

Xerox 4850/4890 HighLight Color Laser Printing System Forms Creation Reference

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

1994 by Xerox Corporation. All rights reserved.

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

Printed in the United States of America

Publication number: 720P93650

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

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

This document was created on the Xerox 6085 Professional Computer System using VP software. The typefaces used are Optima, Terminal, and Monospace.

Table of contents

Introduction	vii
Document conventions	vii
1. Overview	1-1
2. Basic concepts	2-1
Page orientation	2-1
Fonts	2-2
Page frames	2-3
Physical page	2-4
System page	2-4
Virtual page	2-4
Form origin	2-5
Positioning form elements	2-6
Grids	2-7
x and y coordinates	2-7
Predefined formats	2-8
Data types	2-9
3. Forms description language	3-1
FDL command overview	3-1
Command format	3-1
Form creation process	3-1
Setup commands	3-2
FORM command	3-3
PAPER command	3-3
LANDSCAPE/PORTRAIT command	3-4
GRID command	3-5
FONT command	3-7
ICATALOG command	3-7
PALETTE command	3-9
INK command	3-9
INK command syntax	3-10
Substituting inks	3-11
Description commands	3-12
LINE command	3-13

	BOX command	3-17
	TEXT commands	3-19
	LOGO command	3-28
	GRAPHIC command	3-29
	IRESULT command	3-31
	SECTION command	3-31
	COMMENT command	3-33
	END command	3-33
4.	Compiling and printing forms	4-1
	Form printing process	4-1
	FSL data transfer	4-1
	Compiling a form	4-2
	Compilation options	4-4
	FDL compilation time	4-6
	Printing a compiled form	4-6
5.	Forms layout considerations	5-1
	Image complexity	5-1
	Line tables	5-1
	Scan line density	5-3
	Superimposition	5-4
	Page generation errors	5-5
	Local density and page setup errors	5-5
	Summary	5-6
	Using boxes	5-6
	Text in boxes	5-7
	Locating the closest box	5-7
	Default color	5-10
	Shading	5-10
	Sections	5-11
	Rounding measurements	5-12
	Converting other unit values to dots	5-12
	Rounding variable data	5-13
	Suggested coding techniques	5-14
	Converting preprinted forms	5-14
	Designing new forms	5-14
	Recommended coding sequence	5-14
	Syntax ambiguities	5-16
6.	Coding a complex form	6-1
	Getting started	6-1

Writing the setup commands	6-2
Using the COMMENT command	6-2
Drawing lines	6-2
Drawing boxes	6-4
Placing text at a location	6-5
Placing text in a box	6-5
END command	6-7
Compiling the form	6-8

Appendices

A. FDL command summary	A-1
B. FDL-generated messages	B-1
C. System default summary	C-1
D. GRID and TEXT command examples	D-1
E. FDL capacity limits	E-1
F. Support tools and measurements	F-1

Glossary

GLOSSARY-1

Index

INDEX-1

This reference is intended for forms designers who have basic knowledge of the 4850 or the 4890 HighLight Color Laser Printing System (LPS). You may have received on-site training, or you may have attended a Xerox forms description language (FDL) training course.

Forms designer's tasks consists of writing FDL commands to produce electronic forms. These tasks include entering source code into the LPS to create new electronic forms, as well as updating, revising, and printing the forms on the laser printing system. If you are also responsible for printing variable data on the form in addition to performing other operations on the printer, refer to the related documents provided with your laser printing system.

Document conventions

This guide uses the following conventions:

- < > Angle brackets are used for keys on the system controller keyboard.
- { } Curly brackets are used for required characters.
- ... Ellipses indicate that you can repeat a parameter or list a series of parameters.
- [] Square brackets are used for optional command characters.
- | Vertical bars are used to separate parameters in a series. The vertical bar stands for "or."
- bold** Bold is used for characters you enter at the command line.
- italics* Italics is used for variable information.
- `TERMINAL FONT` Terminal or monospace font is used to display system responses.
- underline System default parameters are underlined.
- UPPERCASE Uppercase letters are used for command names.

CAUTION: Cautions alert you to an action that could damage hardware or software.

WARNING: Warnings alert you to conditions that may affect the safety of people.

The 4850 and 4890 HighLight Color Laser Printing Systems (LPS) provide the capability to create electronic forms using a variety of colors, shades, and tints. This feature supports the centralized printing and publishing markets in creating highlight color printing applications. The term “highlight color” refers to black plus one other color, such as red, blue or green. The 4850 and 4890 LPS also supports black-only printing and is compatible with the entire family of centralized laser printing systems.

You create electronic forms using forms description language (FDL). FDL enables you to precisely define forms electronically so they can be stored and printed on a Xerox laser printing system. FDL-defined forms are input to the printer as data, using the LPS keyboard and display or a host computer terminal.

To define a form, FDL uses elements such as lines, logos, images, signatures, shading, and different font styles and sizes, to make full use of laser printing system features and color capability. FDL provides a quick and easy means of designing a form electronically to achieve maximum visual impact and readability.

Advantages of FDL include the following:

- Lines can be drawn at specified intervals without being redefined each time. You may choose to have the lines printed in color or black only.
- You state the origin and the dimensions of a box to have it drawn at any specified location on the page.
- An entire section of a form, once defined, can be repeated anywhere on the same form.
- In defining a location on a page, you are not restricted to lines and character positions. You can specify coordinates in inches, centimeters, or dots with a resolution of 1/300 inch in either direction.
- Once you create a form, it can be stored on the system and printed—as many as you need and as often as you need.
- You can use three types of lines (solid, broken, and dotted) in four thicknesses: invisible [0], hairline, medium [1], or bold [2]. You can also mix fonts and arrange text in many ways.

Certain basic concepts are required to understand the forms creation process on laser printing systems. These involve the size, shape, and location of the overall image on the page of a document, the orientation of the text or graphics on the page, the size and style of the characters to be used, the type of data to be entered, and the use of highlight color.

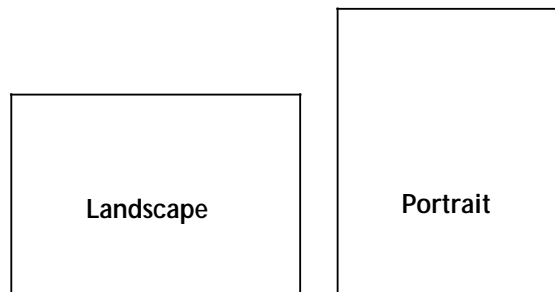
Page orientation

There are two types of page, or text, orientation:

- Portrait page
- Landscape page.

In portrait orientation, the vertical side of the page is longer than the horizontal side. In landscape orientation, the horizontal side of the page is longer than the vertical side. Portrait and landscape orientations are illustrated in figure 2-1.

Figure 2-1. **Page orientation**

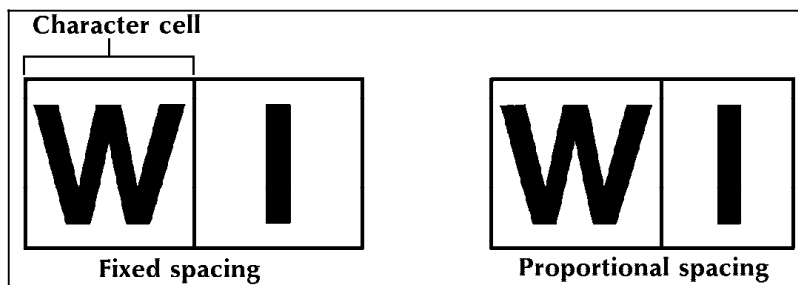


Landscape is the usual orientation for computer reports. Portrait is the usual orientation for letters, manuals, and other text printing. On a laser printing system, you can change page orientation from page to page without interrupting the printing operation. This capability, coupled with variable character size, permits maximum flexibility for the effective presentation of information.

Fonts

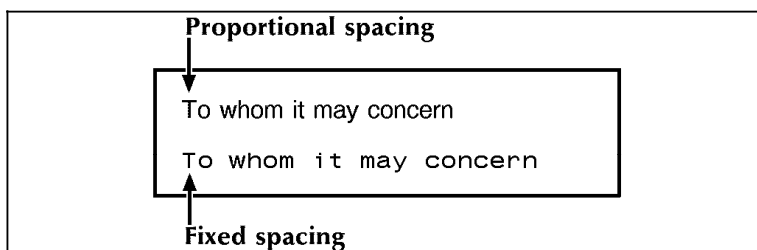
Fonts are character sets, each having a unique type style, type size, and orientation. Fixed and proportionally spaced fonts are available for use on laser printing systems. Each character occupies an area on the form called a character cell. With fixed fonts, all character cells in the set are the same width. With proportional fonts, character cells vary in width, as shown in figure 2-2.

Figure 2-2. 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 forms data with variable data. Proportional fonts are normally used for forms data, that is, titles, headings, and so forth. A letter is an example of the use of proportional fonts for variable data. The differences in width are illustrated in figure 2-3.

Figure 2-3. Character spacing examples



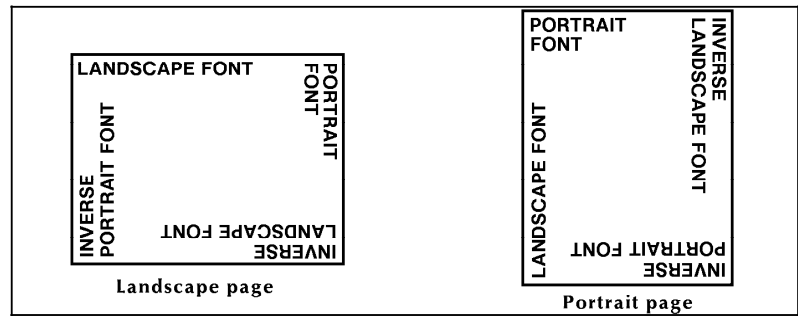
Fonts are available in various typefaces (for example, OCR and Titan), sizes, styles (for example, serif and sans serif), and weights (for example, medium and bold).

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

- Landscape font
- Portrait font
- Inverse landscape font
- Inverse portrait font.

Font orientation, as shown in figure 2-4, is relative to the physical page.

Figure 2-4. **Font orientation**



Even though type style and size are the same, four different fonts are used, each with a unique orientation. The inverse portrait font could be used for certain types of titles or labels on a landscape form. Conversely, the inverse landscape font could be used for titles on a portrait form.

The *LPS Standard Font Library Font User Guide* and the *4135 and 4850 Font User Guide* list the standard fixed and proportional fonts, as well as five special "core" fonts provided with the 4850 and 4890. Special fonts not included in the standard font library may be ordered from the Xerox Font Center or Rank Xerox. (Refer to your *Xerox 4850 HighLight Color LPS Installation Planning Guide*, or *Xerox 4890 HighLight Color LPS Installation Planning Reference* for information on ordering fonts.)

Page frames

A page frame is a set of boundaries associated with a page as a unit of printing or imaging. Three page frames are defined in this reference:

- Physical page
- System page
- Virtual page.

Physical page

The paper upon which a form is printed is the “physical page.” The default paper size used on an LPS is defined during system generation (sysgen). The following paper sizes may be specified:

- 8.5 by 11 inches (USLETTER)
- 8.5 by 14 inches (USLEGAL)
- 8.27 by 11.69 inches (A4)
- 8.0 by 10.02 inches (203 by 254 mm)
- 8.0 by 10.5 inches (203 by 266 mm)
- 8.0 by 13.0 inches (203 by 330 mm)
- 8.27 by 10.63 inches (210 by 270 mm)
- 8.27 by 13.0 inches (210 by 330 mm)
- 8.37 by 10.78 inches (212 by 273 mm)
- 8.46 by 10.83 inches (214 by 275 mm)
- 8.46 by 12.40 inches (214 by 314 mm)
- 8.46 by 14.02 inches (214 by 356 mm)
- 8.50 by 10.75 inches (215 by 273 mm)
- 8.50 by 13.0 inches (215 by 330 mm)

The PAPER command may be used to identify the paper size. If it is not present, the sysgen value is assumed.

When variable data is to be merged with a form, the paper size for both must match and the correct size paper must be loaded.

System page

The maximum area in which graphics and text can be imaged is called the “system page.”

To accommodate edgemarking, which is the placement of marks along the edge of the physical page, the system page must be larger than the physical page. Edgemarking capability is limited because the system page boundaries correspond to the physical page on at least two edges for all paper sizes.

Virtual page

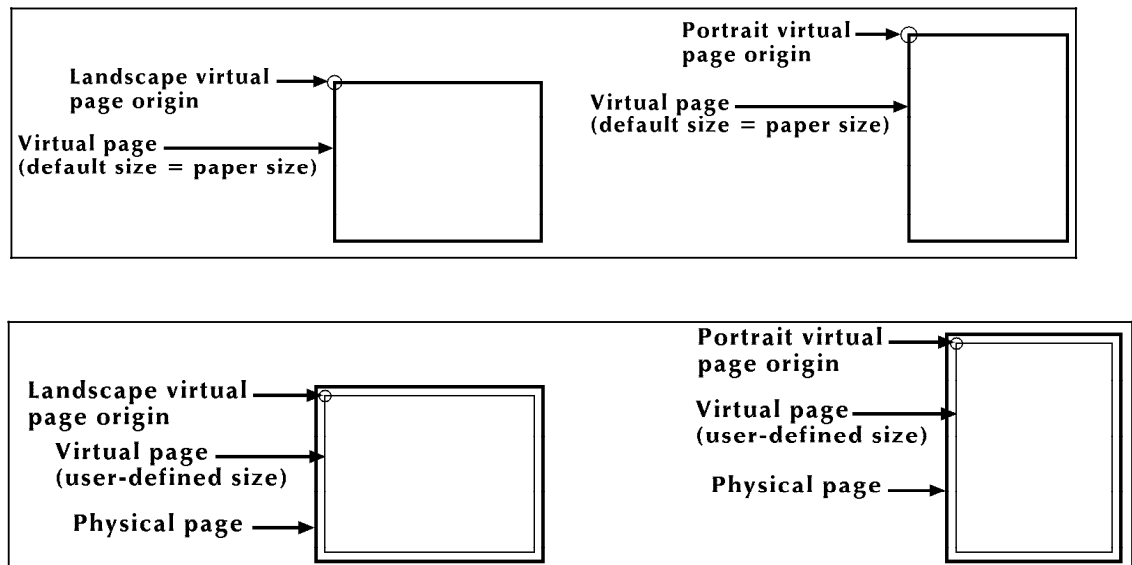
The area chosen for printing is called the “virtual page” and may or may not be as large as the system page. If you do not define its dimensions explicitly, the virtual page dimensions are, by default, the same as the sysgen-specified paper size or as the PAPER command paper size.

