024 data bytes. When the length of the block is less than the minimum physical block length, padding bytes of unspecified value are appended to make up a physical block of the appropriate length. These padding bytes are discarded by the recipient of the system.

3. All routing packet information is ignored by the LPS.

4. The job is aborted when the checksum verification is enabled but fails.

5. The PDL parameter command, VOLUME PLABEL=YES or NO, is not supported for Interpress tapes. This parameter specifies if the tape labels are to be printed with the job.

6. All logical blocks are shown as they appear in the memory of a processor that has a 16-bit word; the first word of the logical block is in memory location \(m\), and the next word in the location \(m + 1\). The most significant byte of word \(m\) is placed on the tape first, followed by the least significant byte; the most significant byte of word \(m + 1\) is placed on the tape, followed by the least significant byte.

### Sample JSL

Figure D-3. **Sample JSL**

```
IPTAPE:SYS;
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* DEFINE SYSTEM COMMAND SET FOR IPTAPE.JDL */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
VOLUME CODE=NONE, HOST=XEROX, LABEL=NONE;
BLOCK LENGTH=528; /* 6 BYTE PREAMBLE */
RECORD LENGTH=512, STRUCTURE=VB, LTHFLD=1;
LINE PCC=(0,NOTRAN), PCCTYPE=NONE;
OUTPUT GRAPHICS=YES, OFFSET=FIRST;
ACCT USER=TRAY,
   DEPT='IPTPE';
VOLUME INTERPRESS=YES;
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* DEFINE JDE'S */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
DFLT:JOB;
1:   JOB;
   OUTPUT DUPLEX=YES;
END;
END;
```
Sample layout of data (Interpress)

The figure below represents the layout of data on tape for a simple print request of one copy of the accompanying Interpress master.

Figure D-4. **PKT HDR (Packet Header)**

The dashed delimiters in the diagram above represent the boundaries within the logical blocks. The solid delimiters represent any required physical block padding and the inter record gap.

The following tables explain the diagram further.
Table D-3. **PKT HDRa (Address logical block header)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFF</td>
<td>Checksum (Don't Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>1002</td>
<td>Connection control (last packet of this data stream type = 10 hexadecimal, in high byte); data stream type in (logical address = 2, in low byte)</td>
</tr>
<tr>
<td>0014</td>
<td>Length of packet + data, in bytes</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket # for Courier = 5)</td>
</tr>
<tr>
<td>0000</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>

Table D-4. **DATAa (Address logical block data, Courier encoded)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Source address (Courier-encoded string of zero length = no source address)</td>
</tr>
<tr>
<td>0000</td>
<td>Destination address (Courier-encoded string of zero length = no destination address)</td>
</tr>
</tbody>
</table>

Table D-5. **PKT HDRb (Service request logical block header)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFF</td>
<td>Checksum (Don't Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>1000</td>
<td>Connection control (last packet of this data stream type = 10 hexadecimal, in high byte); data stream type (Courier =0, in low byte)</td>
</tr>
<tr>
<td>0022</td>
<td>Length of packet + data, in bytes</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket # for Courier = 5)</td>
</tr>
<tr>
<td>0000</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>
Table D-6. **DATAb (Service request logical block data, Courier encoded)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Courier encoding of a Printing.Print, version 3 procedure call</td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>0004</td>
<td></td>
</tr>
<tr>
<td>0003</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
</tbody>
</table>

Table D-7. **PKT HDRc (Bulk data logical block header)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF</td>
<td>Checksum (Don't Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>0001</td>
<td>Connection control (not the last packet of this data stream type = zero, in the high byte); data stream type (bulk data = 1, in low byte)</td>
</tr>
<tr>
<td>0410</td>
<td>Length of packet + data, in bytes; first block in Interpress master</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket # for Courier = 5)</td>
</tr>
<tr>
<td>0000</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>0000</td>
<td></td>
</tr>
</tbody>
</table>

Table D-8. **DATAc (Bulk data logical block data)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>496E</td>
<td>Interpress/Xerox/3.0</td>
</tr>
<tr>
<td>7465</td>
<td></td>
</tr>
<tr>
<td>7270</td>
<td></td>
</tr>
<tr>
<td>7265</td>
<td></td>
</tr>
<tr>
<td>7373</td>
<td></td>
</tr>
<tr>
<td>2F58</td>
<td></td>
</tr>
<tr>
<td>6572</td>
<td></td>
</tr>
<tr>
<td>6F78</td>
<td></td>
</tr>
<tr>
<td>2F33</td>
<td></td>
</tr>
<tr>
<td>2E30</td>
<td></td>
</tr>
<tr>
<td>20...</td>
<td>... rest of data for first block of Interpress master</td>
</tr>
</tbody>
</table>
Table D-9. **PKT HDRc + 1 (Bulk data logical block header)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF</td>
<td>Checksum (Don’t Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>0001</td>
<td>Connection control (not the last packet of this data stream type = zero, in high byte); data stream type (bulk data = 1, in low byte)</td>
</tr>
<tr>
<td>0410</td>
<td>Length of packet + data, in bytes; next block of Interpress master</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket # for Courier = 5)</td>
</tr>
<tr>
<td>0000</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>

Table D-10. **DATAc + 1 (Bulk data logical block data)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>&gt;more of Interpress master</td>
</tr>
</tbody>
</table>

This logical block is followed by as many additional logical blocks as needed to contain the master. The last bulk data logical block has the following structure:

Table D-11. **PKT HDRc + n (Bulk data logical block header)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF</td>
<td>Checksum (Don’t Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>1001</td>
<td>Connection control (last packet of this data stream type = 10, in high byte); data stream type (bulk data = 1, in low byte)</td>
</tr>
<tr>
<td>nnnn</td>
<td>Length of packet + data, in bytes; last block of Interpress master, if length is odd, a garbage byte is appended to data</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket # for Courier = 5)</td>
</tr>
<tr>
<td>0000</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>
Table D-12. **Data + n (Bulk data in logical block data)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...end of Interpress master</td>
</tr>
</tbody>
</table>

Table D-13. **PKT HD Re (End of logical block header; there is no data associated with this logical block)**

<table>
<thead>
<tr>
<th>Hexadecimal value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF</td>
<td>Checksum (Don't Check = FFFF hexadecimal)</td>
</tr>
<tr>
<td>0001</td>
<td>Protocol version # = 0.1 (zero in high byte; 1 in low byte)</td>
</tr>
<tr>
<td>10FE</td>
<td>Connection control (last packet of this data stream type = 10, in high byte); data stream type (End = 254, in low byte)</td>
</tr>
<tr>
<td>0010</td>
<td>Length of packet + data, in bytes</td>
</tr>
<tr>
<td>0000</td>
<td>Source socket (zero = unknown)</td>
</tr>
<tr>
<td>0005</td>
<td>Destination socket (socket for # for Courier = 5)</td>
</tr>
<tr>
<td>0000 0000</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>
This appendix summarizes the basic Editor commands and steps for creating and modifying JSLs. For detailed information on using the Editor, refer to your Xerox 4635 LPS Operations Reference.

Note that you may enter the Editor commands by keying in only the first three letters, for example, CLEAR can be entered as CLE, and INSERT can be entered as INS.

Creating a new file on system disk

When you have prepared your JSL and are ready to enter it onto system disk, you must use the Editor. To use the Editor for coding your JSL, follow these steps:

1. Gain access to the file by entering the Editor. Enter:
   
   `LOG 3,password`

2. Begin the editing session by entering:
   
   `EDIT`

3. Clear the work file of all data, enter:
   
   `CLEAR`

4. Verify the new file name you have selected and protect existing files by entering:
   
   `GET file-id`

   The file-id consists of the file name and file type, represented as `file-name.file-type`, for example `XAMPLE.JSL`.

   A message appears on the system controller display if a file already exists with the file-id you have selected. To protect the original file, clear the work file, rename your file, and repeat the GET file-id command.

5. Number the lines
   
   • Automatically by entering:
   
   `INSERT n,y`

   `n` refers to the starting line, also called record number. `y` refers to increments at which the line numbers should appear. For example, the command `INSERT 10,10` means to insert lines starting with 10 and increase in increments of 10. The first record then would be 00010, with subsequent records numbered 00020, 00030, 00040, and so on.
To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

PCE

PCE represents PRINT, CLEAR, and END. This provides a print version of your source, clear the workfile, and end the editing session in one command.

SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

Obtaining files already stored on system disk

To transfer the source code to a work file, follow these steps:

1. Gain access to the file by entering:
   LOGON 3, password

2. Begin the editing session by entering:
   EDIT

3. Clear the work file of all data by entering:
   CLEAR

4. Retrieve the file you want by entering:
   GET file-name.file-type

5. SAVE file-id

Remember that file-id consists of file-name.file-type.

END

To end the editing session:

• To end the editing session without printing the work file, simply enter:
  END

• To end the editing session and print the work file, enter:
  PCE

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.

For example, if you wish to add line 00011 following 00010, you would simply enter PCE. The display would then provide the next record, 00020, for you to enter another line of text.
Displaying the text of a file

If you wish to see the lines of source code in the work file, you can use any of the following options:

- Display the entire work file by entering:
  ```
  DISPLAY
  ```

- Display a particular portion of the work file by entering:
  ```
  DISPLAY n-y
  ```
  
  n is the starting line number to be displayed.
  
  y is the ending line number to be displayed.

- Display a single line, or record, of the work file by entering:
  ```
  DISPLAY n
  ```
  
  n is the desired line.

- Display a particular word or phrase within a work file wherever it exists by entering:
  ```
  FIND/x/
  ```
  
  x is the word or phrase to be found and displayed.

- Display the starting and ending line numbers of the work file if you wish to know on which lines the work file is located by entering:
  ```
  KEYS
  ```
  
  The system provides these line numbers, as in this example:
  ```
  EDIT KEYS
  BEGINNING LINE NUMBER 00010
  ENDING LINE NUMBER 00190
  EDIT.
  ```

- Print a copy of the work file by entering:
  ```
  PCE
  ```
  
  This command is an abbreviated form of the PRINT, CLEAR, and END commands.

  Be sure to save the work file before using the PCE command by entering:
  ```
  SAVE file-id
  ```
  
  The file-id consists of the file name and file type, represented as file-name.file-type.

Modifying the text in a file

Use the following procedures to change lines of the source code in your program:

- Modify a portion of the work file line or
- Modify entire lines of the work file.
• Save your source code file.
• Terminate the editing session.

Modifying a portion of the work file line

To modify portions of a line, you must first enter:

**MODIFY n**

n is the number of the line to be modified, for example, **MODIFY 60**.

Then you can perform any of the following activities:

• Substituting one set of characters for another by entering:

  **S/x/y or C / x / y /**

  x is the text to be replaced.

  y is the text to be inserted.

  or

  **S [k]/x/y/ or C [k]/x/y/**

  k is the occurrence of string x in each line specified in the **MODIFY** command that is affected by the **S** or **C** command.

  k defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string x, in the specified lines are affected.

  Add a space between **S** or **C** and **k** in the command.

  • Overlay one set of characters over another by entering:

    **O/x/y/**

    x is the text to be replaced.

    y is the new text.

    or

    **O [k]/x/y/**

    k is the occurrence of string x in each line specified in the **MODIFY** command that is affected by the **OVERLAY** command. k defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string x, in the specified lines are affected.

    Add a space between **O** and **k** in the command.

  When **O** is used to replace characters that are not at the end of the line, be sure that the characters and spaces inserted equal those being replaced.

  • Insert a set of characters before other characters by entering:

    **P/x/y/**

    x is the word following the text to be added.

    y is the text to be added.

    or

    **P [k]/x/y/**
k is the occurrence of string x in each line specified in the MODIFY command that will be affected by the P command. k defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string x, in the specified lines are affected.

Add a space between P and k in the command.

• Insert a set of characters following other characters by entering:

F/x/y/
x is the word preceding the text to be added.
y is the text to be added.

or

F [k]/x/y/
k is the occurrence of string x in each line specified in the MODIFY command that is affected by the F command. k defaults to 1 when not specified, while 0 indicates that all occurrences of the first set of characters, or string x, in the specified lines are affected.

Add a space between F and k in the command.

Modifying entire lines

To change entire lines, or records, or your JSL coding, you may perform any of the following activities:

• Move lines in a file by entering:

MOVE n-m,p,i

n is the starting line number.
m is the ending line number.
p is the new starting line number.
i is the increment of line numbering you want.