If you explicitly define virtual page size (using the LANDSCAPE/PORTRAIT command PAGE SIZE parameter), the defined area is centered relative to the physical page dimensions.

The virtual page is assigned a landscape orientation by default, or you can specify an orientation either with the LANDSCAPE/PORTRAIT command or by using the FMT*n* option of the GRID command.

The upper left corner of the virtual page is called the “virtual page origin” (Figure 2-5). The virtual page origin is also used to establish a form origin.

Figure 2-5. Virtual page origin

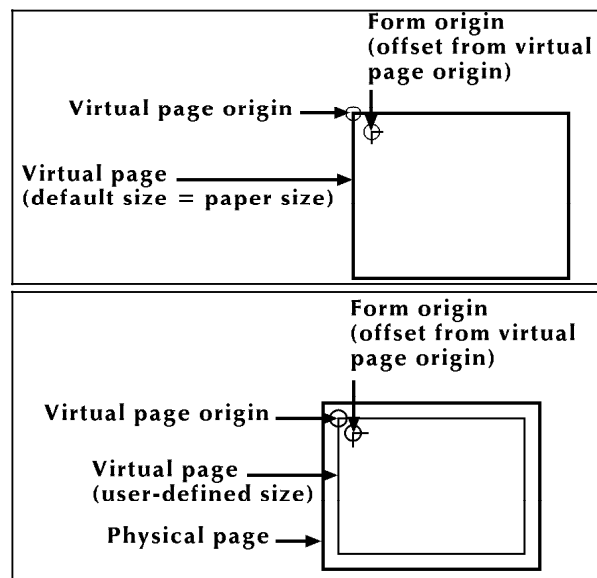


Form origin

All forms data described by FDL commands is positioned relative to a point called the "form origin" (figure 2-6). This point is offset from the virtual page corner by horizontal and vertical displacement values, which you specify using the GRID command. If none are specified, standard default values are used to establish the form origin.

If variable data is merged with the form, set the form origin to coincide with the beginning position of the variable data so the form and variable data coordinate accurately. Standard computer printing formats are stored on the system. Refer to the "Predefined formats" section later in this chapter.

Figure 2-6. Form origin



Positioning form elements

Elements you can place on a form using FDL commands are as follows:

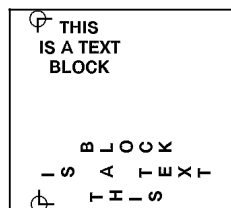
- Lines
- Boxes
- Fixed text
- Logos and signatures
- Images
- Sections.

All of these elements are located in relation to the form origin at the upper left corner of the form. Each form element has an origin, a point used to position it relative to the form origin.

Form element	Element origin
Vertical ruled line	Top of the line at midpoint of the line thickness
Horizontal ruled line	Left end of the line at midpoint of the line thickness
Box (outlined)	Upper left corner of the box at midpoint of the outline thickness
Box (shaded)	Upper left corner of the box
Fixed text (single line)	Upper left corner of the first character cell when line appears upright to the viewer
Fixed text (multiple lines)	Upper edge of the topmost character cell and the leftmost edge of the leftmost character cell when viewed in an upright position (illustrated in figure 2-7)
Graphics	Upper left corner of the image
Logos and signatures	Upper left corner of the first (or only) character cell (with a few exceptions)
Sections (of a form)	Upper left corner of the section

Figure 2-7 shows two text blocks, one using a portrait font and one using a landscape font. A symbol shows where the text block origin is located. This is the point used to position the text block relative to the form origin.

Figure 2-7. Text block origin



Grids

The location of a form element on a page is specified in terms of its horizontal and vertical displacement from the form origin. The units of measurement used to define this displacement can be any of the following:

- Linear units—inches or centimeters
- Dots—300 per inch
- Cpi and lpi—characters per inch horizontally and lines per inch vertically.

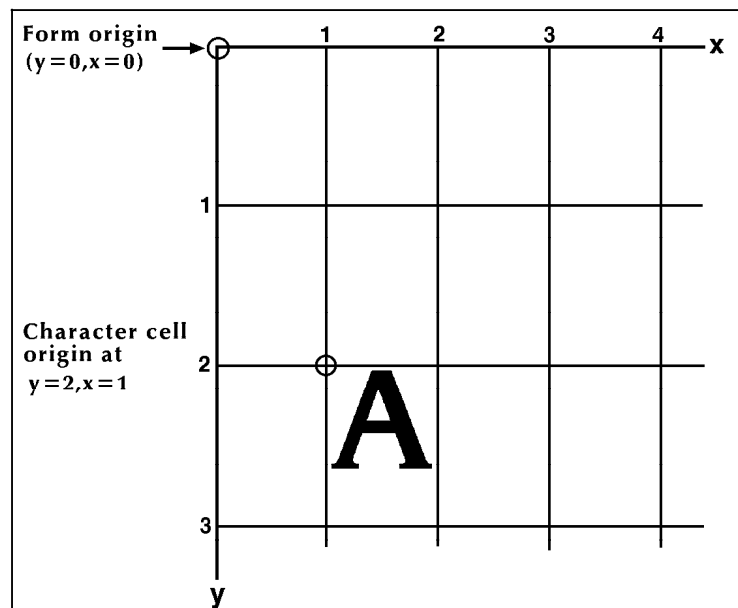
x and y coordinates

The y coordinate describes the vertical position on a grid. The x coordinate describes the horizontal position. When both coordinates are given together, the y coordinate is always specified first.

If you draw a horizontal and a vertical line through the form origin to create x and y coordinates, you would express the location of the form origin as $y=0, x=0$.

If you draw more lines to mark horizontal and vertical measurements away from the form origin so that the lines were one unit of measurement apart, you create a grid like the one shown in figure 2-8. You can then position form elements (lines, boxes, and so on) by specifying a grid location.

Figure 2-8. Form grid



Example

You placed the letter "A" at location (2,1). If you are measuring grid units in inches, the origin of the "A" character cell is located two inches down and one inch to the right of the form origin. If you are measuring the grid in cpi and lpi units, the origin of the "A" character cell is located two lines down and one character width to the right of the form origin. At 10 cpi and 6 lpi, the "A" is 0.33 inch down from the form origin and 0.10 inch to the right.

Negative coordinates

Negative x and y values also may be used to place form elements above and to the left of the form origin, as long as the values are within system page boundaries.

UNIT value

The unit value of the GRID command specifies the unit of measurement FDL commands use to position form elements on the page. Optionally, you can select a predefined format that automatically provides grid unit values. Individual FDL commands may override GRID unit values.

Predefined formats

A set of predefined print description entries or formats, having standard format specifications, is provided on the operating system software (OSS) tapes. Table 2-1 lists the standard LPS print formats. You may use these standard formats or define your own to suit your specific needs. These standard formats provide commonly used impact printer conversion formats for use with specific page sizes and orientations. (Refer to table 2-1.) Use the Xerox Design Ruler to assist you in measuring character and line spacing. The ruler has eight scales that conform to the standard format grid. (Refer to table F-1.)

FMT1 (landscape) or FMT6 (portrait) is used by FDL to provide default values for page orientation, form origin, and grid unit dimensions, provided there are no explicit overriding parameters.

Unless overridden by an explicit orientation PAGE SIZE parameter, the virtual page size in a predefined format is always the sysgen-specified paper size. The virtual page origin is located at the upper left corner of the physical page.

Table 2-1. Standard LPS print formats

Format ID	No. of lines	No. of columns	lpi	cpi	Approx. point size	Page size and orientation*	BEGIN values (y,x)	Default font ID
FMT1	66	132	8.1	13.6	9	11 by 8.5 in.	(.18,.66)	L0112B
FMT2	66	150	8.1	15	9	11 by 8.5 in.	(.18,.50)	L0212A
FMT3	88	132	10.7	13.6	7	11 by 8.5 in.	(.14,.66)	L0312A
FMT4	88	150	10.7	15	7	11 by 8.5 in.	(.14,.50)	L0412A
FMT5	49	100	6	10	12	11 by 8.5 in.	(.17,.50)	L0512A
FMT6	80	100	8.1	13.6	9	8.5 by 11 in.	(.57,.58)	P0612A
FMT7	60	90	6	12	12	8.5 by 11 in.	(.50,.50)	P07TYA
FMT8	60	75	6	10	12	8.5 by 11 in.	(.50,.50)	P0812A
FMT9	80	200	10.0	20.0	7	11 by 8.5 in.	(.15,.25)	L0912A
FMT10	132	132	12.5	17.6	6	8.5 by 11 in.	(.22,.51)	P1012A
FMT11	132	150	12.5	20	6	8.5 by 11 in.	(.22,.50)	P1112A
FMT12	66	172	8.1	13.6	9	14 by 8.5 in.	(.18,.66)	L0112B
FMT13	104	100	8.1	13.6	9	8.5 by 14 in.	(.57,.58)	P0612A
A4 formats								
FMT1A	66	132	8.3	12.5	9	297 by 210 mm	(.18,.57)	R112BL
FMT2A	66	150	8.3	14.3	9	297 by 210 mm	(.18,.60)	R212BL
FMT3A	88	132	11.1	12.5	7	297 by 210 mm	(.18,.57)	R312BL
FMT4A	88	150	11.1	14.3	7	297 by 210 mm	(.18,.60)	R412BL
FMT5A	48	100	6	10	12	297 by 210 mm	(.22,.85)	R512BL
FMT6A	80	100	8.1	13.6	9	210 by 297 mm	(.91,.46)	R612BL
FMT7A	60	90	6	12	12	210 by 297 mm	(.85,.39)	R7TIBP
FMT8A	60	75	6	10	12	210 by 297 mm	(.85,.39)	R812BP
FMT9A	80	200	10.0	20.0	7	297 by 210 mm	(.14,.85)	R912BL
FMT10A	132	132	12.5	17.6	6	210 by 297 mm	(.57,.39)	RA12BL
FMT11A	132	150	12.5	20.0	6	210 by 297 mm	(.57,.39)	RB12BL

* First dimension given is the horizontal dimension. Therefore, 11 by 8.5 inches is landscape orientation, while 8.5 by 11 inches is portrait orientation. For A4 paper, 297 by 210 mm is landscape, and 210 by 297 mm is portrait orientation.

Data types

Two types of data are used in LPS forms creating and printing:

- Variable data
- Forms data.

The term "variable data" generally refers to computer produced information that the LPS merges with a form. Typically this data varies from page to page.

The term "forms data" refers to information that is used to print the form, such as lines between columns and rows, boxes, and shading. Forms data also refers to information that is part of the form such as titles, headings, captions, logos, and signatures. Forms data typically does not vary page to page.

This reference is intended for forms designers who have basic knowledge of the 4850 or the 4890 HighLight Color Laser Printing System (LPS). You may have received on-site training, or you may have attended a Xerox forms description language (FDL) training course.

Forms designer's tasks consists of writing FDL commands to produce electronic forms. These tasks include entering source code into the LPS to create new electronic forms, as well as updating, revising, and printing the forms on the laser printing system. If you are also responsible for printing variable data on the form in addition to performing other operations on the printer, refer to the related documents provided with your laser printing system.

Document conventions

This guide uses the following conventions:

- < > Angle brackets are used for keys on the system controller keyboard.
- { } Curly brackets are used for required characters.
- ... Ellipses indicate that you can repeat a parameter or list a series of parameters.
- [] Square brackets are used for optional command characters.
- | Vertical bars are used to separate parameters in a series. The vertical bar stands for "or."
- bold** Bold is used for characters you enter at the command line.
- italics* Italics is used for variable information.
- `TERMINAL FONT` Terminal or monospace font is used to display system responses.
- underline System default parameters are underlined.
- UPPERCASE Uppercase letters are used for command names.

CAUTION: Cautions alert you to an action that could damage hardware or software.

WARNING: Warnings alert you to conditions that may affect the safety of people.

4. Compiling and printing forms

Once the file of form source library (FSL) has been created, the file can be stored on the laser printing system (LPS) disk. The next step is to compile the file and print a copy of the form.

Form printing process

To print an electronic form on a Xerox laser printing system:

1. Enter the editor utility and create a file of forms description language (FDL) commands to describe the form. This file is called a forms source library (FSL) file. Refer to the "Forms description language" chapter of this reference.

You create an .FSL file containing FDL commands such as coordinates for placing lines, boxes, text, and so forth. You can create the .FSL file directly at the LPS keyboard and display, at a host computer input device, or through remote communications.

2. Save the FSL file to the LPS system disk.
3. Compile the .FSL to create a printable electronic form file (.FRM).
4. Debug, modify, and recompile the file if necessary.
5. Print the form.

FSL data transfer

Once you have created the .FSL file, you can send it to the system disk either offline or online.

Offline

In the offline mode, .FSL files can be transmitted to the system disk from magnetic tape or by using the LPS Editor.

1. Create the source file at the host computer and store it on a magnetic tape.
2. Mount the tape on the LPS tape drive and copy the files electronically to the system disk.

The source code can also be entered directly to the LPS at the system user interface and transmitted electronically to the system disk.

Online

If your LPS is channel-attached to a host computer, enter the source code at the host terminal through the host's editor

facility. The file must be preceded by the necessary control record identifiers. Refer to your host reference documentation for details. The LPS Hostcopy utility enables you to download files from the host to be stored on the LPS system disk.

The LPS can also receive forms data from a remote device on the Ethernet network.

Compiling a form

Once the file has been created, the next step is to enter the command that compiles the .FSL to an .FRM.

Multiple FSLs contained in one file are paginated so that each new FSL starts on the front side of a new page.

Involving the forms compiler

To invoke the forms compiler, enter the following command at the LPS keyboard:

FDL [parameters]

The FDL compilation command includes any combination of the following options. The entire command must fit on one line.

FDL [filename][,NOPRINT][,TRAY][,DEBUG][,XPAN][,NOSOURCE][,PROOF][,SIMPLEX]

filename

Specifies the name of the FSL to be compiled. The file's extension, .FSL, may be included. No other extension is allowed. The file name may not be a keyword option such as TRAY, DEBUG, XPAN, and so on. These are reserved names that could cause problems when you use a task outside the forms compiler.

NOPRINT

Specifies that no listing of the FSL source statements, no form summary, and no sample form are to be printed. However, if an error occurs during the processing of the form, the listing and the summary are printed.

TRAY

Specifies that the sample form, source statement, and summary sheet are to go to the sample tray instead of the currently active output stacker.

DEBUG

Specifies that no attempt to print the sample form is to be made. This ensures that the source statement and summary sheet are printed even if the sample form cannot be printed.

XPAN

Specifies printing of the source statement of section expansions.

NOSOURCE

Specifies that disk source file creation will be suppressed when compiling from magnetic tape or disk, when the form name is the same as the filename given.

PROOF

Specifies that the .FSL is to be compiled and the source statement and summary sheet are to be printed. No FRM file is generated.

SIMPLEX

Specifies that the source statement or summary sheet generated with each compile is printed in simplex rather than duplex.

If the source file contains multiple form descriptions, the system does not start printing until all forms have been compiled.

When an .FSL is compiled and the name of the source input file differs from the name in the FOR command, an .FSL is created using the name from the FORM command. This newly created .FSL contains only those commands necessary for that form. However, if the input file has the same name as the one in the FORM command, the original .FSL file is unaffected.

If there are multiple forms in the input source and the source input file name is used for any of the forms other than the first, an error results during compilation. Therefore, if you wish to use the source input file name as one of the form names, you must use it as the first form to be compiled. Typically, forms contained in a multiple-file .FSL do not have the same name as the source input file. An example of this is DC20.FSL.

If an FSL with multiple forms has one form name which has the same name as another FSL, this second FSL will be overwritten when the first is compiled. As an example, assume that file A.FSL contains forms X, Y, and Z. When it is compiled, not only are files X.FRM, Y.FRM, and Z.FRM produced, but X.FSL, Y.FSL, and Z.FSL as well. Hence, if there already is a file name X.FSL on the system, it is overwritten by X.FSL from the A.FSL file.

Summary sheet

If the compilation is successful, a summary sheet of statistics is printed, containing the following:

- The size of the source output file
- The size of the printable form file
- The size of the form image buffer
- The total amount of font memory used.

The summary sheet is by default printed duplex. If you want it printed simplex, include the "SIMPLEX" option in the task invocation line as follows:

FDL TRYIT,SIMPLEX

Error checks

If the compilation fails, a list of error messages is printed. If a command extends over several lines, an error message may not appear on the next consecutive line following the error. This may result in two consecutive error messages at the end of a command. An example of this is the CANNOT FIND THE BOX message that may occur for a TEXT IN BOX command.

As the .FSL file is processed, it is automatically checked for the following errors.

- Unknown commands, keywords, or parameters
- Illegal or conflicting commands, keywords, or parameters
- Missing commands, keywords, or parameters
- Requests for items to be placed or drawn outside of box or page boundaries
- Invalid font or logo requests
- Illegal grid units
- Illegal syntax.

Errors are flagged in the source statement with messages specifying the type of error. Refer to the "FDL-generated messages" appendix for a description of error messages that may appear while forms are being compiled.

Compilation options

File modification

Whenever you first create a file, it rarely produces the form exactly as you imagined it to look.

- You may have syntax errors that prevent the form from printing.
- You may have successfully followed all the FDL rules and produced a form, but inadvertently placed some text or graphic on the form in the wrong place or misspelled a word.
- Everything comes out the way you planned, but the printed form just does not look right; that is, it would look better if that box were a little larger.

It is easy to identify the specific causes of the problems (debugging) and correct it (modifying).

File storage

A large number of forms may be stored in the forms source directory (uncompiled) and form directory (compiled) on the system disks. Two files with the same name may not, however, be stored in the same directory. The number of forms that can be stored depends on the amount of data required to describe each form and the total amount of disk space allocated to the libraries.

The forms are filed in the system under their identifying names. Information stored in the file includes the identifier, the encoded form, and the list of fonts and logos specified. The original source statements that created the forms description are stored separately under the same identifier.

A list of all forms stored in the directory, a sample of each, and the source statement lists can all be obtained by operator commands.

PROOF

The forms compiler makes it possible to compile a form and delete the .FRM file after it has been printed. If you use this option, less disk space is consumed by resident form files. To invoke the PROOF option, enter:

FDL TRYIT,PROOF

If the form is compiled with no errors, the .FRM file and the source statement summary sheet, or both are printed. Since the PROOF option causes it to be deleted, no form file then resides on the disk.

When you use the PROOF option, the form file that is created is given a name that differs from the one specified in the FSL. When TRYIT.FSL, for example, is compiled, the form that is created is not automatically named TRYIT.FRM. The newly created .FRM file is given a different name, and it is this renamed form that is eventually deleted. Due to this safeguard, if a file named TRYIT.FRM already exists on the system disk, that form still exists after the new .FRM file is compiled and deleted.

Secured files

- If you attempt to overwrite a secured file while compiling a form, a security violation message appears and the compilation process is aborted.
- If you attempt to overwrite a secured file having the same name both on the task invocation line and in the .FSL itself, a security violation message is displayed and the form is aborted.

Example: TRYIT1.FSL is a secured file:

```
You enter:          FDL TRYIT1,TRAY
.FSL contains:      FORM TRYIT1;
                   GRID IS 1 INCH;
                   END;END;
```

In this case, a security violation message would appear and compilation would abort.

- If the form names on the invocation line and in the .FSL are different, the compilation is aborted and a single sheet is printed. It contains the heading for the page and the portion of the .FSL that was processed before the security violation was detected.

Example: TRYIT1.FSL is a secured file:

```
You enter:          FDL TRYIT,TRAY
.FSL contains:      FORM TRYIT1;
                   GRID IS 1 INCH;
                   END;END;
```

In this case, a security violation message would still appear and the compilation would abort, but a single page would be printed containing the page heading and the .FSL line, FORM TRYIT1;.

Example 1 FDL TRYIT,TRAY

This command causes the system to compile the FSL named TRYIT (provided it does not contain errors), generate the source statement, print a copy of the statement and the form, and deliver both to the SAMPLE tray.

If you do not include the word TRAY in your command, the printout is delivered to the currently active bin.

Example 2 If you want to compile the form but you do not want a printout of the form just yet, enter:**FDL TRYIT,NOPRINT**

The FSL compiles, but the form and the source statement does not print.

FDL compilation time

The time it takes to compile a form file depends on the position of the font file in its directory and the size and number of disks on the system. If your LPS has a large number of font files dispersed over several disks, you may notice an increase in compilation time for forms that perform text handling. To ensure that your form files compile as quickly as possible, remove unnecessary font files and use the COMPRESS utility to reorganize your files on the disk. These recommendations will shorten the font lookup time and make the most efficient use of the disk space.

Printing a compiled form

Once a form has been compiled, the object code is stored on the system as a FRM file. To print copies of the form, enter the SAMPLE command at the LPS keyboard. The format of the SAMPLE command is:

SAMPLE *filename*.FRM

Example: SAMPLE TRYIT.FRM

To produce five copies of the form, enter the following:

SAM[PLE] TRYIT.FRM,5

The *Xerox 4850/4890 HighLight Color LPS Command Reference* describes additional options for using the SAMPLE command.

5. Forms layout considerations

Physical printer constraints and forms compiler methods are factors that can affect form design. This chapter explores the implications of these factors relative to the following:

- Image complexity
- Boxes
- Shading
- Sections
- Rounding measurements
- Coding techniques.

Image complexity

Form design is limited by physical restrictions on the amount of image data (characters and lines) that can be printed on a scan line and on a page.

Line tables

Forms description language (FDL) line tables are internal FDL data structures providing a record in memory of the lines to be drawn on a page. These tables allow the system to find the corners and edges of boxes when they are referenced at a later time.

Since excessive amounts of memory would be required to store every line, the FDL compiler retains only the memory of the longest lines that pass through any given coordinate in a particular direction. FDL replaces dotted or broken lines with an invisible line of the same length.

The FDL line tables can store 2,000 vertical and 2,000 horizontal lines in memory. When you exceed either of these line table limits, the following message is displayed:

```
TOO MANY HORIZONTAL OR TOO MANY VERTICAL LINES
```

This message warns you that the new user-specified line cannot be retained in the table. The message is of no real consequence to you unless the new line is intended as the side of a box. In that case, the box is not found and consequently your form may not compile.

Generally, the line table limits are reached because the form contains a large number of short line segments. The most probable reason for this is a series of vertical lines (for example, tick marks) or horizontal (for example, a series of long dashes). With either of these conditions, it is possible to reduce the number of entries in the line table as the form is being constructed.

Importance of orderly construction

One consideration in determining the orderliness of form construction is that the algorithm used in loading the line table is designed to check only two criteria before entering a new line in the line table. These criteria are:

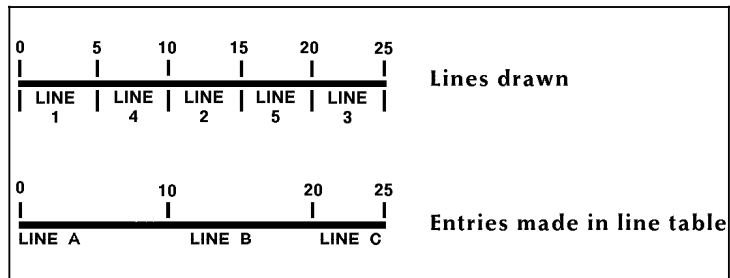
- Is the new line contained within the boundaries of an existing line?
- Is the new line an extension of *one* of the existing lines?

If the new line is contained within the boundaries of an existing line, no change is made to the line table entries. If the new line is an extension of one of the existing lines, the boundaries of the first (and only the first) existing line is extended to include the new line. If neither of the criteria is met, a new entry is made in the line table.

Examples

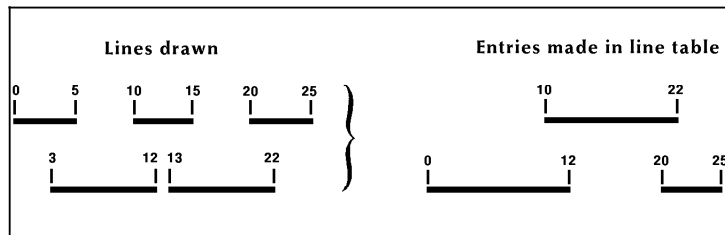
The procedure of line extension results in an irregularity if lines are not specified in an orderly manner. For example, draw five line segments that are each five units in length. If the lines are drawn in an unreasonable or irregular manner, as shown below, the result is more entries in the line table than are necessary. Drawing line segments in the order 0 to 5, 10 to 15, 20 to 25, 5 to 10, and 15 to 20 creates a single visual line that FDL considers to be three lines. This is because the criterion of the line table allows extension of only the first line segment with common coordinates to the new line. In the case described, the three line entries in the line table would be 0 to 10, 10 to 20, and 20 to 25. Figure 5-1 shows examples of extraneous lines drawn.

Figure 5-1. Example of extraneous lines drawn



In addition, it is possible to have the line table show more than one line passing in the same direction through a given coordinate. For example, draw three lines, each of which is five units in length (0 to 5, 10 to 15, and 20 to 25). Then draw two lines, each of which is nine units in length (3 to 12 and 13 to 22). The result is a single visual line for which FDL has three entries in the line table (0 to 12, 10 to 22, and 20 to 25). Figure 5-2 shows lines entered in unintended locations.

Figure 5-2. Lines entered in unintended locations



These two examples of awkwardly entering lines into the line table result in lines that are visually contiguous but not logically contiguous. As indicated in the subsequent sections that describe problems with boxes, such conditions can result in a situation where FDL is unable to find a box or creates a box that has one or both of its dimensions equal to zero.

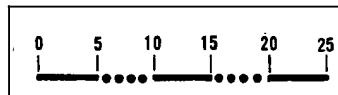
Suggested techniques for entering lines

To enter lines, draw the longest possible logical line first, then draw any shorter elements. If the logical line consists of different types of lines (for example, solid and dotted), draw a SOLID 0 line first extending over the full range of the logical line. For instance, the previously cited example of five line segments of five units each might have consisted of alternating solid and dotted lines. In that case, the most reasonable method of specifying the total line would be as follows:

```
AT 0 LINE 0 TO 25 USING SOLID 0;
AT 0 LINE 0 TO 5 USING SOLID 2 AND AT 10, 20;
AT 0 LINE 5 TO 10 USING DOTTED 1 AND AT 15;
```

Figure 5-3 shows the line that would be drawn:

Figure 5-3. Line made up of different types of lines



Scan line density

Scan line density depends upon printer and imaging speed. Therefore, it can vary with each product and even with different configurations of the same product. The basic limitation is linked to dispatchable items, which become visible in character count and number of lines.

Line density errors

Both characters and lines contribute to the dispatchable item count. When one dispatchable item is superimposed over another dispatchable item, the result is considered two dispatchable items by the system. When too many dispatchable items are present on a scan line on the long axis of the paper, the LPS operating system displays a line density error message:

```
OS6950 LINE DENSITY EXCEEDED
```

The hardware limitation of 320 dispatchable items per 300 spots per inch (spi) scan line includes the required overhead of two dispatchable items. For example, a dispatchable item could be a maximum of 32 dots wide for 300 spi and thus might be a full character (less than point size 8 or 9), a fraction of a character (greater than point size 8 or 9), or a line segment that is equal to or smaller than 32 dots in the scan direction.

Determining line density limitations

Determine 300-spi line density limitations as follows:

- Each 32 dots (or fraction thereof) of a line drawn on the long axis of the paper is one dispatchable item.
- Each line drawn on the short axis of the paper is one dispatchable item.

- Each active character is at least one dispatchable item. (An active character includes all the space from the top of the character cell to the bottom of the character cell, even if there is only white space present on a given scan line.) If the dimension along the scan line is greater than 32 dots, the character consists of some multiple dispatchable items.

It is not possible to construct a form with two adjacent lines if both use the full count of 318 (320 minus the two required for overhead) dispatchable items resulting from text and lines. The imaging system requires a rest period of about four scan lines before it can image the second fully loaded line.

Landscape pages

Line density restrictions differ as a function of the mode (landscape or portrait) of the form. An 8-point or smaller landscape font is smaller than 32 dots for 300 spi. A vertical line is a single dispatchable item. A horizontal line is treated as a series of 32 dots in length, joined end-to-end.

For 300 spi, this means that a form using an 8-point or smaller font on a landscape page may have up to 318 characters (plus a two-item overhead, resulting in 320 dispatchable items) on a scan line. When using a 6-point font, it is possible to have, for example, 132 characters, a line across the page under the text, and up to 90 vertical lines, without exceeding the line density limitations.

If a form requires a 9-point or larger font, the number of characters that may be imaged on a landscape page is reduced to 160 because the characters are generally more than 32 dots wide. While the dispatchable item count remains 320 per line, fonts 9 points and larger use two dispatchable items per character. Since a character in a large font constitutes two dispatchable items, only half as many characters in a large font may be imaged on a scan line as is possible with a smaller font.

Portrait pages

Restrictions for a portrait page are significantly different from those for a landscape page because the system always images in a landscape direction. Therefore, if a page is formatted in portrait orientation, the hardware must reorganize the data into landscape prior to imaging.

On a 300-spi portrait page, a 12-point font drops the dispatchable-item count below the 320-character level. A form with a 12-point font may have not more than 160 characters on a scan line.

When using an 8-point 300-spi font on a portrait page, the maximum dispatchable-item count permitted is approximately 200 per scan line. This limit falls off gradually to 170 characters for a 6-point font, 150 characters for a 5-point font, and 120 characters for a 4-point font.