For example, MOVE 80-11-,51,1 results in the following:

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>00080</td>
<td>00051</td>
</tr>
<tr>
<td>00090</td>
<td>00052</td>
</tr>
<tr>
<td>00100</td>
<td>00053</td>
</tr>
<tr>
<td>00110</td>
<td>00054</td>
</tr>
</tbody>
</table>

• Remove lines from a file by entering:

REMOVE n

or just enter the line number.
n is the line to be removed.

• Duplicate lines in a file by entering:

DUPLICATE n-m,p,i

n is the starting line number.
m is the ending line number.
p is the new starting line number.
i is the increment of line numbering you want.

- Replace lines with new lines by entering:
  \texttt{REPLACE n-m,i}
  \begin{itemize}
  \item n is the starting line number.
  \item m is the ending line number.
  \item i is the increment of line numbering you want.
  \end{itemize}
  This command deletes all lines from \texttt{n} to \texttt{m}. It initiates automatic line numbering at \texttt{n} for you to enter code and increments each line by your value for \texttt{i}.

- Renumber lines of text by entering:
  \texttt{RENUMBER n,i}
  \begin{itemize}
  \item n is the starting line number.
  \item i is the increment of line numbering you want.
  \end{itemize}
  After modifying a file, you may find significant gaps in the sequence of line numbers. The \texttt{RENUMBER} command allows you to renumber the lines by establishing your starting line number \texttt{n} and incrementing value \texttt{i}.

- Merge one file with another by entering:
  \texttt{MERGE file-id,n,i}
  \begin{itemize}
  \item file-id consists of filename.file-type and specifies the file to be merged into the one you are currently working on.
  \item n is the starting line where the merged file is to start.
  \item i is the line number increment you wish to use.
  \end{itemize}

\textbf{Saving your source code file}

After you have finished editing your file, save it by entering:

\texttt{SAVE file-id}

\texttt{file-id} consists of file-name.file-type.

\textbf{Terminating the editing session}

To end the editing session without printing the work file, enter:

\texttt{END}

To end the editing session and print out the work file, enter:

\texttt{PCE}

\texttt{PCE} represents the commands to PRINT the work file, CLEAR the work file, and END the editing session.
Glossary

A3  Paper size measuring 297 by 420 mm.

A4  Paper size measuring 210 by 297 mm.

Abort  To terminate the printing of a job or execution of a program before it completes.

Alignment  Positioning of an image on a page for printing.

Alphanumeric  Set of characters including the letters A through Z, numerals 0 through 9, and all printable special symbols.

AIS  Advanced Imaging Subsystem (AIS), made up of three PWBs, used for character dispatch and image generation functions.

ASCII  American Standard Code for Information Interchange. Standard 7-bit code that represents alphanumeric information. Each alphanumeric character and several nonprinting characters are assigned a binary number, covering 128 possible characters. It is used for information interchange among data processing systems, data communication systems, and associated equipment.

Applications Software  Host- or LPS-resident software that directs the computer to perform specific tasks or functions as opposed to the software used to operate the computer. Common business applications include payroll, accounting, and inventory.

Ascender  Portion of alphabetic character that rises above the body of the character (its x-height portion). See also descender; x height.

Audit Log  Captures the sheet delivery information for every page in an audited report, certain details about each sheet, the planned and actual report control totals, and waste management information.

B4  Paper size measuring 250 by 353 mm.

B5  Paper size measuring 178 by 254 mm.

Backup File  File copied to a storage medium for safekeeping in case the original is damaged or lost.
bar code character  A series of wide and narrow bars and spaces used as alphanumeric representations.

bit map  Visual representation of graphic images in which a bit defines a picture element (pixel); for example, if a bit is 1, the corresponding pixel is printed.

bit mapped  Display image generated bit by bit for each point or dot. A software-driven scanner is used to create characters or graphics.

blocking  Process of combining two or more records into a single block of data which can then be moved, operated upon, or stored, as a single unit by the computer.

block length  Number of characters or bytes contained in a block of data (the block is treated as a unit within the computer). Block length is usually invariable within a system and may be specified in units such as records, words, computer words, or characters.

bypass transport  Finisher for postprocessing activities such as booklet making, shrink wrapping, etc.

collate  To arrange or assemble into ordered sets.

comment  Explanations written with program instructions. They are ignored by the computer.

compiler  Software that translates instructions written in high-level language into machine language for execution by a system.

concatenate  To connect or link in a series, as when files are grouped together for faster processing. See also job concatenation mode.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>Functional unit containing devices used by an operator to communicate with an operating system. It may consist of a display, keyboard, and certain switches or other controls.</td>
</tr>
<tr>
<td>consumable supplies</td>
<td>Supplies such as paper and dry ink that are depleted (used up) during the course of normal printer operation.</td>
</tr>
<tr>
<td>continuous printing</td>
<td>Refers to Interpress job integrity under any of the following conditions: excessive graphics, forms, or font use problems.</td>
</tr>
<tr>
<td>coordinate</td>
<td>Point on the x and y axis that determines a grid position.</td>
</tr>
<tr>
<td>copy</td>
<td>To duplicate data in a new location or on an additional storage medium, for example, to copy files from disk to tape.</td>
</tr>
<tr>
<td>copy-sensitive</td>
<td>Term used to indicate jobs in which multiple copies of a report will contain different data, as with paychecks and banking statements.</td>
</tr>
<tr>
<td>cpi</td>
<td>characters per inch. Designates the number of characters per inch for a particular typeface. See also pitch.</td>
</tr>
<tr>
<td>debug</td>
<td>To detect and correct errors in a program.</td>
</tr>
<tr>
<td>default</td>
<td>Value assigned to a field by the system. Default fields may be used for such items as document formats, menu selections, input fields, font selection, and paper or image size. The default value of a field may be changed.</td>
</tr>
<tr>
<td>descender</td>
<td>Portion of alphabetic character that extends below the baseline. See also ascender, x height.</td>
</tr>
<tr>
<td>display</td>
<td>Viewing device (monitor) that visually communicates system warnings, status, and error messages and reflects operator interaction with the system on a display.</td>
</tr>
<tr>
<td>DJDE</td>
<td>Dynamic Job Descriptor Entry. Command within an input data stream used to modify the printing environment dynamically.</td>
</tr>
<tr>
<td>dot</td>
<td>Unit of measurement representing a fraction of an inch, for example, 300 dots per inch (dpi). It may also be referred to as a picture element (pixel) or spot.</td>
</tr>
<tr>
<td>download</td>
<td>To copy files using communication lines from the host onto LPS system disks.</td>
</tr>
<tr>
<td>duplex printing</td>
<td>Printing on both sides (front and back) of a page. See also simplex printing.</td>
</tr>
<tr>
<td>E13B</td>
<td>A special font with tightly controlled dimensions used in magnetic character recognition.</td>
</tr>
</tbody>
</table>
easy values  Font point sizes, orientations, and names explicitly defined in Font Interchange Standard (FIS).

EBCDIC  Extended Binary Coded Decimal Interchange Code. Coded character set consisting of 8-bit coded characters. It can accommodate 256 characters.

edgemarking  Use of graphic objects, usually lines or boxes, that bleed off the edge of the physical page. See also physical page.

electronic publishing  Integrated production of documents on demand, using digitally stored documents, computerized composition, and electronic printing systems.

elite  Smallest size standard typewriter type: 12 characters per horizontal inch.

embedded blanks  Blank spaces within a command line.

enhanced sheet tracking  A full accounting on a report-by-report basis of every sheet fed into the printer. The components that make up Enhanced Sheet Tracking include audit logging and reporting, feed and delivery data, and Waste Management.

Ethernet  Xerox local area network (LAN) that allows data to be transmitted by cable from one device to another, enabling it to share the network.

FCB  forms control buffer. Buffer for controlling the vertical format of printed output.

FDL  forms description language. LPS-resident source language used for designing electronic forms. See also FSL; form.

FIS  Font Interchange Standard. Standard that defines the digital representation of fonts and character metrics for the generation of an entire series of Interpress fonts.

fixed font  Font containing characters with fixed spacing. See also proportional font.

fixed pitch  Font set in which every character cell has the same width. In reference to character sets, this term describes typefaces in which all character cells are of equal width. Monospaced as opposed to proportional spaced.

fixed spacing  Arrangement of characters on a line so that all characters occupy the same amount of horizontal space.

font  Set of images, usually characters and symbols, having common characteristics such as style, width, height, and weight.
form  1. Compiled forms source library (FSL) file.  2. Printed or typed document with blank spaces for inserting information. Specific arrangement of lines, text, and graphics stored in a computer under an identifying name. Page of data that, when preceded by proper commands, is stored on the system disk as a permanent file. It may be merged with variable data by a form start command. See also FDL; FSL.

format  1. Layout of a document, including margins, page length, line spacing, typeface, and so on.  2. In data storage, the way the surface of a disk is organized to store data.  3. To prepare the surface of a disk for acceptance of data.

FSL  Forms source library. Uncompiled collection of user-created files containing FDL commands. Refer to FDL; form.

graphics  Use of lines and figures to display data, as opposed to using text.

grid  Imaginary pattern of evenly spaced horizontal and vertical lines on a page.

grid unit  Smallest rectangle enclosed by horizontal and vertical lines on a grid. The size of a grid unit is expressed as the length of one side of a rectangle.

hardcopy  Machine output in permanent form, such as printed reports, listings, and so on. Output in a permanent form (usually on paper or paper tape) rather than in temporary form, as on a display. Contains readable printed copy of machine (for example, computer) output.

hard values  Nonoptimal adjustment of particular FIS fonts in terms of point size and orientation.

hardware  Physical components, such as mechanical, magnetic, and electronic elements of a system, as opposed to programs, procedures, rules, and associated documentation. Hardware is operated by software and firmware.

HCF  High-Capacity Feeder. An LPS option that increases the number of input paper trays.

HCS  High-Capacity Stacker. An LPS option that increases the number of feeder stackers.

hexadecimal  Numbering system with a base of 16. In this system, 10 through 15 are represented by A through F, respectively.

hierarchy  Relative priority assigned to arithmetic or logical operations that must be performed.
host  Computer accessed by users which serves as a source of high-speed data processing for workstations with less computer power. See also mainframe.

host interface  Connection between network and host computer.

id  Identifier. Character used to identify or name data and possibly to indicate certain properties of that data.

image area  Area on a physical page that may contain text or graphics.

input  Data or text introduced into a computer-based system.

input/output  General term encompassing the flow of data into and out of a system.

interpolation  Series of logical rules implemented in the printer to convert a 300 spi input video stream to a 600 spi output video stream. Interpolation is functionally analogous to bit doubling (2x scaling), except the logical rules result in superior output.

Interpress  Xerox standard that defines digital representation of lines for printing. Interpress documents can be printed on any sufficiently powerful printer equipped with Interpress print software.

Interpress font utility (IFU) program  Program used to convert FIS fonts to LPS fonts.

Interpress master  File written according to the Interpress standard.

JDE  Job descriptor entry. Collection of job descriptions. See also job; JSL.

JDL  Job descriptor library. Collection of compiled job descriptions. See also JSL.

job  1. Set of instructions (JDEs) defining a unit of work for the system. 2. In setting a separation boundary through the Bin Full Criteria task, job refers to everything printed as the result of a single start command. See also JDE.

job concatenation mode  In HIP, a mode in which multiple print jobs are processed as reports in one print job. See also concatenate.

job control  Program called into storage to prepare each job or job step to be run.

JSL  Job source library. Collection of uncompiled job descriptions. See also job; JDE; and JDL.

keyword  Required part of a command. See also operator command.
label  1. In data storage, a reference to a file saved on tape or disk, a
record indicating the file name or date created, or other control
information.  2. In programming, a name assigned to a particular
instruction or portion of a program as a locational reference (the
computer translates the label into an address).

landscape page orientation  Orientation of print lines or top of an illustration parallel to the
long edge of the paper if the sheet is within the standard size
range. (Sheets larger than standard have the reverse print
orientation.)

language  Defined set of characters and symbols combined together by
specific rules. See also high-level language; low-level language.

laser printing  Technology that uses a laser to transfer character forms to a page
by direct or indirect means.

leading  1. Vertical distance between lines (also called line space),
measured from a baseline of one line to the baseline of the next.
2. Extra spacing between lines of type.  3. In typography,
spacing between lines and paragraphs.