Superimposition

Since lines superimposed over each other count as double the dispatchable items of a single line, they detract from the system capabilities.

Page generation errors

Generally, a form is imaged along with variable data. It is possible to create a form that prints correctly when sampled but cannot be successfully overlaid on certain variable data pages. The LPS may not have enough time to merge a complex form with a large amount of variable data within the limits of the throughput environment. In this case, either the amount of variable data can be reduced, or the form can be simplified. Review the broken page to determine the approximate location of the failure, and try to reduce the number of characters and lines in that area. Most imaging problems are local density of information problems rather than page-wide problems.

Local density and page setup errors

One problem that may occur while a page is being imaged is a page setup error, displaying the following message:

```
OS9300 PAGE SET-UP ERROR
```

This message means that there was not enough time to image a page. The lack of time could be caused by an excessively large amount of data and forms to be imaged, disk errors, image generation problems, or a problem known as "local density."

Even a form designed to avoid exceeding line density limitations may cause page setup errors because of local density. However, such problems are rarely encountered in readable forms. To have a problem area of local density, a form would have to have long lines of small text very closely spaced.

The reason for a local density problem is that the imaging hardware accepts and dispatches characters at two different rates. If a form structure imposes a dispatching load that is too heavy, the hardware may run out of input before the processing of data is finished.

Troubleshooting

If a form you described causes page setup errors, examine the form and variable data to see if there are many lines or a large amount of text in a small area. If so, you may reduce the density of this area by spreading out the material or by deleting parts of the form.

The general guide for determining the possible cause of a page setup error related to local density is as follows:

- Large amounts of closely spaced text in a small font; that is, long lines of solid text with minimum spacing, especially if a form is portrait
- Large amounts of text in an orientation different from the defined form orientation; for example, portrait text in a landscape form
- A large number of lines drawn on the short axis of the paper; that is, vertical lines on a landscape form
- Large amounts of shading.

Summary

Optimize form generation in the following manner:

- Use lines and boxes rather than characters of the form element font, whenever possible.
- Avoid placing too many small characters in one area of the page.
- Avoid overlapping (superimposing) lines or boxes.
- Avoid overlapping characters to achieve bolding or other effects. Use the correct font whenever possible.
- Avoid using too many dotted or broken lines because they create too much overhead in the printing process. You receive a displayed message during compilation each time a broken or dotted line has been defined.
- Avoid too many font switches on the same text line. This creates overhead in the printing process.
- Avoid drawing unneeded lines across the longest side of the physical page.

FDL statistics

If a form is successfully compiled, the following FDL statistics are displayed on the summary sheet.

- Size of the source output file (FSL)
- Size of the printable form file (FRM)
- Size of the form image buffer
- Amount of font memory used by the form.

Using boxes

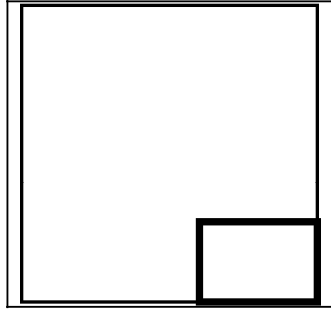
You may draw boxes with a single BOX command, or construct them with commands on a line-by-line basis.

A line density problem can occur if a single BOX command is used to specify a box that is adjoined to another box by a single side. When two BOX commands are issued in this manner, one line of a box is superimposed over another along the adjoined side. Since two lines are drawn between the same pair of coordinates, a line density problem may occur.

When to use the BOX command

To avoid this situation, use the BOX command only for the following conditions when designing a form:

- The box is fully isolated from any other lines
- The line type of a box is different from any other lines that share its boundaries (for example a SOLID 2 box sharing a small portion of a SOLID 1 edge, as illustrated in Figure 5-4).

Figure 5-4. **SOLID 2 box sharing part of a SOLID 1 box edge**

BOX FILL option

A box filled with a light shade may appear to be smaller than a same size box filled with a darker shade. The barely noticeable difference is usually apparent only when boxes with different light shading lie immediately adjacent to each other, as in some bar graphs or histograms. If this is considered a problem, it may be corrected as follows:

- If the top of a box is affected, move the box up or down by one dot (1/300 inch).
- If the bottom is affected, reduce or extend the height of the box by one dot (1/300 inch).
- If the left side of the box is affected, move the box left or right by one dot (1/300 inch).
- If the right side is affected, reduce or extend the width of the box by one dot (1/300 inch).

Text in boxes

When text is entered for placement within a box, the system attempts to determine whether the given text exceeds the largest font size that can fit within the boundaries of the box. If FDL computations indicate that the entered text may not fit within the box, the following error message is displayed:

TEXT WILL NOT FIT IN THE BOX

The system checks both the horizontal and vertical directions of the box. If an error is detected in either or both directions, an attempt is made to indicate an approximate value for the largest font.

Locating the closest box

When you specify text to be placed in a box, the system locates the closest box within a ten-dot square of the specified point.

Example If text is to be placed in a box located two grid units by two grid units from the origin of the form, the system selects the box closest to this point and within ten dots square. The upper left corner of this box becomes the relative origin for subsequent text placement.

- If text is to be placed in the next box horizontally, FDL begins searching at the upper right corner of the box in which previous text was placed.
- If text is to be placed in the next box vertically, the search begins at the lower left corner.

Error calculations

Horizontal errors

The calculations made for horizontal errors (for example, too many characters in a line) are usually accurate. FDL uses the following formula for reporting horizontal errors:

$$S = P * (B/T)$$

- S target point size
- P original font point size
- B width of the box in dots
- T width of the text in dots

Vertical errors

The calculation made by FDL for vertical errors (for example, too many lines in the box) is a rough approximation of the actual size of text, and the computation may, in fact, be larger than the font for which an error message was given. This is because the FDL system of reckoning does not take into account the leading used within the font or provided by line spacing specifications.

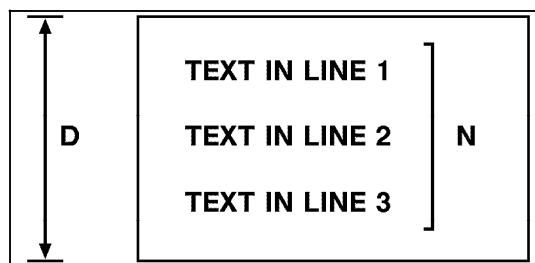
The formula used by the error-handling routine to compute text size when the box is too small in a vertical direction is:

$$S = (72/300) * (D/N)$$

- S target point size
- 72 number of points in an inch
- 300 number of dots in an inch
- D distance between the top and bottom of the box (in dots)
- N number of lines that are involved

Figure 5-5 shows measurements used in computing vertical box errors.

Figure 5-5. **Measurements used in computing vertical box errors**



The result of this equation is rounded up and displayed in the error message. Thus, what the error message actually signifies is that the largest font could be accommodated within the box if no leading were involved.

Correcting TEXT-IN-BOX errors

If an error is detected, you have the following choices:

- Make the box larger.
- Reduce the number of lines.
- Use a smaller font.
- If text visually fits, use a TEXT AT command rather than a TEXT IN BOX command.

Centering text in boxes

FDL uses line spacing as the vertical unit of size in centering text within a box. In a small box, text that may not appear to be properly centered may actually be centered accurately if line spacing is considered. FDL cannot evaluate the aesthetics of the final printed product. Therefore, it is up to you to create an appropriate appearance for centered text.

Specifying line spacing within boxes

Most often, it is easiest to allow FDL to position text within boxes. In cases where the result is not satisfactory, you can do the following:

- Specify an overriding line spacing value.
- Exercise more control over text placement and assure satisfying aesthetic requirements with the SPACED AT option, possibly used in conjunction with blank lines, or the TEXT AT command (that is specifically positioning text).
- If overriding the line spacing does not achieve a satisfactory appearance in particular form, specify a line spacing that is smaller than the natural spacing of the font. A line spacing smaller than the height of the characters could be used in conjunction with blank lines.

Example

Text might be spaced at four dots, while the line spacing is really 59 dots. It seems paradoxical to designate that text be spaced at four dots and create a form that actually has 59 dots of spacing. However, this can be achieved through the use of blank lines. A small line spacing along with blank lines allows you to get positioning control and forces visual centering of text within a box.

The statements below provide an example of how this technique is used:

 represents a blank line.

TEXT SPACED AT 4 DOTS IN BOX 10, 10



'TEXT LINE 1'



'TEXT LINE 2'



As many blank lines may be used before, between, and after lines of text as are needed to obtain the desired visual effect.

Default color

All objects in a color form have a color association, whether explicitly specified or not. The default, or nonspecified, color for an object in FDL is black.

Shading

SHADING is an option that is available for compatibility with non-color laser printing systems. When an ink is specified in your form, use the FILL option of the BOX command instead. When an ink is not specified, use the SHADING option.

It is sometimes desirable to use shading in a form, but it requires using a large amount of the form image buffer. Also, the space used increases if portrait rather than landscape shading is used.

Landscape shading

The basic element used in shading is a 32-by-32-dot character for dots. This character can be repeated in a landscape direction up to 32 times for dots. This means that in a landscape direction, the system can shade any area from 0.1 to 3.4 inches wide with no difference in space used by the shading.

This facility is not available when shading is done in a portrait direction. Portrait shading is performed on a character-by-character basis.

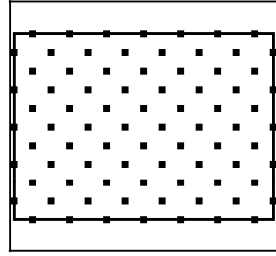
Portrait shading

To reduce shading and improve the visual effect of the form, end shading before the boundary line. Separate the shading from the surrounding lines by 0.1 to 0.2 line/character units. Thus, the shading is 0.2 to 0.4 units smaller than the surrounding box and is centered in that box. (Units referred to are lpi and cpi units.) This sets shading back from the borders by 3 to 5 dots.

Avoiding boundary line density problems

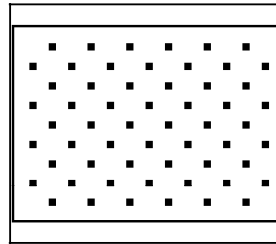
Terminating the shading before reaching the boundary line helps to avoid line density problems, conserves space in the form file and in the form image buffer, and enhances the appearance of the form.

Figure 5-6. **Normal shading—using the same coordinates and sizes as the box**



The early termination of the shading is shown in Figure 5-7:

Figure 5-7. **Early termination—using a smaller box for shading**



Sections

To duplicate identical information on a page, create sections. However, form image buffer limitations and potential line density problems should be taken into account whenever this feature is used.

A form comprised of many sections often takes up more space in the form image buffer than a form created without sections. This occurs because a single long line requires less buffer space than numerous short lines. When you create a form by invoking defined sections, there is a tendency to begin and end lines at section boundaries, even when they are continuous through a series of sections. This is a much less efficient form creation technique than taking the long lines out of the sections and drawing them as continuous lines.

A less obvious inefficiency of using the form image buffer may occur when there is a line that switches from solid to dotted and vice versa. The common method of creating this type of line is to draw short segments of solid and dotted lines. (This is inefficient use of the form image buffer even if a SOLID 0 line is drawn first. A SOLID 0 line is useful in saving space in the line tables, but cannot help save buffer space.)

To save form image buffer space efficiently, draw a continuous dotted line and overlay a series of short, solid line segments. This technique makes best use of the form image buffer space, but does not reduce the load on the imaging section of the system. (Refer to the "Scan line density" section in this chapter.) In a form where there is a problem with exceeding both form image buffer and line density restrictions, you may have to use logos or a specialty form font.

Rounding measurements

Laser printing systems use only dot units in the actual performance of forms compilation. Therefore, as long as all positioning commands within FDL are defined in terms of dots or an integral number of dots, there are no rounding problems.

Converting other unit values to dots

You may define positions in decimal-valued inches, in centimeters, in lines per inch, and in characters per inch. The LPS converts all of these commands to dots while the form is being compiled.

Many of the possible values defined (lines per inch, centimeters, and so forth) cannot be reduced to an integral number of dots. For example, if a coordinate system uses 13.6 characters per inch (cpi), that number may be translated to approximately 22.058823 dots per character unit. The compiler rounds the converted number to the nearest whole dot before starting to compile the form.

Example

The example statements below illustrate the problems encountered using the 13.6 cpi coordinate system.

```
AT 4 DRAW 1 HOR LINE FROM -1 TO 132 USING SOLID 1;
```

```
AT 5 DRAW 1 HOR LINE FROM -1 TO 132 USING SOLID  
HAIRLINE;
```

```
AT 8 DRAW 15 VER LINES FROM 4 TO 5 USING HAIRLINE AND  
REPEAT HOR EVERY 8;
```

In theory, this form description technique creates 14 boxes that can be found simply by addressing the appropriate coordinates. In fact, the coordinates of the boxes are not exactly those expected. The boxes can be found by using the IN NEXT BOX parameter of the TEXT IN BOX command.

The IN NEXT BOX parameter works properly because it specifies the next adjacent box but does not specify the coordinates of that box. The problem in addressing coordinates is that the FDL statement requires vertical lines every 176.47058 dots, which is rounded down by the system to 176.00000 dots.

This difference makes it impossible for the LPS to find box 14 at the coordinates 4, 112. There are two ways to draw the form to overcome this problem. One way is to draw 13 lines with the REPEAT parameter and then specify lines at 112 and 120. Another way is to change the final statement in this manner:

```
AT 8 DRAW VER LINE FROM 4 TO 5 USING HAIRLINE AND AT  
16, 24, 32, 42, 48, 56, 72, 80, 88, 96, 104, 112, 120;
```

Unless a form is described with a coordinate system of only dots or an integral number of dots, the REPEAT parameter cannot be used to draw boxes that must be referenced by their coordinates.

Rounding variable data

Alignment problems

Under certain circumstances, a form and variable data do not align properly, even though it appears that both use the same line spacing. When this happens, it may be the result of the effects of rounding on the line spacing.

As an example, we will construct a form with a grid unit of 13.6 cpi and 9 lpi. Horizontal lines are specified at 0, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, and 67. The data is printed with an override line spacing of "9."

The natural assumption is that the form and the variable data line up. However, when the variable data is merged with the form, it tends to drift up the page.

The reason for this is rounding. Specifying 9 lines per inch for the variable data means that the LPS is requested to place data at approximately 33.333333 dots. The system rounds this number off to 33 dots per line.

Meanwhile, in FDL rounding does not take place until it is time to resolve specifications for the form to a dot address.

Please note the difference: 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.

For instance, the line drawn at 43 is assigned to scan line 1433 using the formula below:

$$(43 * 300) / 9 = 12900 / 9 = 1433.3333 \text{ or } 1433$$

The equivalent line positioning for the variable data is computed with this method:

$$43 * (300 / 9) = 43 * 33 = 1419$$

This yields an error that increases as calculation proceeds down the page.

Furthermore, if, in the example form, we had drawn lines with a REPEAT EVERY 3 LINES command, the uneven dot value (33.333333 dots per line) would not have been rounded off because rounding would not have been performed until it was time to resolve the specification to a dot address. The dot address for three lines is an even 100 dots. The lines for the form would be drawn at exactly 100 dots, and the variable data would use a 99-dot spacing.

How to avoid imperfect alignment

The best way to ensure that a form and variable data share the same coordinate system is to define both in terms of an integral number of dots. It is also important to know when rounding will affect the alignment of the variable data and the form. In the example, the only way to make the form match the lines of data is to set the line spacing for both at 33 dots.

If variable data does not line up properly with a form, and it appears that both are using the same line spacing, determine the dot value. It is most likely that the difficulty is a variation in defining the coordinate system resulting from rounding, and is not a software problem.

Suggested coding techniques

Converting preprinted forms

To convert preprinted forms:

1. Print data, using an appropriate format, on the laser printing system.
2. Draw horizontal and vertical lines on system output.
3. Use the forms design ruler to measure line and character spacings. (Refer to the "Support tools and measurements" appendix.)
4. Code the form.

Designing new forms

To design new forms:

1. Determine the appropriate format.
2. Use the forms design ruler to measure line and character spacings. (Refer to the "Forms design ruler" appendix.)
3. Draw horizontal and vertical lines on a layout sheet.

Use standard layout sheet forms FLSF1 through FLSF11 that are provided in the FDL software:

- FLSF1 through FLSF11 correspond to paper size 8.5 by 11 inch/216 by 279 mm.
 - Forms FLSF1A and FLSF11A correspond to A4 paper size.
 - FLSF12 and FLSF13 correspond to 8.5 by 14 inch/216 by 356 mm paper.
4. Define the y,x coordinate.
 5. Code the form.

Recommended coding sequence

The FDL compiler imposes a number of constraints on the sequence in which commands are coded. The form name must come first. Only one form orientation is permitted, and fonts and boxes must be implicitly or explicitly defined before they are referenced by TEXT commands. After that point, sequence is not significant to the compiler. However, it is worthwhile to develop a sequence while debugging or making changes to a form. Use a method of sequencing commands similar to the following order:

- Horizontal and vertical lines
- Boxes (if needed)
- Text
- Logo placement
- Shading.

Table 5-1. **Forms layout tips, tricks, and pitfalls**

Form id	Develop a logical, systematic approach to naming conventions. Take care not to choose a previously used form id.
GRID command	Choose a format that is compatible with the data.
Coding lines	Use a COMMENT command. Code all horizontal lines, then all vertical lines. Precede each set with a comment statement. Code longer lines first. Then follow with the shorter segments that may overlay the basic long lines. Code lighter lines first. Then code shorter segments that are heavier in density. Code a SOLID 0 line first whenever you have a set of colinear segments, whether they are touching or not.
Coding boxes	Use a COMMENT command. Use a BOX command only when the lines generated by that command are not duplicated by another command. Do not use a REPEAT command when drawing lines that create boxes if you need to find the coordinates of those boxes at a later time.
Use of shading	Use a COMMENT command. Avoid overlapping the boundary line of a box with shading. Set shading back from borders by three to five dots (or about 0.1 units for most coordinates). This conserves memory and results in a cleaner appearance. Try to avoid shading on portrait forms.
Coding text	Use a COMMENT command. Remember that different fonts of a given point size are not necessarily the same physical size. Be aware that vertical error messages generated when the system calculates that too much text has been specified for placement in a box are an approximation at best. Use the TEXT SPACED AT command to force exact placement of text.
Coding sections	Use a COMMENT command. Use coordinates based at 0,0 when defining a section. Then place the section at the desired locations.
Source copy	If the source filename (.FSL) and the form name (name that follows the FORM keyword) do not agree, an FSL file is created that has the same name as the form name. Therefore, you copy X.FSL to Y.FSL and compile Y.FSL without changing the form name in Y.FSL, the original X.FSL is deleted and a new X.FSL is created from the Y.FSL source. Furthermore, the form file (.FRM) that is created is X.FRM and not Y.FRM. If you wish to create a distinct source using another FSL as a base, you must change the form name to the same name as the source name or to some other name that does not already exist. Two files with the same name and type cannot reside on the system.

Syntax ambiguities

Because FDL keywords can be abbreviated to the first three characters, and because certain two-character keywords can be interpreted in more than one way, the following ambiguities exist:

- IN (preposition) is interpreted as IN (inch). For example:

TEXT 1 IN BOX...;

Also,

TEXT SPACED 1 IN BOX...;

The keyword IN is interpreted as INCH, while the required keyword IN (preposition) is detected as missing and an error message results. The solution to this situation is to write:

TEXT 1 IN IN BOX...; or

TEXT 1 INCH IN BOX...; or

TEXT 1 DOTS IN BOX...;

- FMTn cannot be abbreviated.
- [DOT]TED is interpreted as [DOT]S. For example:

AT 18 DRAW 5 HORIZONTAL LINES FROM 27.5 TO 37.5
DOT...;

The keyword DOT is interpreted as DOTS, which is valid in this position. The keyword DOTTED is also valid in this position but must be spelled out in full.

- [CEN]TER is interpreted as [CEN]TIMETERS. For example:

TEXT 10 CEN IN BOX.....;

The keyword CEN is interpreted as CENTIMETERS, which is valid in this position. The keyword CENTER is also valid in this position but must be spelled out in full.

Writing the setup commands

The setup commands precede the form description commands for lines, boxes, and text. In general, the setup commands specify the parameters that describe the form.

Invoke the Editor utility and enter the commands shown in figure 6-2.

Figure 6-2. **Setup commands**

```
FORM 1STFRM;  
GRID FMT1;  
FONTS UN114A,UN106A,UN104C;
```

Notice that the PORTRAIT or LANDSCAPE command and the PAPER command are omitted. These commands are optional; when they are omitted, the system uses the default values.

Using the COMMENT command

Comments enable you to include in your source statement, messages or reminders that help you organize your FDL commands logically. These comments are not printed on the form but can be useful if you or someone else modifies the form at a later date.

If the semicolon is omitted, the forms compiler cannot determine where the comment ends and may treat commands as comments.

Enter the COMMENT shown in figure 6-3, which indicates that horizontal lines are coded next.

Figure 6-3. **COMMENT command**

```
FORM 1STFRM;  
GRID FMT1;  
FONTS UN114A,UN106A,UN104C;  
COMMENT *** DRAW HORIZONTAL LINES ***;
```

Drawing lines

LINE commands are used to construct horizontal and vertical lines which, in turn, are used to construct boxes.

Drawing horizontal lines

Enter the horizontal line commands shown in figure 6-4. Remember that you have the option of abbreviating the commands to the first three characters.

Figure 6-4. **HORIZONTAL LINE** command

```
FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
```

Drawing vertical lines

Insert another COMMENT to indicate that vertical lines are to be coded next, then enter the vertical line commands shown in figure 6-5.

Figure 6-5. **VERTICAL LINE** command

```
FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
COMMENT *** DRAW VERTICAL LINES ***;
AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI;
AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10;
AT 29 DRAW VER LINE FROM 10 TO 61 USI BRO HAI;
AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10;
AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
```

Drawing boxes

The BOX command enables you to construct individual, stand-alone boxes or boxes that share common sides. You can also create a shaded area or construct an invisible box to conveniently place "floating" text such as titles.

Enter the following COMMENT and BOX command shown in figure 6-6.

Figure 6-6. **BOX command**

```
FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
COMMENT *** DRAW VERTICAL LINES ***;
AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI;
AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10;
AT 29 DRAW VER LINE FROM 10 TO 61 USI BRO HAI;
AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10;
AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
COMMENT *** DRAW BOXES ***;
AT 2,2 DRAW BOX 35 WIDE BY 2 HIGH USI HAIRLINE;
AT 2,82 DRAW BOX 50 WIDE BY 2 HIGH USI HAI;
AT 5,0 DRAW BOX 132 WIDE BY 56 HIGH USI HAI;
AT 5.1,0 BOX 132 WIDE BY 1.8 HIGH USI SHADING;
AT 10.1,0 DRAW 9 BOXES 132 WIDE BY 2.8 HIGH USI SOL HAI
  INK 'BLUE' AND FILL USING INK 'XEROX.PICTORIAL.PALE BLUE'
  AND REPEAT VER EVE 6;
```

Using color in boxes

The sample form uses blue ink. You can substitute the color of ink currently loaded in your system, or leave the default coding. When you compile the form, enter the SUBSTITUTE INK CURRENT command, to automatically print your form in the ink that is loaded, no matter which ink is specified in the form.

Placing text at a location

The TEXT AT command enables you to place text at an exact location.

Enter the COMMENT and TEXT AT command as shown in figure 6-7.

Figure 6-7. TEXT AT command

```

FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
COMMENT *** DRAW VERTICAL LINES ***;
AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI;
AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10;
AT 29 DRAW VER LINE FROM 10 TO 61 USI BRO HAI;
AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10;
AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
COMMENT *** DRAW BOXES ***;
AT 2,2 DRAW BOX 35 WIDE BY 2 HIGH USI HAIRLINE;
AT 2,82 DRAW BOX 50 WIDE BY 2 HIGH USI HAI;
AT 5,0 DRAW BOX 132 WIDE BY 56 HIGH USI HAI;
AT 5.1,0 BOX 132 WIDE BY 1.8 HIGH USI SHADING;
AT 10.1,0 DRAW 9 BOXES 132 WIDE BY 2.8 HIGH USI SOL HAI
  INK 'BLUE' AND FILL USING INK 'XEROX.PICTORIAL.PALE BLUE'
  AND REPEAT VER EVE 6;
COMMENT *** TEXT AT COMMAND ***;
TEXT USING FONT 2 AT 2,2.5 'COMP.'
TEXT USING FONT 2 AT 2,11 'BR.';
TEXT USING FONT 2 AT 2,14 'DEPT.';
TEXT USING FONT 2 AT 2,22 'DATE';
TEXT USING FONT 2 AT 2,30 'PP';
TEXT USING FONT 2 AT 2,34 'PAGE';
TEXT USING FONT 1 INK 'BLUE' AT 3,47 'EARNINGS REGISTER';
TEXT USING FONT 2 AT 61,0 '7860-143-1';
VER TEXT USING FONT 3 AT 66,-5 'F8662';

```

Placing text in a box

The TEXT IN BOX command allows you to place text at a particular location within a box that was previously described. Enter the COMMENT and TEXT IN BOX command as shown in figure 6-8.

Figure 6-8. TEXT IN BOX command

```

FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
COMMENT *** DRAW VERTICAL LINES ***;
AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI;
AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10;
AT 29 DRAW VER LINE FROM 10 TO 61 USI BRO HAI;
AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10;
AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
COMMENT *** DRAW BOXES ***;
AT 2,2 DRAW BOX 35 WIDE BY 2 HIGH USI HAIRLINE;
AT 2,82 DRAW BOX 50 WIDE BY 2 HIGH USI HAI;
AT 5,0 DRAW BOX 132 WIDE BY 56 HIGH USI HAI;
AT 5.1,0 BOX 132 WIDE BY 1.8 HIGH USI SHADING;
AT 10.1,0 DRAW 9 BOXES 132 WIDE BY 2.8 HIGH USI SOL HAI
  INK 'BLUE' AND FILL USING INK 'XEROX.PICTORIAL.PALE
  AND REPEAT VER EVE 6;
COMMENT *** TEXT AT COMMAND ***;
TEXT USING FONT 2 AT 2,2.5 'COMP.'
TEXT USING FONT 2 AT 2,11 'BR.';
TEXT USING FONT 2 AT 2,14 'DEPT.';
TEXT USING FONT 2 AT 2,22 'DATE';
TEXT USING FONT 2 AT 2,30 'PP';
TEXT USING FONT 2 AT 2,34 'PAGE';
TEXT USING FONT 1 INK 'BLUE' AT 3,47 'EARNINGS REGISTER';
TEXT USING FONT 2 AT 61,0 '7860-143-1';
VER TEXT USING FONT 3 AT 66,-5 'F8662';
COMMENT *** TEXT IN BOX COMMANDS ***;
TEXT USI FONT 2 IN BOX 5.1,24 'THIS PAY' BOX 'REGULAR' BOX 'OVERTIME';
TEXT USI FONT 2 IN BOX 7,0 'NAME'
  BOX 'GROSS' 'ADJ. NET' 'NET PAY'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
TEXT USI FONT 2 IN BOX 8,19 'EMP.##';
TEXT USING FONT 2 IN LEFT CENTER BOX 9,0 'SAL/RATE';
TEXT USING FONT 2 IN BOX 9,0 'GROSS YTD';

```

END command

END is the final command you need to enter, as shown in figure 6-9.

Figure 6-9. END command

```

FORM 1STFRM;
GRID FMT1;
FONTS UN114A,UN106A,UN104C;
COMMENT *** DRAW HORIZONTAL LINES ***;
AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3;
AT 8 DRAW 18 LINES FROM 0 TO 24 USI HAI AND REPEAT EVE 3;
AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3;
COMMENT *** DRAW VERTICAL LINES ***;
AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI;
AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10;
AT 29 DRAW VER LINE FROM 10 TO 61 USI BRO HAI;
AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10;
AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3;
COMMENT *** DRAW BOXES ***;
AT 2,2 DRAW BOX 35 WIDE BY 2 HIGH USI HAIRLINE;
AT 2,82 DRAW BOX 50 WIDE BY 2 HIGH USI HAI;
AT 5,0 DRAW BOX 132 WIDE BY 56 HIGH USI HAI;
AT 5.1,0 BOX 132 WIDE BY 1.8 HIGH USI SHADING;
AT 10.1,0 DRAW 9 BOXES 132 WIDE BY 2.8 HIGH USI SOL HAI
  INK 'BLUE' AND FILL USING INK 'XEROX.PICTORIAL.PALE
  AND REPEAT VER EVE 6;
COMMENT *** TEXT AT COMMAND ***;
TEXT USING FONT 2 AT 2,2.5 'COMP.'
TEXT USING FONT 2 AT 2,11 'BR.';
TEXT USING FONT 2 AT 2,14 'DEPT.';
TEXT USING FONT 2 AT 2,22 'DATE';
TEXT USING FONT 2 AT 2,30 'PP';
TEXT USING FONT 2 AT 2,34 'PAGE';
TEXT USING FONT 1 INK 'BLUE' AT 3,47 'EARNINGS REGISTER';
TEXT USING FONT 2 AT 61,0 '7860-143-1';
VER TEXT USING FONT 3 AT 66,-5 'F8662';
COMMENT *** TEXT IN BOX COMMANDS ***;
TEXT USI FONT 2 IN BOX 5.1,24 'THIS PAY' BOX 'REGULAR' BOX 'OVERTIME';
TEXT USI FONT 2 IN BOX 7,0 'NAME'
  BOX 'GROSS' 'ADJ. NET' 'NET PAY'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
  BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'
TEXT USI FONT 2 IN BOX 8,19 'EMP.##';
TEXT USING FONT 2 IN LEFT CENTER BOX 9,0 'SAL/RATE';
TEXT USING FONT 2 IN BOX 9,0 'GROSS YTD';
END;

```

Compiling the form

This exercise gives you practice using most of the FDL commands discussed throughout this guide.