LED  Light emitting diode. Solid substance that glows when a current
is passed through it. Often used for indicator lights on disk
drives or modems, as well as for displays on other electronic
equipment.

LEF  long-edge feed. The movement of paper through the printer in
the direction of the paper length (the longer side of a sheet of
paper).

legal size  Sheet the standard size of legal briefs, 8.5 by 14 inches.

letter size  Paper sized 8.5 by 11 inches/216 by 279 mm.

library  In data storage, a collection of related files or programs.

line  One horizontal flow of characters.

line feed  Control character that, unless set to be interpreted as a line end,
causes the printing system to begin printing in the current
character position of the next line.

line printer  High-speed printer that prints an entire line of characters at one
time.

line tables  Internal data structures providing a record in memory of lines to
be drawn on a page.

location  Place in which data can be stored.
**logical page**  In the Xerox printing systems environment, a formatted page that is smaller than the physical page. A logical page is defined by an origin, thus allowing more than one logical page to be placed on a physical page.

**logo**  Small illustration or design, usually simple, typically used to identify a company.

**log off**  Procedure by which a user ends a session.

**log on**  Procedure by which a user begins a session between an application program and a logical unit.

**magnetic media**  Term for all storage devices, such as disks, diskettes, and tape, on which data is stored in the form of magnetized spots on surface of the media.

**magnetic tape**  Flexible plastic tape, with one side offering a magnetic surface suitable for storing computer data in the form of magnetized spots. Magnetic tape is often used for long-term storage since it can accommodate large volumes of information.

**mainframe**  Central processing unit (CPU) and memory of a large computer. More often used to denote any large computer of the type that might be used to control a group of smaller computers, terminals, or other devices. See also host.

**margins**  White space on each side of printed text.

**mask**  1. Selection of bits from a storage unit by using an instruction that eliminates the other bits in the unit. 2. In accessing files, a file name mask is used to reference one or more files with similar file-id (identifier) syntax. 3. In Interpress, a mask serves as a template, indicating the shape and position of an object on a page.

**menu**  List of available functions, commands, and options.

**message**  Unit of information transmitted by one facility to another in a form that the receiving facility can understand and act upon. The standard message format consists of a header (containing identifying and control information), followed by the actual message content, followed by a trailer (indicating that the message is completed).

**MICR**  Magnetic Ink Character Recognition. Allows a computer to read characters printed in special metallic ink by scanning the shapes of their magnetic fields. MICR fields are often imprinted on bank checks before processing.

**nesting**  Subroutine or set of data, such as a comment, contained sequentially within another.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>object file</td>
<td>Source file converted into machine language (binary code).</td>
</tr>
<tr>
<td>octal</td>
<td>Numbering system which uses base 8.</td>
</tr>
<tr>
<td>offline</td>
<td>Devices not under the active control of a central processing unit. For example, a computer makes output to a magnetic tape. The tape is then used by an offline printing system to produce printed data. Offline operations are much slower than online operations. See also online.</td>
</tr>
<tr>
<td>offset</td>
<td>To place pages currently being printed in slightly different positions from previous pages.</td>
</tr>
<tr>
<td>online</td>
<td>Devices under the direct control of a central processing unit, for example a printing system in interactive communication with a mainframe. See also offline.</td>
</tr>
<tr>
<td>operating system</td>
<td>Basic host- or LPS-resident controlling program that governs the operations of a computer, such as job entry, input/output, and data management. The operating system is always running when the computer is active. Unlike other types of programs, it does not run to an end point and stop. The operating system of a Xerox LPS is referred to as the operating system software (OSS).</td>
</tr>
<tr>
<td>operator area</td>
<td>The 24-inch exclusive clearance that must be available directly in front of each component of an LPS for operator activities.</td>
</tr>
<tr>
<td>operator command</td>
<td>Statement to control a program, issued via a console device or terminal, causing a control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.</td>
</tr>
<tr>
<td>orientation</td>
<td>1. In reference to image area, orientation describes whether the printed lines are parallel to the long edge of the paper or the short edge of the paper. 2. Choice of printing portrait (vertically) or landscape (horizontally).</td>
</tr>
<tr>
<td>origin</td>
<td>In reference to image area, this is the upper left corner of a sheet.</td>
</tr>
<tr>
<td>overprinting</td>
<td>Printing more than one character at the same position.</td>
</tr>
<tr>
<td>overprint lines</td>
<td>Print lines whose carriage control specifies printing with no line spacing after the last printed line.</td>
</tr>
<tr>
<td>page detail</td>
<td>Non-printable customer encoded data used to describe each audited page of a report. Page details are used to create 4635MX audit files. Examples of page detail include check amount, payee name, account number, and check number.</td>
</tr>
<tr>
<td>page orientation</td>
<td>Direction in which data is printed on a report. See also landscape page orientation; portrait page orientation.</td>
</tr>
</tbody>
</table>
**parameter**  Part of a command, other than the keyword. See also keyword; operator command.

**pass through job**  On systems with XPAF, a job that is sent directly from a host to a Xerox printer via XPAF, without undergoing XPAF processing.

**password**  Unique word or set of characters that an operator or user must supply to log on to a system.

**patch**  In programming, to modify a portion of the program at the machine language level (as opposed to modifying at the source program level).

**PDL**  Print description language. Language used to describe printing jobs to an LPS. PDL describes the input (type, format, characteristics), performs the processing functions (logical processing), and describes the output (type, format, font selection, accounting options).

**physical page**  Sheet of paper on which printing is done. See also edgemarking.

**point**  Unit of measurement equal to 0.0139 inch. Points are always used to express type size and leading. There are 12 points to a pica and about 72 points to every inch. See also pica.

**point size**  Height of character set from the top of its ascenders to the bottom of its descenders in units (points). Point size does not always include leading.

**portrait page orientation**  Orientation of print lines or the top of an illustration parallel to the short edge of the paper if the sheet is within the standard size range. Sheets larger than standard have the reverse print orientation.

**print assurance**  The complete offering of the Assurance feature including the validation of the print job by the bar code reader. Print Assurance includes Audit Logging and Reporting, Bar Code option, Page Detail and Enhanced Sheet Tracking.

**printer**  Output device that produces hardcopy printouts. Also referred to as the IOT.

**prompt**  Message or symbol displayed on a system console requiring the operator to take action.

**proportional font**  Font containing characters that vary in width. See also fixed font.

**proportional spacing**  Text in which each alphanumeric character is given a weighted amount of space. Such output has print-like appearance. Proportional spacing allows more space for wide characters and less space for narrow characters.
proportional type: Characters that vary in width.

purge: To delete data from a system.

queue: List of documents waiting to be processed.

raster data: Binary data, usually consisting of dots arranged in scan lines, according to the print order.

rasterization: Creation of a page's bit map image for printing.

record: Collection of data or words treated as a unit.

report: In setting a separation boundary through the Bin Full Criteria task, report refers to a subset of a job (a job may consist of one or more reports).

resolution: Number of dots per unit. The LPS imaging system converts a character from digitized data into a printed image composed of these tiny dots. The greater the number of dots per inch, that is, the resolution, the clearer the image that is produced.

response time: Time elapsed between an inquiry transmitted from a terminal to a central processor and the receipt of a response at the same terminal.

RIP: raster image processor. LPS option that supports the Interpress V3.0 Publications Set.

ROM: Read-only memory. Solid-state memory for programs. It cannot be rewritten.

save: To store programs, data, or text for retrieval at a later time.

scale: To adjust font or image size according to given proportions.

scroll: Manipulation of a display to bring upper or lower portions of a document page into view when no space is available for the entire document at once.

SEF: short-edge feed. The movement of paper through the printer in the direction of the paper width (the shorter side of a sheet of paper). For the 4135 LPS, short-edge feed allows larger sizes of paper (up to 11 by 17 inches/279 by 432 mm) to be printed.

sequence control data: Customer encoded data used to provide the IOT with control information for the Bar Code reader.

set: In setting a separation boundary through the Bin Full Criteria task, set refers to multiple copies of the same report.
simplex printing  Printing on one side of the page. See also duplex printing.

source file  File containing source language statements or commands.

special processing  Commands allowing the user to process special reports, such as printing certain records, or printing on special paper.

spot  Unit of measurement representing a fraction of an inch, for example, 300 spots per inch (spi). May also be referred to as a picture element (pixel) or dot.

statement  Detailed written instructions in a program step. A statement is written according to specific rules called syntax.

static data  Information usually found on preprinted forms or overlays.

stock  User-defined name in the JSL that specifies a certain type of paper for printing a job. See also cluster.

stockset  Collection of stocks to be used on a print job. See also stock; cluster.

storage  Retention of information. Primary storage refers to internal storage where data and program instructions pertinent to current operations/jobs are held. Auxiliary storage refers to external media, such as disks or tapes, for use at a later time.

string  Connected sequence of alphanumeric characters treated as one unit of data by a program.

symbol  Character used in a computer language to specify a particular function.

synchronous  Efficient encoding of data suitable for high-speed, block-oriented data transmission by using equal time elements.

syntax  Rules governing the structure of expressions in a programming language.

syntax error  System response to a mistake in a command entry.

system  1. In data processing, a collection of parts and procedures organized to accomplish a set of specific functions. 2. Assembly of components united by some form of regulated interaction to form an organized whole. 3. Operations or procedures through which a business activity is accomplished.

system controller  Part of the LPS that provides interfacing capability, data handling, formatting, buffering, and operator control for the system. Also referred to as the “ESS.”
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>system disk</td>
<td>Magnetic disk storage medium, usually of large capacity, that is not removable as opposed to floppy disk or disk packs.</td>
</tr>
<tr>
<td>system file</td>
<td>Master software program that keeps all components working together.</td>
</tr>
<tr>
<td>system generation</td>
<td>Process whereby the system is made ready to operate. Typically involves selecting the operative parameters and activating the relevant software.</td>
</tr>
<tr>
<td>system page</td>
<td>Maximum area in which text and graphics can be imaged on a printing system.</td>
</tr>
<tr>
<td>system software</td>
<td>Software programs that support and/or control system functions by governing hardware operation and input/output processes, interpreting source programs and breaking them down into machine language, distributing tasks among various processors, and so on.</td>
</tr>
<tr>
<td>systems specialist</td>
<td>The person(s) at a customer site responsible for software, applications, and programming issues.</td>
</tr>
<tr>
<td>tab</td>
<td>To move the cursor on a display or printer to a prespecified column on the display or paper.</td>
</tr>
<tr>
<td>tape</td>
<td>Recording media for data or computer programs. Tape can be in permanent form, such as perforated paper tape. Generally, tape is used as a mass storage medium in magnetic form and has a far higher storage capacity than disk storage, but it takes longer to write or recover data from tape than from disk.</td>
</tr>
<tr>
<td>tape density</td>
<td>The number of characters that can be stored on magnetic media, such as how close together data can be recorded. The Xerox LPS may use either 1600 bpi or 6250 bpi density magnetic media.</td>
</tr>
<tr>
<td>tape drive</td>
<td>Input/output device that controls the movement of magnetic storage tape past the read/write head while data is accessed or stored.</td>
</tr>
<tr>
<td>task</td>
<td>1. Any major job performed by a computer. 2. One of several programs being executed by a system.</td>
</tr>
<tr>
<td>templates</td>
<td>1. Preset document formats, usually furnished along with application software, such as electronic spreadsheets or database programs. 2. Also applies to keyboard overlays showing function keys for particular software packages.</td>
</tr>
<tr>
<td>temporary storage</td>
<td>Main memory locations reserved for intermediate results of processing, control values, or other information which need to be kept on hand as a program proceeds.</td>
</tr>
</tbody>
</table>
**terminal**  Device equipped with a keyboard and connected to a computer or a network.

**testing**  1. Process of running a program for the express purpose of discovering any errors it may contain.  2. For computer-oriented systems, the process of verifying a system’s ability to meet performance objectives in a simulated environment or validating its actual performance in a live environment.

**text**  In communications, the content portion of a transmitted message.

**text string**  Consecutive series of characters to be printed exactly as specified in a command.

**throughput**  In data processing systems, the amount of data that can be processed, transmitted, printed, and so on, per a specified unit of time.

**toggle**  To switch (alternate) from one tray to another. The system will switch from an active feeder or stacker tray to an inactive one to allow continuous printing when the proper commands are invoked.

**token**  Coded character representing a word used in programs, for example, STOP is a word and the one-byte token for it in a BASIC dialect is 250.

**track**  Unit of storage space on a data storage medium, such as a disk. All information stored in a track can be retrieved or recorded in one positioning of a read/write head.

**trailer**  In data communications, the last portion of a message that signals the end.

**transaction processing**  Method of data processing in which files are updated and results are generated immediately after data entry.

**translation**  1. In data communications, the conversion of one code to another on a character-by-character basis.  2. In programming, the function of a language processor which converts a source program from one programming language to another.

**transmission speed**  In data communications, the rate at which data is passed through communication lines, usually measured in bits per inch (bpi).

**transmit**  To send data from one place to another.

**truncated**  Cut off before completion, as when data transfer from a host to a printer is cut off before all data has been transmitted.
**turnaround time**
1. In data processing, the time elapsed between submission of a job for processing and the turnaround for results.
2. In communications, the time required to convert a device from receiving mode to its sending mode or vice versa.

**two-up**
Application that prints two logical pages on one side of a physical page.

**typeface**
1. All type of a single design.
2. Set of characters with design features that make them similar to one another.

**type size**
Height of a typeface, measured from the bottom of its descenders to the top of its ascenders, expressed in points.

**type style**
Italic, condensed, bold, and other variations of typeface that form a type family.

**UCS**
Universal Character Set. Printer feature that permits the use of a variety of character arrays.

**upload**
To copy files from a remote peripheral device to a host. LPS files are not copied to the host because one of the LPS functions is to store files for the host.

**utility program**
General-purpose program that performs activities, such as initializing a disk or sorting, which are not specific to any application.

**validation**
Process of testing a system’s ability to meet performance objectives by measuring and monitoring its performance in a live environment.

**variable**
Information of a changeable nature which is merged with a standard or a repetitive document to create specialized or personalized versions of that document.

**variable data**
Variable data is not a part of a form design. It varies from page to page.

**variable text**
Text of changing nature, such as various names and addresses combined with a form letter to make a complete document.

**virtual page**
Page area selected by a forms designer for printing.

**warning message**
System-issued message indicating that an error has been encountered even though the program has not failed.

**waste management**
Provides more control of sensitive paper stocks by increasing the amount of information available with regard to how many sheets were fed from which tray, and the final destination of these sheets.
**weight**  Characteristic of type determined by how light or dark it appears.

**wildcard**  Part of a command (* symbol, / symbol, ? symbol) that represents a category for which the possible options are requested.

**wildcarding**  In a command, calling out a general category rather than a specific item within that category. The purpose of wildcarding is to generate the options within the given category.

**write**  To record data in memory or an external storage medium.

**write protection**  Data protection feature implemented on magnetic media (for example floppy disk, 9-track tape) to prevent stored data from being modified, written over, or erased.

**x axis**  Horizontal axis on a forms grid.

**xdot**  Unit of measurement representing a fraction of an inch. It may also be referred to as a picture element (pixel) or spot, for example, 1/600 spots per inch (spi).

**xerographic engine**  Component of a printer that develops an image, transfers it to paper, and fuses it for output as hardcopy.

**x height**  Height of lowercase letters without their ascenders or descenders (height of letter “x”). See also ascender; descender.

**x shift**  XSHIFT feature allows short-edge binding to shift in the x-direction to support the INVERT option.

**y axis**  Vertical axis on a forms grid.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS</td>
<td>Advanced Imaging Subsystem</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>BCD</td>
<td>binary coded decimal</td>
</tr>
<tr>
<td>BOF</td>
<td>bottom of form</td>
</tr>
<tr>
<td>BOT</td>
<td>beginning of tape</td>
</tr>
<tr>
<td>bpi</td>
<td>bits per inch</td>
</tr>
<tr>
<td>bps</td>
<td>bits per second</td>
</tr>
<tr>
<td>BSC</td>
<td>binary synchronous communications</td>
</tr>
<tr>
<td>CCID</td>
<td>character code identifier</td>
</tr>
<tr>
<td>CD</td>
<td>character dispatcher</td>
</tr>
<tr>
<td>CDC</td>
<td>control data communications</td>
</tr>
<tr>
<td>CD/IG</td>
<td>character dispatcher/image generator</td>
</tr>
<tr>
<td>CM</td>
<td>control module</td>
</tr>
<tr>
<td>CME</td>
<td>copy modification entry</td>
</tr>
<tr>
<td>CMT</td>
<td>character mapping table</td>
</tr>
<tr>
<td>cpi</td>
<td>characters per inch</td>
</tr>
<tr>
<td>CPU</td>
<td>central processing unit</td>
</tr>
<tr>
<td>CR</td>
<td>carriage return</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>CSI</td>
<td>command status interface</td>
</tr>
<tr>
<td>CSR</td>
<td>customer service representative</td>
</tr>
<tr>
<td>DC</td>
<td>data center</td>
</tr>
<tr>
<td>DDCMP</td>
<td>Digital Data Communication Message Protocol</td>
</tr>
<tr>
<td>DEC</td>
<td>Digital Equipment Corporation</td>
</tr>
<tr>
<td>DJDE</td>
<td>dynamic job descriptor entry</td>
</tr>
<tr>
<td>DMA</td>
<td>direct memory access</td>
</tr>
<tr>
<td>DOS</td>
<td>disk operating system</td>
</tr>
<tr>
<td>dpi</td>
<td>dots per inch</td>
</tr>
<tr>
<td>DSDD</td>
<td>double sided double density</td>
</tr>
<tr>
<td>DSR</td>
<td>disk save and restore</td>
</tr>
<tr>
<td>DSSD</td>
<td>double sided single density</td>
</tr>
<tr>
<td>EAIPX</td>
<td>enhanced advanced image processor</td>
</tr>
<tr>
<td>EBCDIC</td>
<td>Extended Binary Coded Decimal Interchange Code</td>
</tr>
<tr>
<td>ECMA</td>
<td>European Computer Manufacturers Association</td>
</tr>
<tr>
<td>EGIFM</td>
<td>enhanced graphics ink and font memory</td>
</tr>
<tr>
<td>EMT</td>
<td>emulator trap</td>
</tr>
<tr>
<td>ENET</td>
<td>ethernet network</td>
</tr>
<tr>
<td>EOT</td>
<td>end of tape</td>
</tr>
<tr>
<td>Term</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>EPBIX</td>
<td>enhanced page buffer interface</td>
</tr>
<tr>
<td>ESS</td>
<td>electronic subsystem, also referred to as the system controller</td>
</tr>
<tr>
<td>FCB</td>
<td>forms control buffer</td>
</tr>
<tr>
<td>FCP</td>
<td>file control parameter</td>
</tr>
<tr>
<td>FDL</td>
<td>forms description language</td>
</tr>
<tr>
<td>FDR</td>
<td>file directory</td>
</tr>
<tr>
<td>FFM</td>
<td>font file management</td>
</tr>
<tr>
<td>FIS</td>
<td>Font Interchange Standard</td>
</tr>
<tr>
<td>FMS</td>
<td>file management subsystem</td>
</tr>
<tr>
<td>FPS</td>
<td>formatting print service</td>
</tr>
<tr>
<td>FSL</td>
<td>forms source library</td>
</tr>
<tr>
<td>FST</td>
<td>font specification table</td>
</tr>
<tr>
<td>GCR</td>
<td>group code recording</td>
</tr>
<tr>
<td>gsm</td>
<td>grams per square meter</td>
</tr>
<tr>
<td>HCF</td>
<td>high capacity feeder</td>
</tr>
<tr>
<td>HCS</td>
<td>high capacity stacker</td>
</tr>
<tr>
<td>HFDL</td>
<td>host forms description language</td>
</tr>
<tr>
<td>HIP</td>
<td>Host Interface Processor</td>
</tr>
<tr>
<td>hpos</td>
<td>horizontal positioning</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines Corporation</td>
</tr>
<tr>
<td>IFU</td>
<td>Interpress font utility</td>
</tr>
<tr>
<td>IGM</td>
<td>image generator module</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IOM</td>
<td>image output module</td>
</tr>
<tr>
<td>IOT</td>
<td>input output terminal, also referred to as “printer”</td>
</tr>
<tr>
<td>IPD</td>
<td>Interpress decomposer</td>
</tr>
<tr>
<td>IPFONTS</td>
<td>Interpress fonts</td>
</tr>
<tr>
<td>IPL</td>
<td>initial program load</td>
</tr>
<tr>
<td>IPM</td>
<td>Interpress mapping</td>
</tr>
<tr>
<td>ips</td>
<td>inches per second</td>
</tr>
<tr>
<td>JCB</td>
<td>job control block</td>
</tr>
<tr>
<td>JCL</td>
<td>job control language</td>
</tr>
<tr>
<td>JDE</td>
<td>job descriptor entry</td>
</tr>
<tr>
<td>JDL</td>
<td>job descriptor library</td>
</tr>
<tr>
<td>JID</td>
<td>job identifier</td>
</tr>
<tr>
<td>JSL</td>
<td>job source library</td>
</tr>
<tr>
<td>LAN</td>
<td>local area network</td>
</tr>
<tr>
<td>laser</td>
<td>light amplification by stimulated emission of radiation</td>
</tr>
<tr>
<td>LED</td>
<td>light-emitting diode</td>
</tr>
<tr>
<td>LEF</td>
<td>long-edge feed</td>
</tr>
<tr>
<td>LF</td>
<td>line feed character</td>
</tr>
<tr>
<td>lpi</td>
<td>lines per inch</td>
</tr>
<tr>
<td>LPS</td>
<td>Laser Printing System</td>
</tr>
<tr>
<td>LUN</td>
<td>logical unit number</td>
</tr>
<tr>
<td>MICR</td>
<td>magnetic ink character recognition</td>
</tr>
<tr>
<td>MIS</td>
<td>management information systems</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MPU</td>
<td>microprocessor unit</td>
</tr>
<tr>
<td>MTU</td>
<td>magnetic tape unit (refers to the 9-track magnetic tape drive; also referred to as “magnetic tape drive”)</td>
</tr>
<tr>
<td>OCR</td>
<td>optical character recognition</td>
</tr>
<tr>
<td>OCS</td>
<td>operator communications subsystem</td>
</tr>
<tr>
<td>OLI</td>
<td>online interface</td>
</tr>
<tr>
<td>OS</td>
<td>operating system</td>
</tr>
<tr>
<td>OSDS</td>
<td>operating system diagnostic software</td>
</tr>
<tr>
<td>OSEEXEC</td>
<td>operating system executive</td>
</tr>
<tr>
<td>OSS</td>
<td>operating system software</td>
</tr>
<tr>
<td>PC</td>
<td>personal computer</td>
</tr>
<tr>
<td>PCC</td>
<td>printer carriage control</td>
</tr>
<tr>
<td>PC UI</td>
<td>personal computer user interface</td>
</tr>
<tr>
<td>PDL</td>
<td>print description language</td>
</tr>
<tr>
<td>PE</td>
<td>phase encoded</td>
</tr>
<tr>
<td>ppm</td>
<td>pages per minute</td>
</tr>
<tr>
<td>PQA</td>
<td>print quality adjustment</td>
</tr>
<tr>
<td>prom</td>
<td>programmable read-only memory</td>
</tr>
<tr>
<td>PSC</td>
<td>printer subsystem controller</td>
</tr>
<tr>
<td>pt</td>
<td>point</td>
</tr>
<tr>
<td>PWBA</td>
<td>printed wiring board assembly</td>
</tr>
<tr>
<td>QIC</td>
<td>quarter-inch cartridge</td>
</tr>
<tr>
<td>RAM</td>
<td>random access memory</td>
</tr>
<tr>
<td>RIP</td>
<td>raster image processor</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>ROS</td>
<td>raster output scanner</td>
</tr>
<tr>
<td>RTC</td>
<td>real-time clock</td>
</tr>
<tr>
<td>SAFES</td>
<td>stand-alone field engineering software</td>
</tr>
<tr>
<td>SAN</td>
<td>software analysis number</td>
</tr>
<tr>
<td>sci</td>
<td>START command index</td>
</tr>
<tr>
<td>SCSI</td>
<td>small computer system interface</td>
</tr>
<tr>
<td>SEF</td>
<td>short-edge feed</td>
</tr>
<tr>
<td>SFS</td>
<td>status file services</td>
</tr>
<tr>
<td>SIF</td>
<td>sequence insert file</td>
</tr>
<tr>
<td>SNA</td>
<td>system network architecture</td>
</tr>
<tr>
<td>spi</td>
<td>spots per inch</td>
</tr>
<tr>
<td>SST</td>
<td>system software tape</td>
</tr>
<tr>
<td>sysgen</td>
<td>system generation</td>
</tr>
<tr>
<td>TL/DL</td>
<td>text line/display list</td>
</tr>
<tr>
<td>TOF</td>
<td>top of form</td>
</tr>
<tr>
<td>tpi</td>
<td>tracks per inch</td>
</tr>
<tr>
<td>UCS</td>
<td>Universal Character Set</td>
</tr>
<tr>
<td>UCSB</td>
<td>Universal Character Set Buffer</td>
</tr>
<tr>
<td>vpos</td>
<td>vertical positioning</td>
</tr>
<tr>
<td>WAN</td>
<td>wide area network</td>
</tr>
<tr>
<td>XDGI</td>
<td>Xerox DCF and GDDM Interface</td>
</tr>
<tr>
<td>XDSS</td>
<td>Xerox Documentation and Software Services</td>
</tr>
<tr>
<td>XICS</td>
<td>Xerox Integrated Composition System</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>XJCF</td>
<td>Xerox Job Control Facility</td>
</tr>
<tr>
<td>XMP</td>
<td>xerographic mode persistence</td>
</tr>
<tr>
<td>XMS</td>
<td>xerographic mode switching</td>
</tr>
<tr>
<td>XNS</td>
<td>Xerox Network Systems</td>
</tr>
<tr>
<td>XPAF, XP</td>
<td>Xerox Printer Access Facility</td>
</tr>
<tr>
<td>XPMF-VMS</td>
<td>Xerox Print Management Facility - VMS Version</td>
</tr>
<tr>
<td>XPPI</td>
<td>Xerox Pen Plotter Interface</td>
</tr>
<tr>
<td>XPS</td>
<td>Xerox Publishing System</td>
</tr>
</tbody>
</table>
Numerals
3211/4245 mode, online, 4-39
4635 compatibility, 3-22 to 3-26