The next step is to compile the FSL which produces a sample of the form, provided there are no syntax errors. The summary statement also prints whether or not there are errors.

After you have entered the END command, SAVE, CLEAR, and END the editing session.

Figure 6-10. **Sample of SAVE, CLEAR, and END the editing session**

```
END;  
SAVE 1STFRM.FSL  
CE  
FDL 1STFRM,TRA
```

CE is an abbreviation for CLEAR and END.

Do not be surprised if you inadvertently omit a semicolon or misspell a word. The system indicates where the error occurred and you can easily go back into the form and correct the error.

If there are errors proceed as follows:

At the OS1000 READY FOR COMMANDS message on the LPS user interface, enter:

EDIT 1STFRM.FSL

The FSL file is displayed. When you identify where the error occurred, modify that record, then SAVE, CLEAR and END the editing session. Since this FSL already exists, use the abbreviated method to SAVE, CLEAR and END (SCE). Refer to the example in figure 6-11.

Figure 6-11. **Abbreviated method to SAVE, CLEAR and END (SCE)**

```
END;  
SCE  
FDL 1STFRM,TRA
```

When the form compiles error-free, your form prints successfully. Figure 6-12 illustrates the complete FSL and the form it produces.

Figure 6-12. Source statement (1STFRM.FSL)

FDL VERSION B03 13:13 30-APR-XX PAGE 1
 DISK-ID: ARIEL VX.7 R9.7 REL #5 XX/XX/XX

```

1 FORM 1STFRM;                                000010
2 GRID FMT1;                                  000020
3 FONTS UN114A,UN106A,UN104C;                 000030
4 AT 7 DRAW 18 LINES FROM 0 TO 132 USING HAI AND REPEAT EVE 3 000040
5 AT 8 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3 000050
6 AT 9 DRAW 18 LINES FROM 0 TO 24 USING HAI AND REPEAT EVE 3 000060
7 COMMENT *** DRAW VERTICAL LINES ***         000070
8 COMMENT *** DRAW VERTICAL LINES ***         000080
9 AT 24 DRAW VER LINE FROM 5 TO 61 USING HAI  000090
10 AT 32 DRAW 11 VER LINE FROM 5 TO 61 USING HAI AND REPEAT EVE 10 000100
11 AT 29 DRAW VER LINE FROM 10 TO 61 USING BRO HAI 000110
12 AT 38 DRAW 10 VER LINE FROM 10 TO 61 USI BRO HAI AND REPEAT EVE 10 000120
13 AT 8 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3 000130
14 AT 19 DRAW 18 VER LINE FROM 8 TO 9 USI HAI AND REPEAT VER EVE 3 000140
15 COMMENT ***DRAW BOXES ***;                 000150
16 AT 2,2 DRAW BOX 35 WIDE BY 2 HIGH USI HAIRLINE; 000160
17 AT 2,82 DRAW BOX 50 WIDE BY 2 HIGH USI HAI; 000170
18 18 AT 5,0 DRAW BOX 132 WIDE BY 56 HIGH USI HAI; 000180
19 AT 5.1,0 BOX 132 WIDE BY 1.8 HIGH USI SHADING; 000190
20 AT 10.1,0 DRAW 9 BOXES 132 WIDE BY 2.8 HIGH USI SOL HAI 000200
21 INK 'BLUE' AND FILL USING INK 'XEROX.PICTORIAL.PALE BLUE' 000210
22 AND REPEAT VER EVE 6;                      000220
23 COMMENT *** TEXT AT COMMAND ***;           000230
24 TEXT USING FONT 2 AT 2,2.5 'COMP.';        000240
25 TEXT USING FONT 2 AT 2,11 'BR.';          000250
26 TEXT USING FONT 2 AT 2,14 'DEPT.';        000260
27 TEXT USING FONT 2 AT 2,22 'DATE';         000270
28 TEXT USING FONT 2 AT 2,30 'PP';           000280
29 TEXT USING FONT 2 AT 2,34 'PAGE';         000290
30 TEXT USING FONT 1 INK 'BLUE' AT 3,47 'EARNINGS REGISTER'; 000300
31 TEXT USING FONT 2 AT 61,0 '7860-143-1';    000310
32 VER TEXT USING FONT 3 AT 66,-5 'F8662';    000320
33 COMMENT *** TEXT IN BOX COMMANDS ***;      000330
34 TEXT USI FONT 2 IN BOX 5.1,24 'THIS PAY' BOX 'REGULAR' BOX 'OVERTIME'; 000340
35 TEXT USI FONT 2 IN BOX 7,0 'NAME'          000350
36     BOX 'GROSS' 'ADJ. NET' 'NET PAY'       000360
37     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000370
38     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000380
39     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000390
40     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000400
41     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000410
42     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000420
43     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000430
44     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000440
45     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT' 000450
46     BOX 'UNITS-UNITS YTD' 'CURRENT AMT.' 'YTD AMOUNT'; 000460
47 TEXT USING FONT 2 IN BOX 8,19 'EMP.##';    000470
48 TEXT USING FONT 2 IN LEFT CENTER BOX 9,0 'SAL/RATE'; 000480
49 TEXT USING FONT 2 IN BOX 9,0 'GROSS YTD';  000490
50 END;                                        000500

```

A. FDL command summary

This appendix contains an alphabetized summary of the forms description language (FDL) commands described in this manual. The conventions and definitions used in describing FDL commands are shown in Table A-1.

Table A-1. FDL command summary conventions

KEYWORD	Required words in the command appear in uppercase and may be abbreviated to the first three characters (except for FMTn).
<i>id</i>	Identifier consists of 1 to 6 alphanumeric characters (0 through 9, A through Z, "-").
Stacked words	Parameters arranged one above another to denote a choice.
...	Multiple occurrence of the preceding bracketed parameter is permitted.
lowercase	Word, character, or phrase in lowercase is to be replaced by a value.
[]	Optional. If omitted, system substitutes its default.
{ }	Required choice of one of the parameters the braces enclose.
<i>unit</i>	DOTS, IN, INCH, INCHES, CM, or CENTIMETERS. If omitted, defaults to grid size.

Table A-2. FDL command syntax

Command	Syntax
BOX	<p>AT y [<i>unit</i>] x [<i>unit</i>] [DRAW] [n] BOX [ES]</p> <p>[IN <i>unit</i>] p [<i>unit</i>] [WIDE] [BY] q [<i>unit</i>] [HIGH] [USING]</p> <p> $\left[\begin{array}{l} \text{SOLID} \\ \text{BROKEN} \\ \text{DOTTED} \end{array} \right] \left[\begin{array}{l} \text{HAIRLINE} \\ 0 \\ \frac{1}{2} \\ 2 \end{array} \right] \left[\text{[USING] INK } inkref_1 \right] \left[\text{[AND] FILL [USING] INK } inkref_2 \right]$ </p> <p> $\left[\text{[SHADING]} \left[\begin{array}{l} \text{LIGHT} \\ \text{MEDIUM} \\ \text{HEAVY} \end{array} \right] \right]$ </p> <p> $\left[\text{[AND] [REPEAT]} \left[\begin{array}{l} \text{HORIZONTALLY} \\ \text{VERTICALLY} \end{array} \right] \left\{ \text{AT } c_a \text{ [unit]} [c_a \text{ [unit]}] \dots \right\} \left[\text{EVERY } c_i \text{ [unit]} \right] \right];$ </p>
COMMENT	COMMENT <i>text</i> ;
END	END;
FONT	FONT[S] <i>id</i> [,.. <i>id</i>];
FORM	FORM <i>id</i> ;
GRAPHIC	<p>GRAPHIC <i>name</i> AT $vpos$ [<i>unit</i>] ,$hpos$ [<i>unit</i>] [SCALE [IS] n/d]] [[USING] INK $inkref_1$[$inkref_2$] [,...]];</p>

Table A-2. FDL command syntax (continued)

Command	Syntax
GRID	GRID [UNIT] [IS] $\left\{ \begin{array}{l} \textit{format-id} \\ n \text{ INCHES} \\ n \text{ CM} \\ n \text{ CENTIMETERS} \\ n \text{ CPI } n \text{ LPI} \\ [n] \text{ DOTS } [[n] \text{ DOTS}] \end{array} \right\} \left[\text{ORIGIN } y \right]$ $\left[\begin{array}{l} \text{INCHES} \\ \text{CM} \\ \text{CENTIMETERS} \\ \text{DOTS} \end{array} \right] \times \left[\begin{array}{l} \text{INCHES} \\ \text{CM} \\ \text{CENTIMETERS} \\ \text{DOTS} \end{array} \right] ;$
ICATALOG	ICATALOG [IS] <i>inkcatalogname</i> ;
INK	INKS $\left\{ \begin{array}{l} \textit{inkname} \\ (\textit{inkname}, [\text{NOSUBSTITUTION}], \textit{inkname}, [\text{NOSUBSTITUTION}]] [, \dots]) \end{array} \right\}$
IRESULT	IRESULT [IS] $\left\{ \begin{array}{l} \text{DEFAULT} \\ \text{BLACK} \\ \text{COLOR} \end{array} \right\}$

Table A-2. FDL command syntax (continued)

Command	Syntax
<p>LANDSCAPE/ PORTRAIT</p>	$\left\{ \begin{array}{l} \underline{\text{LANDSCAPE}} \\ \underline{\text{PORTRAIT}} \end{array} \right\} \left[\text{PAGE} [\text{SIZE}] [\text{IS}] n \left[\begin{array}{l} \underline{\text{DOTS}} \\ \underline{\text{INCHES}} \\ \underline{\text{CM}} \\ \underline{\text{CENTIMETERS}} \end{array} \right] \right.$ $\left. \left[\underline{\text{WIDE}} \right] [\text{BY}] m \left[\begin{array}{l} \underline{\text{DOTS}} \\ \underline{\text{INCHES}} \\ \underline{\text{CM}} \\ \underline{\text{CENTIMETERS}} \end{array} \right] \left[\underline{\text{HIGH}} \right] \right] ;$
<p>LINE</p>	$\text{AT } c_o [\textit{unit}] [\text{DRAW}][n] \left[\begin{array}{l} \underline{\text{HORIZONTAL}} \\ \underline{\text{VERTICAL}} \end{array} \right] \left[\begin{array}{l} \underline{\text{LINE}} \\ \underline{\text{LINES}} \end{array} \right] [\text{IN } \textit{unit}] [\text{FROM}] c_s$ $[\textit{unit}] \text{ TO } c_e [\textit{unit}] [\text{USING}] \left[\begin{array}{l} \underline{\text{SOLID}} \\ \underline{\text{BROKEN}} \\ \underline{\text{DOTTED}} \end{array} \right] \left[\begin{array}{l} \underline{\text{HAIRLINE}} \\ 0 \\ 1 \\ 2 \end{array} \right] \left[\begin{array}{l} \underline{\text{INK } \textit{inkref}} \end{array} \right]$ $\left[\begin{array}{l} [\text{AND}] [\text{REPEAT}] \\ \left[\begin{array}{l} \underline{\text{HORIZONTALLY}} \\ \underline{\text{VERTICALLY}} \end{array} \right] \left[\begin{array}{l} \text{AT } c_a [\textit{unit}][c_a [\textit{unit}] \dots] \\ \text{EVERY } c_j \textit{ unit} \end{array} \right] \end{array} \right] ;$

Table A-2. FDL command syntax (continued)

Command	Syntax
LOGO	LOGO <i>id</i> AT <i>y</i> [<i>unit</i>] x [<i>unit</i>] [[USING] INK <i>inkref</i> ₁ [<i>inkref</i> ₂] [, ...]];
PALETTE	PALETTE [IS] ' <i>palettename</i> ';
PAPER	<p>For standard paper sizes:</p> <p>PAPER [SIZE] [IS] { USLETTER A4 USLEGAL } ;</p> <p>For nonstandard paper sizes:</p> <p>PAPER [SIZE] [IS] { x [INCHES CM CENTIMETERS DOTS] [BY] y [INCHES CM CENTIMETERS DOTS] } ;</p>
SECTION	[BEGIN] SECTION <i>id</i>; END SECTION; DO SECTION <i>id</i> AT <i>y</i> [<i>unit</i>] x [<i>unit</i>];

Table A-2. FDL command syntax (continued)

Command	Syntax
TEXT AT	<p> $\left[\begin{array}{l} \text{HORIZONTAL} \\ \text{VERTICAL} \end{array} \right] \text{TEXT} \left[\text{[SPACED]} d \left[\begin{array}{l} \text{DOTS} \\ \text{PTS} \\ \text{IN} \\ \text{CM} \\ \text{LPI} \end{array} \right] \text{[PER LINE]} \right]$ $\left[\text{[ALIGNED]} \left\{ \begin{array}{l} \text{LEFT} \\ \text{RIGHT} \\ \text{CENTER} \\ \text{TOP} \\ \text{BOTTOM} \end{array} \right\} \right] \left[\text{[USING] FONT } n \text{ [AND]} \right] \left[\text{[USING] INK } inkref \right]$ AT <i>y</i> [<i>unit</i>] x [<i>unit</i>] '<i>text</i>' ['<i>text</i>'] . . . ; </p>

Table A-2. FDL command syntax (continued)

Command	Syntax
TEXT IN BOX	<p> $\left[\begin{array}{c} \text{HORIZONTAL} \\ \text{VERTICAL} \end{array} \right] \text{TEXT} \left[\text{[SPACED]} d \left[\begin{array}{c} \text{DOTS} \\ \text{PTS} \\ \text{IN} \\ \text{CM} \\ \text{LPI} \end{array} \right] \text{[PER LINE]} \right]$ $\left[\text{[ALIGNED]} \left\{ \begin{array}{c} \text{LEFT} \\ \text{RIGHT} \\ \text{CENTER} \\ \text{TOP} \\ \text{BOTTOM} \end{array} \right\} \right] \left[\text{[USING] FONT } n \text{ [AND]} \right] \left[\text{[USING] INK } inkref \right]$ $\text{IN} \left[\begin{array}{c} \left[\begin{array}{c} \text{LEFT} \\ \text{CENTER} \\ \text{RIGHT} \end{array} \right] \left[\begin{array}{c} \text{LEFT} \\ \text{CENTER} \\ \text{RIGHT} \end{array} \right] \\ \left[\begin{array}{c} \text{TOP} \\ \text{CENTER} \\ \text{BOTTOM} \end{array} \right] \left[\begin{array}{c} \text{TOP} \\ \text{CENTER} \\ \text{BOTTOM} \end{array} \right] \end{array} \right] \text{BOX } y \text{ [unit] } x \text{ [unit] 'text' ['text'] \dots}$ $\left[\text{[IN] [NEXT]} \left[\begin{array}{c} \text{HORIZONTAL} \\ \text{VERTICAL} \end{array} \right] \text{BOX 'text' ['text'] \dots};$ </p>
FDL compiler command	<p> FDL $[\textit{filename}][,\text{NOPRINT}][,\text{TRAY}][,\text{DEBUG}][,\text{XPAN}][,\text{NOSOURCE}][,\text{PROOF}][,\text{SIMPLEX}]$ </p>

B. FDL-generated messages

These messages may appear while forms are being compiled using the forms description language (FDL) compiler.