A
ABNORMAL command
description, 6-2
ERROR parameter, 6-2 to 6-3
IMISMATCH parameter, 6-3 to 6-4
ISUBSTITUTE parameter, 6-5
marker page, 6-7 to 6-8
OTEXT parameter, 6-5
PAGES parameter, 6-5 to 6-6
parameters, summary, 6-2
recovery, 6-6 to 6-7
REP parameter, 6-6
SECURITY parameter, 6-6
accounting
page, delimiter, 7-27
reports, see ACCT command
ACCT command
accounting data, 6-9, 6-12
dept parameter, 6-14
description, 6-9 to 6-10
example, 6-15
installation accounting, 6-10 to 6-11
overall usage, 6-9, 6-13
parameters, summary, 6-9
points to note, 6-15
USER parameter, 6-14
ACCTINFO parameter, RSTACK command, 7-28
ADJUST parameter
BLOCK command, 4-2
RECORD command, 4-14
Advanced Image Subsystem, see AIS
ADVTAPE parameter, PCC command, 4-9
AIS, description, 1-2, 1-6 to 1-7, 8-51
ALTER parameter, DJDE, 8-15
ANSI carriage control commands, 4-40
ASCII
job source language, see JSL
string constants, PDL, 2-7
ASSIGN parameter
CODE command, 4-7 to 4-8
DJDE, 8-15 to 8-16
PCC command, 4-10 TO 4-11
STOCKSET command, 6-86
audit log, 6-36, 6-42 to 6-45

B
BANNER command
description, 7-10
examples, 7-13 to 7-14

H
HCOUNT parameter, 7-11
HJOBNO parameter, 7-11
HRPTNA parameter, 7-12
parameters, summary, 7-11
points to note, 7-13
TCOUNT parameter, 7-12
TEST parameter, 7-12
TYPE parameter, 7-13
banner page detection, 7-10, 7-26
BARCODE command
description, 6-15
points to note, 6-16 to 6-17
BATCH
mode, graphics, 9-3 to 9-7
parameter, DJDE, 8-18
BDELETE command
description, 7-14
examples, 7-15 to 7-16
points to note, 7-15
TEST parameter, 7-14
BEGIN parameter
DJDE, 8-18 to 8-19
PDE command, 6-80
RRESUME command, 7-32
RSUSPEND command, 7-32
BFORM parameter
DJDE, 8-19
OUTPUT command, 6-47
block
delimiter constant, 4-6
length, 4-6
mode, 9-2
BLOAD command
ADJUST parameter, 4-2
CONSTANT parameter, 4-3
example, 4-6 to 4-7
FORMAT parameter, 4-3
LENGTH parameter, 4-3
LMULT parameter, 4-4
LTHFLD parameter, 4-4
OFFSET parameter, 4-4
parameters, summary, 4-2
points to note, 4-5
PSTAMBLE parameter, 4-5
PREAMBLE parameter, 4-5
sample, 4-7
ZERO parameter, 4-5
BMULT parameter, VOLUME command, 4-28 to 4-29
BOF parameter, 8-19 to 8-20
BSELECT command
description, 7-14
examples, 7-15 to 7-16
points to note, 7-15
INDEX

TEST parameter, 7-14
BSEQ parameter, DJDE, 8-16
BSIDE parameter, DJDE, 8-17
BSKIP parameter, DJDE, 8-17
BTEXT parameter
   DJDE, 8-20 to 8-26
   MESSAGE command, 6-36 to 6-39

C
   C text parameter, DJDE, 8-26
   card-image files, 4-48, 8-48 to 8-50
   carriage control commands, 4-40 to 4-41
   catalog level, JSL, 2-11 to 2-12, 3-4
   cataloged
      CMEs, 6-17
      RTEXT files, 6-83
   change mode, 7-6 to 7-7
   CHANGE parameter, CRITERIA command, 7-3
   channel-attached LPS, 4-39
   character
      code assignment tables, C-1 to C-8
      constants, PDL, 2-6
      set, 3-21
      spacing, 3-21
      types, 7-8
   checkpoint data, 6-6 to 6-7
   CLUSTER.LIB, 5-2
   clusters
      creating, 5-6 to 5-7
      defining
         DJDEs, 5-5
         PDL, 5-5
      description, 5-1 to 5-2
      points to note, 5-5 to 5-6
      stocksets, 5-3 to 5-8
      storing, 5-2 to 5-3
      using, 5-3 to 5-4
   CME
      command
         catalogued, 6-17
         CONSTANT parameter, 6-18
         description, 6-17
         example, 6-21
         FONTS parameter, 6-18 to 6-19
         LINE parameter, 6-19
         parameters, summary, 6-17
         points to note, 6-20 to 6-21
         POSITION parameter, 6-19
         short form, 6-17
         copy-sensitive, 4-41
         description, 1-9
         string constant, font switch, 6-20
   CMD files, executing, 1-5
   CODE command
      ASSIGN parameter, 4-7 to 4-8
      DEFAULT parameter, 4-8
      description, 4-7
      examples, 4-8
      parameters, summary, 4-7
      point to note, 4-8
   CODE parameter, VOLUME command, 4-29
   codes, tape, 4-49
   COLLATE parameter
      DJDE, 8-27
      OUTPUT command, 6-47 to 6-48
   command levels, JSL, 2-10, 3-4 to 3-5
   command structure, PDL
      components
         command keywords, 2-3, 3-5
         comments, 2-4, 3-6
         description, 2-2
         identifiers, 2-3, 3-5
         parameters, 2-4, 3-5
         syntax, 2-4 to 2-5, 3-6
         line length, 2-2
         right-part constants
            string, 2-6 to 2-8
            value, 2-5
   commands, online specific, 4-40 to 4-41
   comments, PDL, 2-4, 3-6
   compiling, JSL, 3-13 to 3-14
   components
      hardware, 1-2
      software, 1-3
   constant mode, 7-6 to 7-7
   CONSTANT parameter
      BLOCK command, 4-3
      CRITERIA command, 7-4
      RECORD command, 4-14
      TABLE command, 7-35
   COPIES parameter
      DJDE, 8-27 to 8-28
      OUTPUT command, 6-48
   COVER parameter, OUTPUT command, 6-48 to 6-49
   CRITERIA command
      CHANGE parameter, 7-3
      CONSTANT parameter, 7-4
      description, 7-2 to 7-3
      examples, 7-7 to 7-8
      LINEDNUM parameter, 7-4
      points to note, 7-7
      test expressions, 7-4 to 7-7
   CYCLEFOMRS parameter, OUTPUT command, 6-49

D
   DATA
      parameter
         DJDE, 8-28
         LINE command, 6-27
      segment, 4-47
   data types
      formatted, 1-10
      unformatted, 1-8 to 1-9
   DEFAULT parameter
      CODE command, 4-8
      PCC command, 4-11
      TCODE command, 4-24 to 4-25
   disk saturation, 8-14
   DJD task, description, 1-6
   DJDE FILE parameter
      deletion, 8-49
      description, 8-47 to 8-48
      effectiveness, 8-48
      replacement, 8-49
INDEX

points to note, 8-48
processing
   card-image files, 8-49 to 8-50
   LPS-labeled files, 8-50 to 8-51
DJDEs
   applying, 8-9 to 8-11
   benefits, 8-1 to 8-2
   COPIES, processing, 8-14
   description, 1-9
   duplex printing, 8-13
   FILE command, 4-48
IDEN command
   description, 1-6, 8-6
   OFFSET parameter, 8-6
   OPRINFO parameter, 8-6 to 8-7
   point to note, 8-7
   PREFIX, 8-7
   SKIP parameter, 8-7
multiple logical pages, 8-13
operator information pages, 8-11
page-oriented, 8-2 to 8-4
parameters
   ALTER, 8-15
   ASSIGN, 8-15 to 8-16
   BSEQ, 8-16
   BSKIP, 8-17
   BSIDE, 8-17
   BATCH, 8-18
   BEGIN, 8-18 to 8-19
   BFORM, 8-19
   BOF, 8-19 to 8-20
   BTEXT, 8-20 to 8-26
   C text, 8-26
   CANCEL, 8-26 to 8-27
   COLLATE, 8-27
   COPIES, 8-27 to 8-28
   DATA, 8-28
   DEPT, 8-28
   DESTINATION, 8-28
   DUFLEX, 8-28
   END, 8-29
   examples, SEFMAP, 8-47
   FEED, 8-29
   FILE, 8-29 to 8-30, 8-47 to 8-51
   FONT INDEX, 8-30
   FONTS, 8-30 to 8-31
   FORMAT, 8-31
   FORMS, 8-32
   GRAPHICS, 8-32 to 8-33
   IMAGE, 8-33 to 8-34
   INVERT, 8-34 to 8-35
   ITEXT, 8-35
   JDE, 8-35
   JDL, 8-35
   MAP, 8-35
   MARGIN, 8-36
   MODIFY, 8-36
   NUMBER, 8-36 to 8-37
   OTEXT, 8-37
   OVERPRINT, 8-37 to 8-38
   points to note, 8-45 to 8-47
   PMODE, 8-38
   RFORM, 8-38
   RTEXT, 8-39
   SAVE, 8-39
   SEFMAP, 8-39 to 8-40
   SEPAREATORS, 9-40
   SF1FUNCTION, 8-41
   SF2FUNCTION, 8-41
   SHIFT, 8-41
   SIDE, 8-42
   SNUMBER, 8-42
   SPLIT, 8-43
   SRECOVER, 8-43 to 8-44
   STING, 8-44
   STOKES, 8-44
   TMODE, 8-44
   TOF, 8-45
   TRAN, 8-45
   XSHIFT, 8-45
   processing
      online, 4-40 to 4-41
      optimization, 8-14
      record-oriented, 8-4 to 8-6
      record specification, 8-8 to 8-9
   restrictions
      modifying job parameters, 8-11 to 8-13
      online, 8-14
   summary, A-1 to A-10
delimiter
display, 7-27
modes, 7-26
DELIMITER parameter, RSTACK command, 7-28
DENSITY parameter, OUTPUT command, 6-50
DESTINATION parameter, 6-50
DEPT parameter, DJDE, 8-28
DESTINATION parameter, DJDE, 8-28
document interleaved, 9-3 to 9-5, 9-7
documentation, other, iii
downloading file, host, 4-47 to 4-48
dump, online, 4-46 to 4-47
duplex printing, DJDE page, 8-13
dynamic job descriptor, see DJD
dynamic job descriptor entries, see DJDEs
E
EBCDIC constants, PDL, 2-7
Editor
   commands, quick reference, E-1 to E-6
   task, description, 1-5
   edgemarking, 3-15
END
   command, 2-13
   parameter, DJDE, 8-29
   EO V parameter, VOLUME command, 4-29 to 4-30
   ERROR parameter, ABNORMAL command, 6-2 to 6-3
   examples
      ACCT command, 6-15
      BANNER command, 7-13 to 7-14
      BDELETE command, 7-15 to 7-16
      BLOCK command, 4-6 to 4-7
INDEX