FD0900	OPERATOR-REQUESTED ABORT. ABORT command was entered. Compiling is stopped.
FD1000	ALL FORMS COMPILED. FORMS COMPILER EXITING. FDL file name command was entered. Forms have been successfully compiled, and FDL compiling session is terminated.
FD1050	FORMS COMPILER RESUMING. System is beginning to compile another form after having completed one previously sent.
FD1800	FORMS COMPILER ABORTING. FDL task is being cancelled. ABO[RT] command was entered, disk error has occurred, or a specified file was not found. Message appears with all fatal error messages.
FD2700	UNRECOGNIZED KEY-IN, KEY-IGNORED. MAY BE RE-ENTERED. A command was entered incorrectly or placed incorrectly in a line. System is ignoring entry. Reenter command correctly.
FD2710	INVALID CHARACTER, RE-ENTER. Symbol or word entered was incorrect for location in command line. Reenter line correctly.
FD2720	KEY-IN TOO LONG. RE-ENTER. Name or word entered exceeds permitted number of characters. Enter different word within permitted range.
FD2730	PARAMETER TOO LONG. RE-ENTER. Command parameter entered contained more characters than permitted. Reenter command correctly.
FD2740	FSL NOT FOUND OR INVALID OPTION. CHECK AND RETRY. The specified forms source library (FSL) file does not exist on the system disk or the option is invalid. Compiler will abort. <ol style="list-style-type: none">1. Check for correct file name.2. Create FSL file on disk if necessary.3. Retry FDL task.

FD2915 SECURITY VIOLATION.
Requested file has been protected by ABNORMAL SECURITY = YES command. Attempt was made to access it at wrong logon level. To access file, obtain correct password and log on at Level 5.

FD4500 ERROR IN READING MAGNETIC TAPE.
System is unable to read input tape containing FDL. Compiler will abort.
1. Clean tape drive.
2. If problem recurs, remake FDL tape.

FD8500 STACK UNDERFLOW.
Forms compiler software error has occurred. Compiler will abort.
1. Retry compiling.
2. If message reappears, call your system support representative.

FD9400 ERROR IN CLOSING THE SOURCE-INPUT FILE
FD9410 ERROR IN CLOSING THE SOURCE-OUTPUT FILE
FD9420 ERROR IN CLOSING THE LISTING/SUMMARY FILE
FD9430 ERROR IN OPENING THE SOURCE/OUTPUT FILE
FD9440 ERROR IN OPENING THE LISTING/SUMMARY FILE
FD9450 ERROR IN READING THE SOURCE-INPUT FILE
FD9460 ERROR IN READING THE SOURCE-OUTPUT FILE
FD9470 ERROR IN WRITING THE SOURCE-OUTPUT FILE
FD9475 ERROR IN BACK-SPACING THE SOURCE-OUTPUT FILE
FD9480 ERROR IN WRITING THE LISTING/SUMMARY FILE
FD9485 ERROR IN READING THE LISTING/SUMMARY FILE
FD9490 ERROR IN BACK-SPACING THE LISTING/SUMMARY FILE
FD9510 ERROR IN CLOSING CONTEXT FILE
FD9520 ERROR IN OPENING CONTEXT FILE
FD9530 ERROR IN READING CONTEXT FILE
FD9540 ERROR IN WRITING CONTEXT FILE

Disk/file error messages

System is unable to access (read, write, open, or close) indicated file due to problem with disk. Forms compiler will abort.

1. Retry entry. Message should not reappear if failure was intermittent memory error.
2. If problem persists, call your system support representative.

C. System default summary

This appendix lists the default parameters that the system automatically substitutes when specifications are omitted in FDL commands. Grid unit dimensions are rounded to the nearest tenth.

The grid and origin substitutions made when printing on 8.5 by 11 inch/216 by 269 mm paper are shown in table C-1.

Table C-1. **Grid and origin substitutions for 8.5 by 11 inch/216 by 279 mm paper**

Specification	Substitution
FMT1	LANDSCAPE; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 0.18 INCH, 0.66 INCH;
FMT2	LANDSCAPE; GRID UNIT IS 15 CPI, 8.1 LPI, ORIGIN 0.18 INCH, 0.50 INCH;
FMT3	LANDSCAPE; GRID UNIT IS 13.6 CPI, 10.7 LPI, ORIGIN 0.14 INCH, 0.66 INCH;
FMT4	LANDSCAPE; GRID UNIT IS 15 CPI, 10.7 LPI, ORIGIN 0.14 INCH, 0.50 INCH;
FMT5	LANDSCAPE; GRID UNIT IS 10 CPI, 6 LPI, ORIGIN 0.17 INCH, 0.50 INCH;
FMT6	PORTRAIT; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 0.57 INCH, 0.58 INCH;
FMT7	PORTRAIT; GRID UNIT IS 12 CPI, 6 LPI, ORIGIN 0.50 INCH, 0.50 INCH;
FMT8	PORTRAIT; GRID UNIT IS 10 CPI, 6 LPI, ORIGIN 0.50 INCH, 0.50 INCH;
FMT9	LANDSCAPE; GRID UNIT IS 20 CPI, 10 LPI, ORIGIN 0.25 INCH, 0.25 INCH;
FMT10	PORTRAIT; GRID UNIT IS 17.6 CPI, 12.5 LPI, ORIGIN 0.22 INCH, 0.51 INCH;
FMT11	PORTRAIT; GRID UNIT IS 20 CPI, 12.5 LPI, ORIGIN 0.57 INCH, 0.39 INCH;

The grid and origin substitutions made when printing on 8.5 by 14-inch/216 by 256 mm paper are shown in table C-2.

Table C-2. **Grid and origin substitutions for 8.5 by 14 inch/216 by 356 mm paper**

Specification	Substitution
FMT12	LANDSCAPE; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 0.18 INCH, 0.66 INCH;
FMT13	PORTRAIT; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 0.57 INCH, 0.58 INCH;

The grid and origin substitutions made when printing on 8.27 by 11.69 inch/A4 paper are shown in table C-3.

Table C-3. **Grid and origin substitutions for 8.27 by 11.69 inch/A4 paper**

Specification	Substitution
FMT1A	LANDSCAPE; GRID UNIT IS 12.5 CPI, 8.3 LPI, ORIGIN 0.46, 1.45 CM (0.18, 0.57 INCH)
FMT2A	LANDSCAPE; GRID UNIT IS 14.3 CPI, 8.3 LPI, ORIGIN 0.46, 1.53 CM (0.18, 0.60 INCH)
FMT3A	LANDSCAPE; GRID UNIT IS 12.5 CPI, 11.1 LPI, ORIGIN 0.46, 1.45 CM (0.18, 0.57 INCH)
FMT4A	LANDSCAPE; GRID UNIT IS 14.3 CPI, 11.1 LPI, ORIGIN 0.46, 1.53 CM (0.18, 0.60 INCH)
FMT5A	LANDSCAPE; GRID UNIT IS 10 CPI, 6 LPI, ORIGIN 0.56, 2.16 CM (0.22, 0.85 INCH)
FMT6A	PORTRAIT; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 2.32, 117 CM (0.91, 0.46 INCH)
FMT7A	PORTRAIT; GRID UNIT IS 12 CPI, 6 LPI, ORIGIN 2.16, 100 CM (0.85, 0.39 INCH)
FMT8A	PORTRAIT; GRID UNIT IS 10 CPI, 6 LPI, ORIGIN 2.16, 100 CM (0.85, 0.39 INCH)
FMT9A	LANDSCAPE; GRID UNIT IS 20 CPI, 10 LPI, ORIGIN 0.36, 2.16 CM (0.14, 0.85 INCH)
FMT10A	PORTRAIT; GRID UNIT IS 17.6 CPI, 12.5 LPI, ORIGIN 1.45, 100 CM (0.57, 0.39 INCH)
FMT11A	PORTRAIT; GRID UNIT IS 20 CPI, 12.5 LPI, ORIGIN 1.44, 100 CM (0.57, 0.39 INCH)

Substitutions made for all paper sizes are shown in table C-4.

Table C-4. **Grid and origin substitutions for all paper sizes**

Specification	Substitution
FORM <i>id</i> ;	FORM <i>id</i> ; GRID FMT1;
FORM <i>id</i> ; PORTRAIT;	FORM <i>id</i> ; PORTRAIT; GRID FMT6;
Page orientation not specified	Landscape
PAPER SIZE not specified	Sysgen-specified paper size is assumed
PAPER SIZE not specified	Virtual page size set to sysgen-specified paper size
Explicit GRID UNIT measurement is specified, and ORIGIN is not specified	Form origin set to virtual page origin
GRID format-id is specified, and ORIGIN is not specified	Form origin set to the BEGIN values of the specified format-id
No GRID UNIT or format-id is specified, and ORIGIN is specified	Grid unit dimensions set by FMT1 (landscape) or FMT6 (portrait)
No GRID command is specified	Grid unit dimensions and form origin set by FMT1 (landscape) or FMT6 (portrait)
Unit not specified in ORIGIN parameter of the GRID command or PAGE SIZE parameter of the LANDSCAPE or PORTRAIT command	Inches
No line direction	Horizontal and repeat vertically
No line/box thickness	SOLID 1
No direction for box repeat	Repeat box horizontally
No text direction	Horizontal text
No box text alignment	Aligned centered in the box
No font index specified in TEXT command	Current font in use
Unit not specified in SPACED parameter of the TEXT command	Dots
No units specified in LOGO command	Grid unit

D. GRID and TEXT command examples

Table D-1 shows grid unit scaling specifications when the grid parameters are omitted or specified.

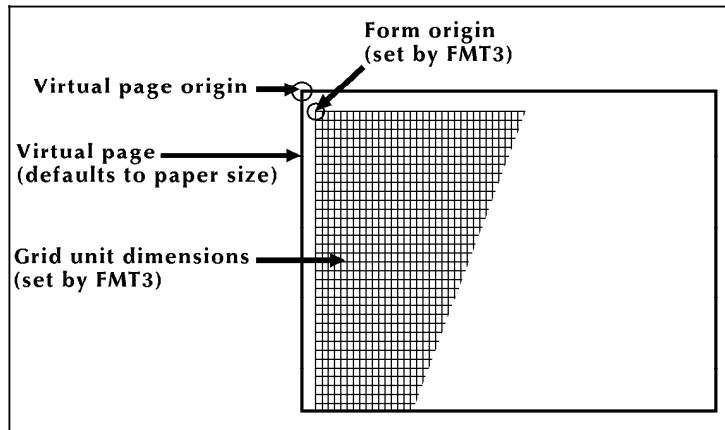
Table D-1. Grid unit scaling specifications

Grid unit specification	System response
Omitted: default scaling	When the system receives the FORM id command, it sets grid scaling to that of FMT1; if it subsequently receives the PORTRAIT command, it sets the scale to that of FMT6.
Specified in GRID command	Replaces default scaling
Specified in BOX or LINE command (IN unit)	Replaces GRID command scaling for that command only (becomes the default scaling).
Specified for a parameter	Overrides any scaling specified via GRID, BOX, or LINE commands for the parameter only. Examples: GRID 1 INCH; AT 2 CM LINE FROM 3 TO 1500 DOTS; COMMENT EQUIVALENT TO: AT 2 CM LINE FROM 3 INCHES TO 1500 DOTS; AT 2 LINE IN CM FROM 3 INCHES TO 1500 DOTS; COMMENT EQUIVALENT TO: AT 2 CM LINE FROM 3 INCHES TO 1500DOTS;

Example 1 LANDSCAPE;
 GRID [UNIT] [IS] FMT3;

Figure D-1 shows a form with the grid specified, but no page size or origin.

Figure D-1. **Form with grid specified, but no page size or origin**

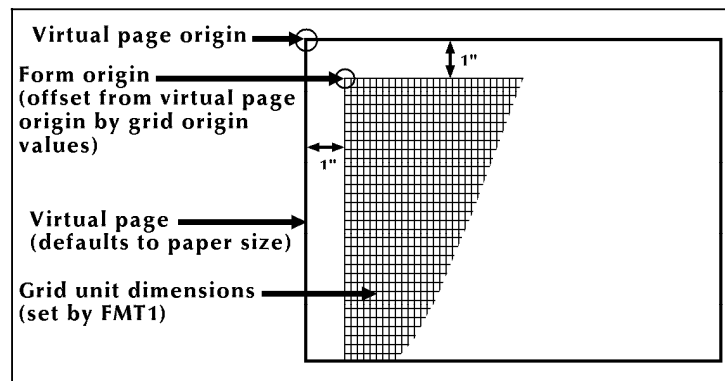


In example 1, no page size is specified, so the virtual page defaults to paper size. FMT3 in the GRID command provides the grid size and form origin (which is offset from the virtual page origin).

Example 2 LANDSCAPE;
 GRID ([UNITS] [IS] FMT1 ORIGIN [IS] 1 INCH 1 INCH);

Figure D-2 shows a form with the origin offset one inch from the virtual page origin.