BSELECT command, 7-15 to 7-16
BTEXT parameter, MESSAGE command, 6-41
CME command, 6-21
CODE command, 4-8
CRITERIA command, 7-7 to 7-8
END command, 2-13
EXPORT command, 6-26
JOB command, 2-12
LINE command, 6-33 to 6-34
MESSAGE command, 6-34
OUTPUT command, 6-75 to 6-76
PCC command, 4-13
PDE command, 6-82
RAUX command, 7-17
REDLETE command, 7-19
RECORD command, 4-17 to 4-19
ROFFSET command, 7-21
ROUTE command, 6-85
RPAGE command, 7-24 to 7-25
RRESUME command, 7-33 to 7-34
RSTACK command, 7-30 to 7-31
RSEND command, 7-33 to 7-34
SEFFNT command, 4-23 to 4-24
STOCKSET command, 6-87 to 6-88
TABLE command, 7-35 to 7-37
TCODE command, 4-27
VOLUME command, 4-38

EXPORT command
  description, 6-21 to 6-22
  examples, 6-26
  parameters, summary, 6-22
  points to note, 6-25 to 6-26
  SEPARATORS parameter, 6-22 to 6-23
  SNUMBER parameter, 6-23
  SPLIT parameter, 6-24
  SRECOVER parameter, 6-24
  STIMING parameter, 6-25

FACEUP parameter, OUTPUT command, 6-51
F
  FACEUP parameter, OUTPUT command, 6-51
  FCB
    3211/4245, 4-42 to 4-43
    parameter, LINE command, 6-27 to 6-28
  FCP, description, 1-5
  FEED parameter
    DJDE, 8-29
    OUTPUT command, 6-51
  file control program, see FCP
  FILE parameter, DJDE, 8-29 to 8-30, 8-47 to 8-51
  fixed spacing, fonts, 3-21 to 3-22
  font
    index bytes, 6-32
    switch, 6-20
  FONTINDEX parameter
    DJDE, 8-30
    LINE command, 6-28
  fonts
    description, 3-21 to 3-22
    mapping files, 4-20 to 4-21
  FONT parameter
    DJDE, 8-30 to 8-31
    PDE command, 6-80 to 6-81
  form
    data, 1-9
    file, see FRM file
  FORMAT parameter
    BLOCK command, 4-3
    DJDE, 8-31
    OUTPUT command, 6-52
    RECORD command, 4-14
  formats
    dump, 4-46
    packed data, 4-49 to 4-50
    record, 4-50
    tape, 4-49
  forms control buffer, see FCB
  FORMS parameter
    DJDE, 8-32
    OUTPUT command, 6-52 to 6-53
  FRM file, FSL compiled, 1-6

G
  graphics
    considerations, 9-1
    DJDE options, 9-6
    input, 9-1
    online formats, 9-4 to 9-5
    PDL command options, 9-6
    performance, 9-7
    processing modes, 9-2 to 9-3
    restrictions, 9-7 to 9-8
    tape formats, 9-3 to 9-4
  GRAPHICS parameter
    DJDE, 8-32 to 8-33
  OUTPUT command, 6-53

H
  H2/H6 constants, PDL, 2-8
  HCOUNT parameter, BANNER command, 7-11
  HEADER segment, 4-46 to 4-47
  hexadecimal, PDL, 2-6
  hierarchical relationships
    operating software, 1-3 to 1-4
  Highlight Color compatibility, 3-24 to 3-26
  hierarchy, replacement, PDL, 2-16 to 2-17
  HJOBNO parameter, BANNER command, 7-11
  Honeywell 2000/6000 methods, 4-49
  host computer
    downloading files, 4-47 to 4-48
    label specification table, D-2
  HOST parameter, VOLUME command, 4-30
  HOSTCOPY utility, 4-48
  HRPTNA parameter, BANNER command, 7-12

I
  IDEN command
    description, 1-6, 8-6
    OFFSET parameter, 8-6
    OPRINFO parameter, 8-6 to 8-7
    point to note, 8-7
    PREFIX, 8-7
    SKIP parameter, 8-7
  ID level, JSL, 2-11 to 2-13
  identifier, PDL, 2-3, 3-5
INDEX

IMAGE parameter
  DJDE, 8-33 to 8-34
  OUTPUT command, 6-54
IMISMATCH parameter, ABNORMAL command, 6-3 to 6-4
INFEED parameter, STOCKSET command, 6-86
INITIAL parameter, PCC command, 4-12
ink mismatch errors, 6-4
input
data types, 1-8 to 1-10
  parameters, specifying, 4-1 to 4-51
  sources, 4-39
  unpacking examples, D-1
input processing tasks
command summary, 4-1
BLOCK command
  ADJUST parameter, 4-2
  CONSTANT parameter, 4-3
  FORMAT parameter, 4-3
  LENGTH parameter, 4-4
  LMULT parameter, 4-4
  OFFSET parameter, 4-4
  parameters, summary, 4-2
  points to note, 4-5
  PSTABLE parameter, 4-5
  PREMABLE parameter, 4-5
  sample, 4-7
  ZERO parameter, 4-5
description, 1-5, 1-7
CODE command
  ASSIGN parameter, 4-7 to 4-8
  DEFAULT parameter, 4-8
description, 4-7
  examples, 4-8
  parameters, summary, 4-7
  point to note, 4-8
PCC command
  ADV parameter, 4-9
  ASSIGN parameter, 4-10 to 4-11
  DEFAULT parameter, 4-11
description, 4-9
  example, 4-13
  INITIAL parameter, 4-12
  MASK parameter, 4-12
  parameters, summary, 4-9
  points to note, 4-12 to 4-13
RECORD command
  ADJUST parameter, 4-14
  CONSTANT parameter, 4-14
description, 4-17 to 4-19
  example, 4-17 to 4-19
  FORMAT parameter, 4-14
  LENGTH parameter, 4-14 to 4-15
  LMULT parameter, 4-15
  LTHFLD parameter, 4-15
  OFFSET parameter, 4-15 to 4-16
  parameters, summary, 4-13
  points to note, 4-17
  PSTABLE parameter, 4-16
  PREMABLE parameter, 4-16
  STRUCTURE parameter, 4-16
SEFFNT command
description, 4-20
elements, 4-23 to 4-24
  MAP parameter, 4-22
  mapping files, 4-20 to 4-21
  parameters, summary, 4-20
  points to note, 4-22 to 4-23
  SEFFNT SEFMAP parameter, 4-21
  SEFMAP parameter, 4-21 to 4-22
TCODE command
default parameter, 4-24 to 4-25
description, 4-24
elements, 4-27
  parameters, summary, 4-24
  points to note, 4-26 to 4-27
  TASSIGN parameter, 4-25
  TRESET parameter, 4-26
VOLUME command
  BMULT parameter, 4-28 to 4-29
  CODE parameter, 4-29
  EO V parameter, 4-29 to 4-30
description, 4-38
  example, 4-38
  HOST parameter, 4-30
  INTERPRESS parameter, 4-31 to 4-32
  LABEL parameter, 4-32
  LCODE parameter, 4-32 to 4-33
  LPACK parameter, 4-33
  MAXLAB parameter, 4-33
  MINLAB parameter, 4-33 to 4-34
  OPTIMIZE parameter, 4-34 to 4-35
  OSCHON parameter, 4-35
  OSHDP parameter, 4-35
  OSTLP parameter, 4-35
  parameters, summary, 4-28
  PLABEL parameter, 4-36
  points to note, 4-38
  RMULT parameter, 4-36
  RSAT parameter, 4-36 to 4-37
  TCODE parameter, 4-37
  UNPACK parameter, 4-37
  Interpress data, magnetic tape, D-4 to D-13
  INTERPRESS parameter, VOLUME command, 4-31 to 4-32
INVERT parameter
  DJDE, 8-34 to 8-35
  OUTPUT command, 6-54 to 6-55
ISUBSTITUTE parameter, ABNORMAL command, 6-5
ITEXT parameter
  DJDE, 8-35
  MESSAGE command, 6-35
J
JDE
  creating, online, 4-40
description, 1-9
  level, JSL, 2-12
  parameter, DJDE, 8-35
JDL
  creating, online, 4-40
description, 1-6
  hierarchy, 2-16 to 2-17
  level, JSL, 2-11, 3-4
parameter, DJDE, 8-35
samples
 offline, 2-14
 online, 2-15
system software tape, D-3
job
 command, example, 2-12
 flow, process, 1-7 to 1-8
 level, JSL, 2-12, 3-4
 parameter, modification restrictions, 8-11 to 8-13
 preparing, 1-6 to 1-7
 printing, 3-14
 queue buffer, 1-7
job descriptor library, see JDL
job queue entry, see JQE
job source language, see JSL
JQE, OCS writing, 1-7
JSL
 application type, 3-3
 compiling, 3-13 to 3-14
 creating, 3-1 to 3-26
 description, 1-6, 2-1, 2-9
 hints/tips, 3-6 to 3-7
 input data, 3-2 to 3-3
 output specifications, 3-3
 procedure steps, 3-8 to 3-13
 structure
 command levels, 2-10
 ID level, 2-11 to 2-13, 3-4
 sample file format, 2-10
K
 keyword
 commands, 2-3, 3-5
 string constants, PDL, 2-6
L
LABEL parameter, VOLUME command, 4-32
landscape orientation, 3-17, 6-81
LCO DE parameter, VOLUME command, 4-32 to 4-33
LENGTH parameter
 BLOCK command, 4-3
 RECORD command, 4-14 to 4-15
LINE command
 DATA parameter, 6-27
 description, 6-26
 examples, 6-33 to 6-34
 FCB parameter 6-27 to 6-28
 FONTINDEX parameter, 6-28
 MARGIN parameter, 6-28 to 6-29
 OVERPRINT parameter, 6-29
 parameters, summary, 6-27
 PCC parameter, 6-29 to 6-30
 PCCTYPE parameter, 6-30
 points to note, 6-31 to 6-33
 UCSB parameter, 6-30 to 6-31
 VFU parameter, 6-31
LINENUM parameter, CRITERIA command, 7-4, 7-6
LMULT parameter
 BLOCK command, 4-4
 RECORD command, 4-15
logical pages, 6-79, 8-13
logical processing commands
 command format, 7-1 to 7-2
 BANNER command
 description, 7-10
 examples, 7-13 to 7-14
 HCOUNT parameter, 7-11
 HJOBNO parameter, 7-11
 HRPTNA parameter, 7-12
 parameters, summary, 7-11
 points to note, 7-13
 TCOUNT parameter, 7-12
 TEST parameter, 7-12
 TYPE parameter, 7-13
 BDELETE command
 description, 7-14
 examples, 7-15 to 7-16
 points to note, 7-15
 TEST parameter, 7-14
 BSELECT command
 description, 7-14
 examples, 7-15 to 7-16
 points to note, 7-15
 TEST parameter, 7-14
 CRITERIA command
 CHANGE parameter, 7-3
 CONSTANT parameter, 7-4
 description, 7-2 to 7-3
 examples, 7-7 to 7-8
 LINENUM parameter, 7-4
 points to note, 7-7
 test expressions, 7-4 to 7-7
 description, 7-1
 RAUX command
 description, 7-16
 example, 7-17
 points to note, 7-16
 TEST parameter, 7-16
 RBAR command, 7-17
 RDELETE command
 description, 7-18
 example, 7-19
 points to note, 7-19
 TEST parameter, 7-18 to 7-19
 RFEDD command
 description, 7-17
 points to note, 7-18
 TEST parameter, 7-17 to 7-18
 ROFFSET command
 description, 7-19
 example, 7-21
 PASSES parameter, 7-20
 points to note, 7-21
 TEST parameter, 7-20
 RPAGE command
 description, 7-21
 examples, 7-24 to 7-25
 parameters, summary, 7-21
 points to note, 7-23
 SIDE parameter, 7-22, 7-24
 TEST parameter, 7-22
 WHEN parameter, 7-23, 7-25
### INDEX

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RESUME</strong> command</td>
<td>BEGIN</td>
<td>parameter, 7-32</td>
</tr>
<tr>
<td></td>
<td>description, 7-31</td>
<td>points to note, 7-32 to 7-33</td>
</tr>
<tr>
<td></td>
<td>example, 7-33 to 7-34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>parameter, 7-32</td>
</tr>
<tr>
<td><strong>RSELECT</strong> command</td>
<td>description, 7-18</td>
<td>points to note, 7-19</td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>parameter, 7-18 to 7-19</td>
</tr>
<tr>
<td><strong>RSTACK</strong> command</td>
<td>ACCTINFO</td>
<td>parameter, 7-28</td>
</tr>
<tr>
<td></td>
<td>accounting page, 7-27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>delimiter display, 7-27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>delimiter modes, 7-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DELIMITER</td>
<td>parameter, 7-28</td>
</tr>
<tr>
<td></td>
<td>description, 7-26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>example, 7-30 to 7-31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HRPTNA</td>
<td>parameter, 7-2</td>
</tr>
<tr>
<td></td>
<td>online usage, 7-27 to 7-28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>points to note, 7-30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
<td>parameter, 7-29</td>
</tr>
<tr>
<td></td>
<td>status display, 7-27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>parameter, 7-30</td>
</tr>
<tr>
<td><strong>RSUSPEND</strong> command</td>
<td>BEGIN</td>
<td>parameter, 7-32</td>
</tr>
<tr>
<td></td>
<td>description, 7-31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>example, 7-33 to 7-34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>points to note, 7-32 to 7-33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST</td>
<td>parameter, 7-32</td>
</tr>
</tbody>
</table>

### MESSAGE command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTEXT</td>
<td>parameter, 6-36 to 6-39</td>
<td></td>
</tr>
<tr>
<td>ITEXT</td>
<td>parameter, 6-35</td>
<td></td>
</tr>
<tr>
<td>OTEXT</td>
<td>parameter, 6-35 to 6-36</td>
<td></td>
</tr>
<tr>
<td>parameters, summary, 6-34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MINLAB</td>
<td>parameter, VOLUME command, 4-33 to 4-34</td>
<td></td>
</tr>
<tr>
<td>MODIFY</td>
<td>parameter</td>
<td></td>
</tr>
<tr>
<td>DJDE</td>
<td>8-36</td>
<td></td>
</tr>
<tr>
<td>OUTPUT command, 6-55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>move mode, 9-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiple logical pages, 6-79, 8-13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multivolume processing, 4-50 to 4-51</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### N

- non-imaged elements, 3-15 to 3-16
- noninterleaved tape format, 9-3 to 9-4
- NTO1 parameter, OUTPUT command, 6-55 to 6-56
- NUMBER parameter |
| DJDE, 8-36 to 8-37 |
| OUTPUT command, 6-56 to 6-57 |

### O

- OCS |
| description, 1-5 |
| processing, 1-7 to 1-8 |
| octal constants, PDL, 2-7 |
| offline |
| mode, 4-49 to 4-51 |
| specifications, D-1 to D-13 |
| OFFSET parameter |
| BLOCK command, 4-4 |
| IDEN command, 8-6 |
| OUTPUT command, 6-57 |
| RECORD command, 4-15 to 4-16 |
| online |
| DJDE restrictions, 8-14 |
| formats, graphics, 9-4 to 9-5 |
| printing systems, 4-39 to 4-48 |
| RSTACK usage, 7-27 to 7-28 |
| operating system executive task, see OSEXEC |
| operating system portion, record, 4-50 |
| operator |
| communication subsystem, see OCS information pages, DJDEs, 8-11 |
| OPINFO parameter, IDEN command, 8-6 to 8-7 |
| OPTIMIZE parameter, VOLUME command, 4-34 to 4-35 |
| optimizing, DJDE processing, 8-14 |
| ordered stocks, clusters, 5-8 |
| SCHN parameter, VOLUME command, 4-35 |
| SEXEC, description, 1-5 |
| SHDP parameter, VOLUME command, 4-35 |
| STK parameter, OUTPUT command, 6-57 to 6-58 |
| STLP parameter, VOLUME command, 4-35 |

### M

- MAP parameter |
| DJDE, 8-35 |
| SEFNT command, 4-22 |
| mapping files, SEFNT command, 4-20 to 4-23 |
| MARGIN parameter |
| DJDE, 8-36 |
| LINE command, 6-28 to 6-29 |
| marker page, 6-7 to 6-8 |
| MASK parameter |
| PCC command, 4-12 |
| TABLE command, 7-35 |
| masked comparisons, 7-9 |
| MAXLAB parameter, VOLUME command, 4-33 |
| memory |
| font, 3-22 |
| graphic, 3-22 |
| Merganthaler fonts, 6-81 |
INDEX

O TEXT
application, clusters, 5-3
parameter
ABNORMAL command, 6-5
DJDE, 8-37
MESSAGE command, 6-35 to 6-36

OUTPUT command
BFORM parameter, 6-47
COLLATE parameter, 6-47 to 6-48
COPIES parameter, 6-48
COVER parameter, 6-48 to 6-49
CYCLEFORMS parameter, 6-49
DENSITY parameter, 6-50
description, 6-45
DESTINATION parameter, 6-50
dUplex parameter, 6-50 to 6-51
dexamples, 6-75 to 6-76
FACEUP parameter, 6-51
FEED parameter, 6-51
FORMAT parameter, 6-52
FORMS parameter, 6-52 to 6-53
GRAPHICS parameter, 6-53
IMAGE parameter, 6-54
INVERT parameter, 6-54 to 6-55
MODIFY parameter, 6-55
NTO1 parameter, 6-55 to 6-56
NUMBER parameter, 6-56 to 6-57
OFFSET parameter, 6-57
OSTK parameter, 6-57 to 6-58
PAPERSIZE parameter, 6-58 to 6-59
parameters, summary, 6-46
points to note, 6-66 to 6-74
PURGE parameter, 6-59
RESOLUTION parameter, 6-59 to 6-60
SF1FUNCTION parameter, 6-60
SF2FUNCTION parameter, 6-60
SHIFT parameter, 6-60 to 6-61
SIZING parameter, 6-61 to 6-62
STOCKS parameter, 6-62
SYSPPR parameter, 6-63
TMODE parameter, 6-63 to 6-64
TRANS parameter, 6-64
UNITs parameter, 6-64
XSHIFT, 6-65
XMP, 6-65
output processing task, description, 1-6
OVERPRINT parameter
DJDE, 8-38
LINE command, 6-29
override line spacing, 6-81

P
page
buffer, 1-6
considerations, JSL, 3-15 to 3-16
interleaved, 9-4
log, 1-6
orientations, 3-16 to 3-18
page-oriented DJDEs, 8-2 to 8-4
PAGES parameter, ABNORMAL command, 6-5 to 6-6
paper sizes, 3-15 to 3-16
PAPERSIZE parameter, OUTPUT command,
6-58 to 6-59
parameters, PDL, 2-4, 3-5
PASSES parameter, OFFSET command, 7-20
PCC
command
ADV parameter, 4-9
ASSIGN parameter, 4-10 to 4-11
DEFAULT parameter, 4-11
description, 4-9
element, 4-13
INITIAL parameter, 4-12
MASK parameter, 4-12
parameters, summary, 4-9
points to note, 4-12 to 4-13
parameter, LINE command, 6-29 to 6-30
PCCTYPE parameter, LINE command, 6-30
PDE command
BEGIN parameter, 6-80
description, 6-77
dexamples, 6-82
FONTs parameter, 6-80 to 6-81
multiple logical pages, 6-79
parameter, summary, 6-77
PMODE parameter, 6-81
points to note, 6-81 to 6-82
print formats, standard, 6-78
PDL
command structure
components, 2-2 to 2-5, 3-5 to 3-6
line length, 2-2
right-part constants, 2-5
sample, 2-9
description, 1-1, 2-1
grouping commands, 2-16
processor, description, 1-6
quick reference, commands, B-1 to B-18
replacement hierarchy, 2-16 to 2-17
summary, commands, A-1 to a-10
tsyntax, 3-6
physical page, 3-15
pitch mode, 6-71 to 6-74
PLABEL parameter, VOLUME command, 4-36
PMODE parameter
DJDE, 8-38
PDE command, 6-81
portrait orientation, 3-18, 6-78
POSTAMBLE parameter
BLOCK command, 4-5
RECORD command, 4-16
PPI, 4-43
PREAMBLE parameter
BLOCK command, 4-5
RECORD command, 4-16
PREFIX parameter, IDEN command, 8-7
print
formats, standard, 6-78
job, preparing, 1-6 to 1-7
print description language, see PDL
print format commands
  ABNORMAL command
    description, 6-2
    ERROR parameter, 6-2 to 6-3
    IMISMATCH parameter, 6-3 to 6-4
    SUBSTITUTE parameter, 6-5
    marker page, 6-7 to 6-8
    OTEXT parameter, 6-5
    PAGES parameter, 6-5 to 6-6
    parameters, summary, 6-2
    recovery, 6-6 to 6-7
    REP parameter, 6-6
    SECURITY parameter, 6-6
  ACCT command
    accounting data, 6-9, 6-12
    DEPT parameter, 6-14
    description, 6-9 to 6-10
    example, 6-15
    installation accounting, 6-10 to 6-11
    overall usage, 6-9, 6-13
    parameters, summary, 6-9
    points to note, 6-15
    USER parameter, 6-14
  Audit Log, points to note, 6-42 to 6-45
  BARCODE command
    description, 6-15
    points to note, 6-16 to 6-17
  CM E command
    catalogued, 6-17
    CONSTANT parameter, 6-18
    description, 6-17
    example, 6-21
    FONTS parameter, 6-18 to 6-19
    LINE parameter, 6-19
    parameters, summary, 6-17
    points to note, 6-20 to 6-21
    POSITION parameter, 6-19
    short form, 6-17
    description, 6-1
  EXPORT command
    description, 6-21 to 6-22
    examples, 6-26
    parameters, summary, 6-22
    points to note, 6-25 to 6-26
    SEPARATORS parameter, 6-22 to 6-23
    SN U M B ER parameter, 6-23
    SPLIT parameter, 6-24
    SRECOVER parameter, 6-24
    STIMING parameter, 6-25
    LINE command
    DATA parameter, 6-27
    description, 6-26
    examples, 6-33 to 6-34
    FCB parameter 6-27 to 6-28
    F O N TINDEX parameter, 6-28
    OVERPRINT parameter, 6-29
    parameters, summary, 6-27
    PCC parameter, 6-29 to 6-30
    PCCTYPE parameter, 6-30
    points to note, 6-31 to 6-33
    UCSB parameter, 6-30 to 6-31
    VFU parameter, 6-31
    MESSAGE command
    BTEXT parameter, 6-36 to 6-39
    description, 6-34
    example, 6-34
    IT E X T parameter, 6-35
    OT EX T parameter, 6-35 to 6-36
    parameters, summary, 6-34
  OUTPUT command
    BFORM parameter, 6-47
    COLLATE parameter, 6-47 to 6-48
    COPIES parameter, 6-48
    COVER parameter, 6-48 to 6-49
    CYCLEF O RMS parameter, 6-49
    DENSITY parameter, 6-50
    description, 6-45
    DESTINATION parameter, 6-50
    D U P LEX parameter, 6-50 to 6-51
    examples, 6-75 to 6-76
    FACEUP parameter, 6-51
    FEED parameter, 6-51
    FORMAT parameter, 6-52
    FORMS parameter, 6-52 to 6-53
    GRAPHICS parameter, 6-53
    IMAGE parameter, 6-54
    INVERT parameter, 6-54 to 6-55
    MODIFY parameter, 6-55
    NTO1 parameter, 6-55 to 6-56
    NUMBER parameter, 6-56 to 6-57
    OFFSET parameter, 6-57
    O ST K parameter, 6-57 to 6-58
    PAPERSIZE parameter, 6-58 to 6-59
    parameters, summary, 6-46
    points to note, 6-66 to 6-74
    PURGE parameter, 6-59
    RESOLUTION parameter, 6-59 to 6-60
    SF1FUNCTION parameter, 6-60
    SF2FUNCTION parameter, 6-60
    SHIFT parameter, 6-60 to 6-61
    SIZING parameter, 6-61 to 6-62
    STO CKS parameter, 6-62
    SYSPR parameter, 6-63
    TMODE parameter, 6-63 to 6-64
    TRANS parameter, 6-64
    UNITS parameter, 6-64
    XSHIFT, 6-65
    XMP, 6-65
    parameter rules, 6-39 to 6-40
  PDE command
    BEGIN parameter, 6-80
    description, 6-77
    examples, 6-82
    FONTS parameter, 6-80 to 6-81
    multiple logical pages, 6-79
    parameter, summary, 6-77
    PMODE parameter, 6-81
    points to note, 6-81 to 6-82
    print formats, standard, 6-78
    points to note, 6-40 to 6-42
  ROUTE command
    cataloged RTEXT files, 6-83
    description, 6-82
    example, 6-85
parameters, summary, 6-83
RFORM parameter, 6-83 to 6-84
RTEXT parameter, 6-84 to 6-85

STOCKSET command
ASSIGN parameter, 6-86
description, 6-85
test example, 6-87 to 6-88
INIFEED parameter, 6-86
parameters, summary, 6-86
point to note, 6-87
SYSPAGE parameter, 6-87
summary, 6-1

VFU command
ASSIGN parameter, 6-89
BOF parameter, 6-89
description, 6-88
test example, 6-90
parameters, summary, 6-89
point to note, 6-90
TOF parameter, 6-90

print position indexing, see PPI
printer carriage control channels, 6-88
printing systems, online, 4-39 to 4-48
procedures
clusters, creating, 5-6 to 5-7
JSL, creating, 3-8 to 3-13
OTEXT message, creating, 6-36
processing
data, 1-9
modes, graphics, 9-2 to 9-3
proportional spacing, fonts, 3-21 to 3-22
publications, related, iii
PURGE parameter, OUTPUT command, 6-59

R
random mode, 9-2, 9-7
raster image processor, see RIP

RAUX command
description, 7-16
test example, 7-17
point to note, 7-16
TEST parameter, 7-16

RBAR command, 7-17

RDELETE command
description, 7-18
test example, 7-19
point to note, 7-19
TEST parameter, 7-18 to 7-19

record
format types, 4-19, 4-50
length, online, 4-43 to 4-44
specification, DJDEs, 8-8 to 8-9
structure, 4-50

RECORD command
ADJUST parameter, 4-14
CONSTANT parameter, 4-14
test example, 4-17 to 4-19
FORMAT parameter, 4-14
LENGTH parameter, 4-14 to 4-15
RMULT parameter, 4-15
LTHFLD parameter, 4-15
OFFSET parameter, 4-15 to 4-16

parameters, summary, 4-13
point to note, 4-17
POSTAMBLE parameter, 4-16
PREAMBLE parameter, 4-16
STRUCTURE parameter, 4-16
record-oriented DJDEs, 8-4 to 8-5

recovery
ABNORMAL command, 6-6 to 6-7
online, 4-46

RFEED command
description, 7-17
point to note, 7-18
TEST parameter, 7-17 to 7-18

RFORM parameter
DJDE, 8-38
ROUTE command, 6-83 to 6-84

registration
shift, 3-19 to 3-21
skew, 3-19 to 3-21

REP parameter, ABNORMAL command, 6-6
replacement hierarchy, PDL, 2-16 to 2-17

Report
separation, 4-41
task, see RPT

RESOLUTION parameter, OUTPUT command, 6-59 to 6-60
RIP, 1-2, 1-6
RMULT parameter, VOLUME command, 4-36

OFFSETSET command
description, 7-19
test example, 7-21
PASSES parameter, 7-20
point to note, 7-21
TEST parameter, 7-20

ROUTE command
cataloged RTEXT files, 6-83
description, 6-82
test example, 6-85
parameters, summary, 6-83
RFORM parameter, 6-83 to 6-84
RTEXT parameter, 6-84 to 6-85

RPAGE command
description, 7-21
test examples, 7-24 to 7-25
parameters, summary, 7-21
point to note, 7-23
SIDE parameter, 7-22, 7-24
TEST parameter, 7-22
WHEN parameter, 7-23, 7-25

RPT, description, 1-5

RESUME command
BEGIN parameter, 7-32
description, 7-31
test example, 7-33 to 7-34
point to note, 7-32 to 7-33
TEST parameter, 7-32

RSAT parameter, VOLUME command, 4-36

RSELECT command
description, 7-18
point to note, 7-19
TEST parameter, 7-18 to 7-19
INDEX

RSTACK command
   ACCTINFO parameter, 7-28
   accounting page, 7-27
   delimiter display, 7-27
   delimiter modes, 7-26
   DELIMITER parameter, 7-28
   description, 7-26
   example, 7-30 to 7-31
   HRPTNA parameter, 7-2
   online usage, 7-27 to 7-28
   points to note, 7-30
   PRINT parameter, 7-29
   status display, 7-27
   TEST parameter, 7-30
RSUSPEND command
   BEGIN parameter, 7-32
   description, 7-31
   example, 7-33 to 7-34
   points to note, 7-32 to 7-33
   TEST parameter, 7-32
RSX11 host type, 6-21
RTEXT parameter
   DJDE, 8-39
   ROUTE command, 6-84 to 6-85
S
   sample
      accounting page, job, 6-12
      BLOCK command, 4-7
      commands, PDL, 2-9
      customer usage accounting report, 6-13
      data record structure, 6-33
      JDL file, online, 4-45
      RECORD command, 4-18
      RSTACK command, 7-31
SAVE parameter, DJDE, 8-39
SECURITY parameter, ABNORMAL command, 6-6
SEFFNT command
   description, 4-20
   examples, 4-23 to 4-24
   mapping files, 4-20 to 4-21
   parameters, summary, 4-20
   MAP parameter, 4-22
   points to note, 4-22 to 4-23
   SEFFNT SEFMAP parameter, 4-21
   SEFMAP parameter, 4-21 to 4-22
SEFFNT SEFMAP parameter, SEFFNT command, 4-21
SEFMAP parameter
   DJDE, 8-39 to 8-40
   SEFFNT command, 4-21 to 4-22
SEPARATORS parameter
   DJDE, 8-40
   EXPORT command, 6-22 to 6-23
   sequence ##, 6-18
SF1FUNCTION parameter
   DJDE, 8-41
   OUTPUT command, 6-60
SF2FUNCTION parameter
   DJDE, 8-41
   OUTPUT command, 6-60
   shift, registration, 3-19 to 3-21
SHIFT parameter
   DJDE, 8-41
   OUTPUT command, 6-60 to 6-61
   short form CME specifications, 6-17
   short-edge feed (SEF), 4-20
SIDE parameter
   DJDE, 8-42
   RPAGE command, 7-22, 7-24
SIZING parameter, OUTPUT command, 6-61 to 6-62
   skew, registration, 3-19 to 3-21
SKIP parameter, IDEN command, 8-7
SNUMBER parameter
   DJDE, 8-42
   EXPORT command, 6-23
   software, hierarchical relationship, 1-3 to 1-4
SPLIT parameter
   DJDE, 8-43
   EXPORT command, 6-24
   spot-carbon, 6-17
SRECOVER parameter
   DJDE, 8-43 to 8-44
   EXPORT command, 6-24 to 6-25
START command
   identifying JDE/JDL, 1-7
   non-JDL hierarchy, 2-17 to 2-20
   status display, RSTACK command, 7-27
STIMING parameter, DJDE, 8-44
STOCKS parameter
   DJDE, 8-44
   OUTPUT command, 6-62
   stockset application, clusters, 5-3 to 5-8
STOCKSET command
   ASSIGN parameter, 6-86
   description, 6-85
   example, 6-87 to 6-88
   INIFEED parameter, 6-86
   parameters, summary, 6-86
   point to note, 6-87
   SYSPAGE parameter, 6-87
string
   comparisons
      concepts, 7-8
      character types, 7-8
      masked, 7-9
      parameters, 7-10
   constants, PDL, 2-6
STRUCTURE parameter, RECORD command, 4-16
   syntax, PDL, 2-4 to 2-5, 3-6
SYSPAGE parameter, STOCKSET command, 6-87
SYSPPR parameter, OUTPUT command, 6-63
system
   level, JSL, 2-11, 3-4
   page, 3-15
   rollover, 4-50 to 4-51
   software tape, JDLs, D-3
T
TABLE command
   CONSTANT parameter, 7-35
   description, 7-34
   examples, 7-35 to 7-37
   MASK parameter, 7-35
tabs, printing, 6-68 to 6-70

tape
  codes, 4-49
  formats, 4-49, 9-3 to 9-4, D-5 to D-7
  multivolume processing, 4-50
  packed data formats, 4-49 to 4-50
  record
    formats, 4-50
    structure, 4-50
  structure, concepts, 4-49

TASSIGN parameter, TCODE command, 4-25

TCODE command
  DEFAULT parameter, 4-24 to 4-25
description, 4-24
test expression definition, 4-24
test expression definition, 4-27
test expression definition, 4-27

examples, 4-27

parameters, summary, 4-24
test expression definition, 4-27
TASSIGN parameter, 4-25
TRESET parameter, 4-26

TCODE parameter, VOLUME command, 4-37

TCOUNT parameter, BANNER command, 7-12
test expression definition, 7-4 to 7-5

TEST parameter
  BANNER command, 7-12
  BDELETE command, 7-14
  BSELECT command, 7-14
  RAUX command, 7-16
  RDELETE command, 7-17 to 7-18
  RFEED command, 7-17 to 7-18
  ROFFSET command, 7-20
  RPAGE command, 7-22
  RRESUME command, 7-32
  RSELECT command, 7-18 to 7-19
  RSTACK command, 7-30
  RSUSPEND command, 7-32

TMODE parameter
  DJDE, 8-44
  OUTPUT command, 6-63 to 6-64

TOF parameter, DJDE, 8-45

TRANS parameter
  DJDE, 8-45
  OUTPUT command, 6-64

translate tables, 4-41

transparencies, see TRANS parameter

TRESET parameter, TCODE command, 4-26

TYPE parameter, BANNER command, 7-13

UCSB parameter, LINE command, 6-30 to 6-31

UCSBs, 4-39, 4-41 to 4-42

UNITS parameter, OUTPUT command, 6-64
distribute character set buffers, see UCSBs

UNPACK parameter, VOLUME command, 4-37

table's portion, record, 4-50

V
  variable
    data, PDL, 1-9
    name, PDL, 2-6

  value constants, PDL, 2-5
distribute format control processing, 4-42 to 4-43

VFU
  command
    ASSIGN parameter, 6-89
    BOF parameter, 6-89
description, 6-88
test expression definition, 6-90

  example, 6-90

parameters, summary, 6-89
test expression definition, 6-90

TOF parameter, 6-90

parameter, LINE command, 6-31

VOLUME command
  BMULT parameter, 4-28 to 4-29
  CODE parameter, 4-29
test expression definition, 4-30
test expression definition, 4-38

  EOV parameter, 4-29 to 4-30
  test expression definition, 4-38

  HOST parameter, 4-30
test expression definition, 4-38

  INTERPRESS parameter, 4-31 to 4-32
  test expression definition, 4-38

  LABEL parameter, 4-32
test expression definition, 4-38

  LCODE parameter, 4-32 to 4-33
  test expression definition, 4-38

  LPACK parameter, 4-33
test expression definition, 4-38

  MAXLAB parameter, 4-33
test expression definition, 4-38

  MINLAB parameter, 4-33 to 4-34
test expression definition, 4-38

  OPTIMIZE parameter, 4-34 to 4-35
  test expression definition, 4-38

  OSCHON parameter, 4-35
  test expression definition, 4-38

  OSHDP parameter, 4-35
  test expression definition, 4-38

  OSTLP parameter, 4-35
  test expression definition, 4-38

parameters, summary, 4-28

  PLABEL parameter, 4-36
  test expression definition, 4-38

  points to note, 4-38

  RMULT parameter, 4-36
  test expression definition, 4-38

  RSAT parameter, 4-36 to 4-37
  test expression definition, 4-38

  TCODE parameter, 4-37
test expression definition, 4-38

  UNPACK parameter, 4-37
test expression definition, 4-38

W
  WHEN parameter, RPAGE command, 7-23, 7-25

X
  xdots, 6-82, 8-31

  xerographic mode switching, see XMS

XSHIFT parameter
  DJDE, 8-45 to 8-47
  test expression definition, 8-47

XMP parameter, OUTPUT command, 6-65

XMS, 8-7

Z
  ZERO parameter, BLOCK command, 4-5