Figure D-2. **Form with origin offset one inch from virtual page origin**

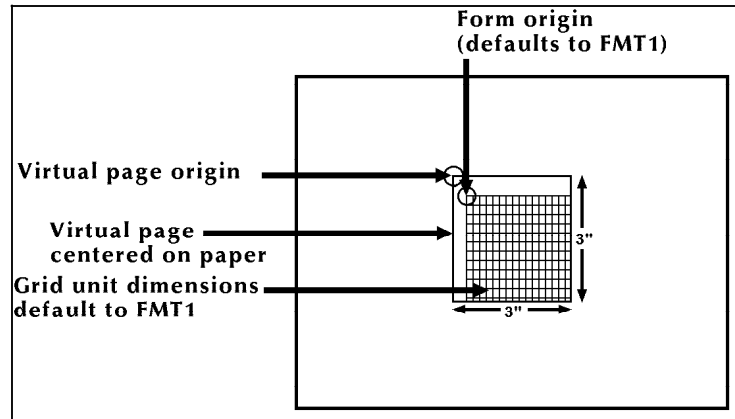


In example 2, the ORIGIN parameter in the GRID command specifies a form origin that is offset from the virtual page origin by one inch vertically and one inch horizontally. Grid unit dimensions are set by FMT1.

Example 3 LANDSCAPE PAGE SIZE IS 3 INCH 3 INCH;

Figure D-3 shows a form with the page size specified, but not the origin or grid.

Figure D-3. **Form with page size specified, but no origin or grid**

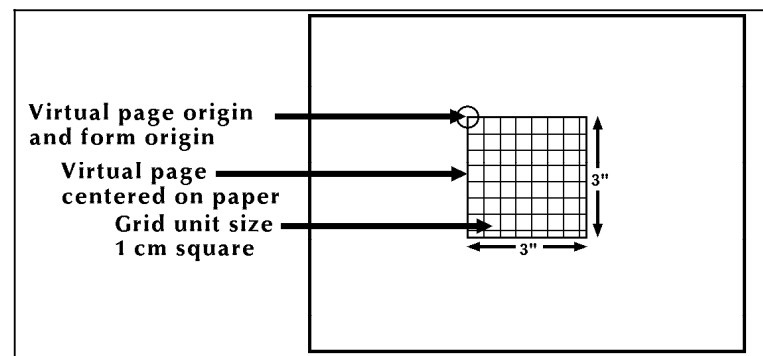


In example 3, the landscape-oriented virtual page is centered on the paper in accordance with the parameters in the LANDSCAPE PAGE SIZE command. Grid size and form origin default to FMT1.

Example 4 LANDSCAPE PAGE SIZE IS 3 INCH 3 INCH;
GRID [UNIT] [IS] 1 CM;

Figure D-4 shows a form with the grid specified, but not the origin or format.

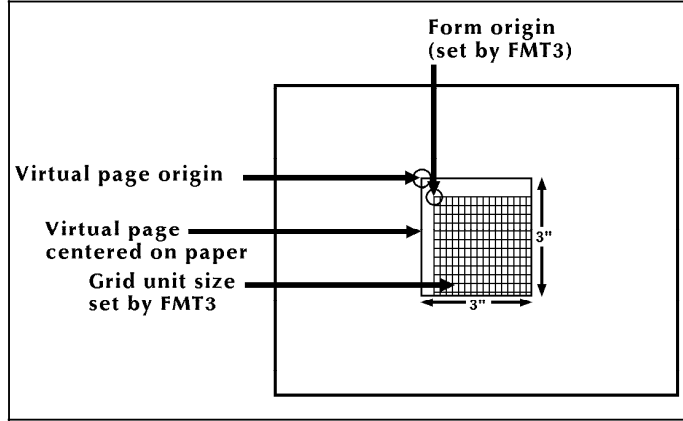
Figure D-4. **Form with grid specified, but no origin or format**



In example 4, explicit grid unit dimensions are set by the GRID command, and no ORIGIN parameter or predefined format is specified. This causes the form origin to coincide with virtual page origin.

Example 5 LANDSCAPE PAGE SIZE IS 3 INCH 3 INCH;
 GRID [UNIT] [IS] FMT3;

Figure D-5 shows a form with the grid and page size specified.
 Figure D-5. **Form with grid and page size specified**

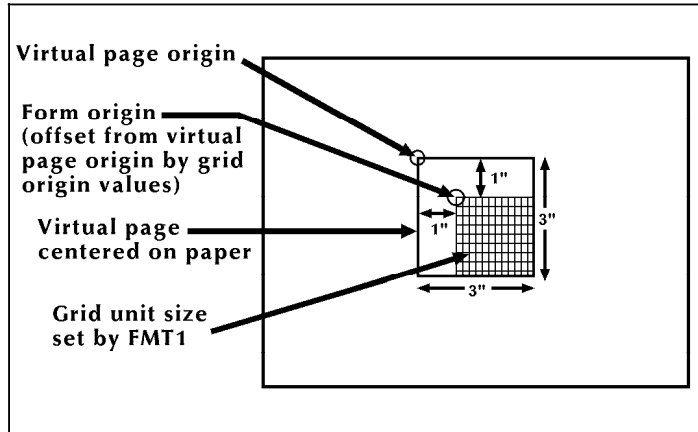


In example 5, FMT3 provides both the grid unit dimensions and form origin.

Example 6 LANDSCAPE PAGE SIZE IS 3 INCH 3 INCH;
 GRID [UNIT] [IS] FMT1 ORIGIN IS 1 INCH 1 INCH;

Figure D-6 shows a form with the grid, origin, and page size specified.

Figure D-6. **Form with grid, origin, and page size specified**



In example 6, the location of the virtual page origin is determined by the PAGE SIZE command. The form origin is offset from the virtual page origin by one inch vertically and one inch horizontally.

Portrait and inverse portrait fonts may be used with landscape page orientation (for example, to insert labels or captions). Likewise, landscape and inverse landscape fonts may be used with portrait page orientation.

Figures D-7 and D-8 illustrate the character block orientations for portrait and landscape pages. Portrait, landscape, inverse portrait, and inverse landscape fonts are listed in the *Xerox Laser Printing Systems Standard Font Library Font User Guide* and the *Xerox 4850/4890 Highlight Color and 4135 LPS Font User Guide*.

Figure D-7. Character block orientations for a portrait page

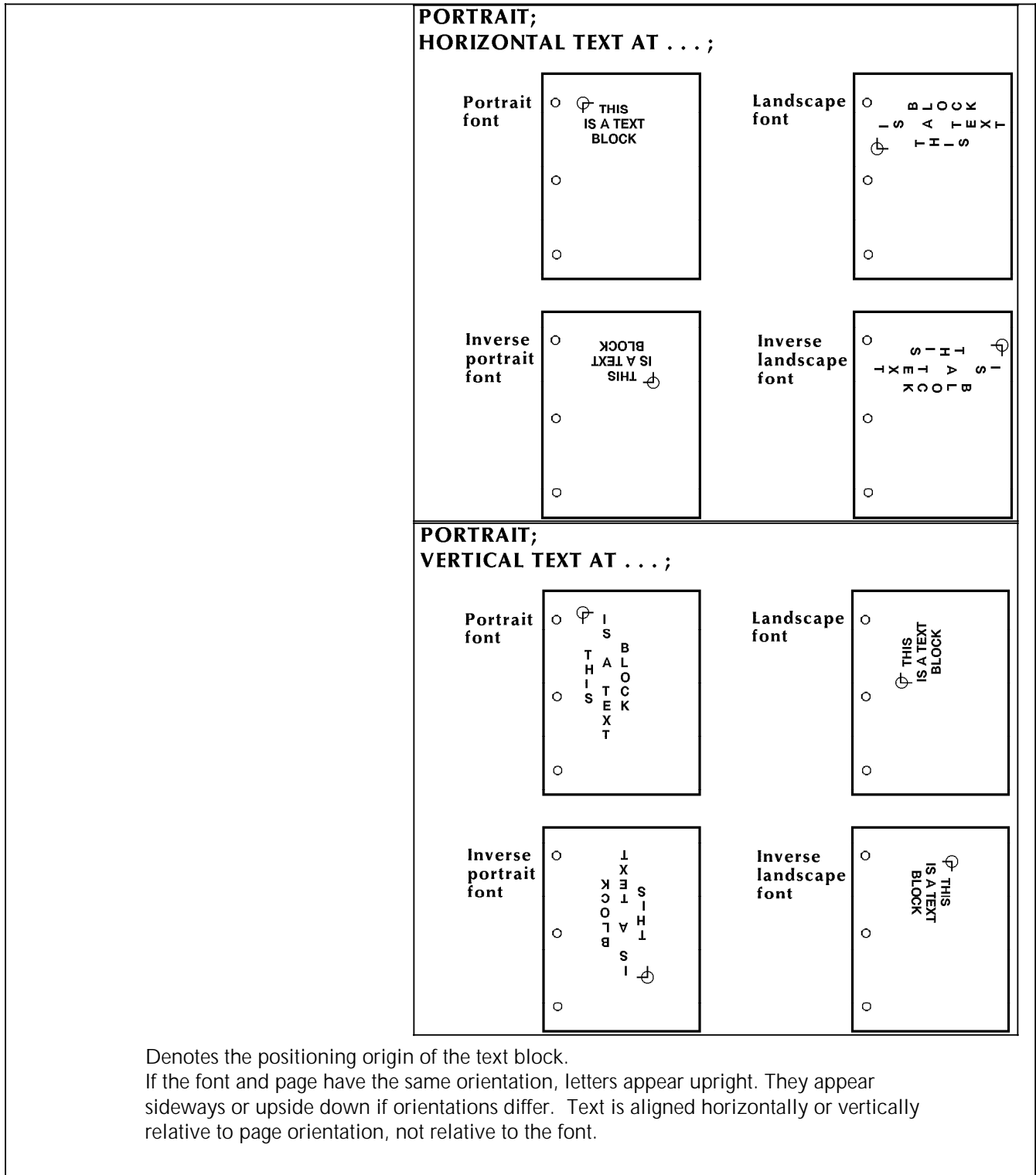
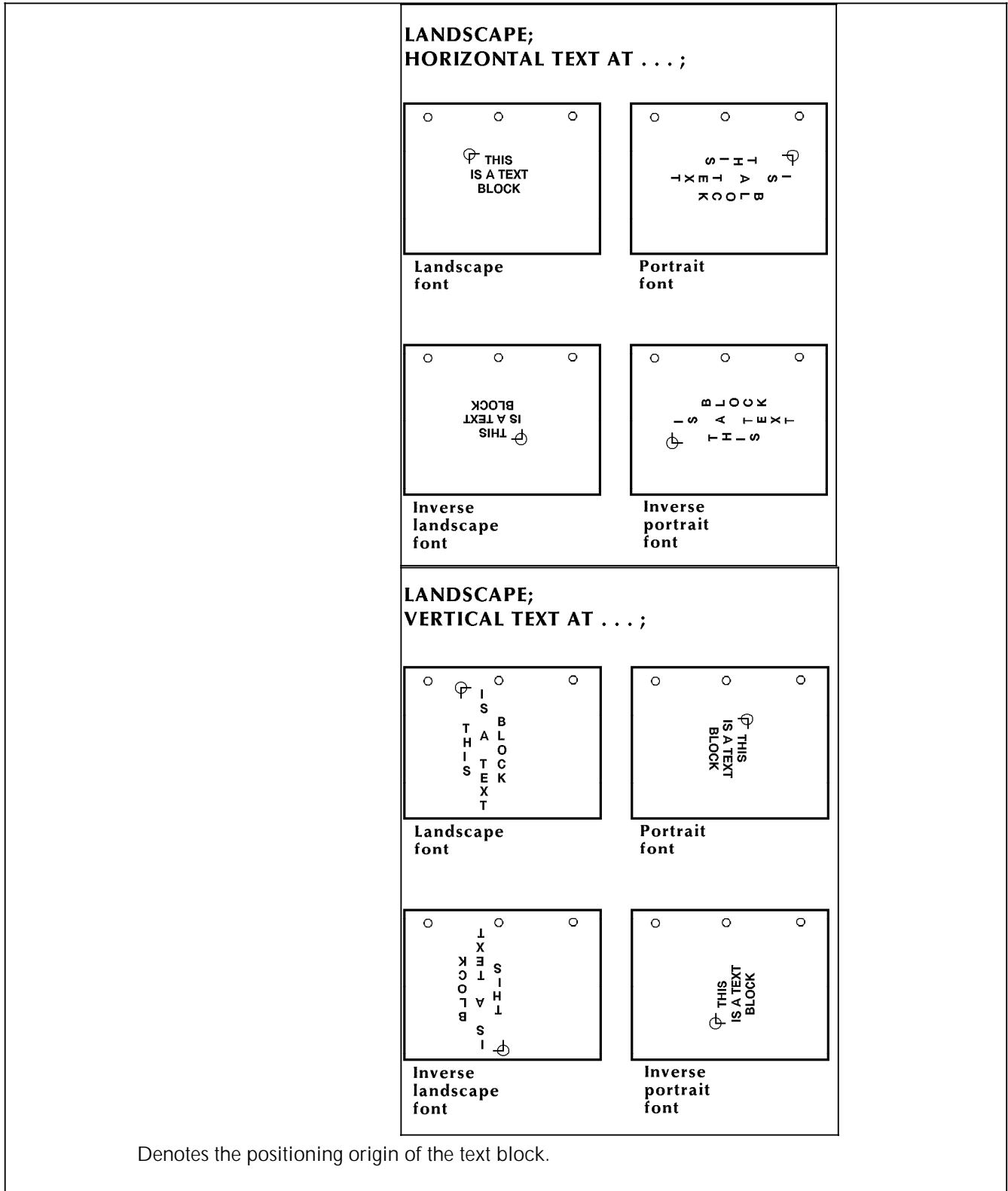


Figure D-8. Character block orientations for a landscape page



E.**FDL capacity limits**

The available memory space imposes the following maximums for use:

Text	Buffer capacity:	512 bytes
	Lines per TEXT statement:	64
Space allocation	TEXT AT command:	16 bytes > S1 > S2 . . . Sn
	TEXT IN BOX command:	18 bytes > S1 > S2 . . . Sn (1st box)
		6 bytes > S1 > S2 . . . Sn (second and all succeeding boxes)
	. . .where	
	n bytes	is fixed overhead.
	S	is number of characters > 2 in a text string, rounded to a multiple of two.
Sections	Definitions:	8 per form
	Invocations:	No limit
Lines	Horizontal:	2000
	Vertical:	2000
Fonts and logos	84	
		This number may vary, depending on memory capacity and use; however, 84 fonts and logos cannot be exceeded.

F. Support tools and measurements

There are two separate functions of forms creation: forms design and forms description. Any form can be converted to an electronic form stored on the system by describing it in forms description language (FDL). Forms design involves laying out and measuring a form and is only a peripheral concern of this reference. However, for the forms designer's convenience, a forms design ruler is provided.

Forms design ruler

The forms design ruler is used for measuring character and line spacings. The ruler has eight scales as shown in table F-1.

Table F-1. **Standard formats and spacings**

Side 1	
Scale 1–10/inch	Standard computer output character spacing; can also be used for measuring inches, graduated in tenths
Scale 2–6/inch	Standard computer output line spacing (6 lpi)
Scale 3–8/inch	Standard computer output line spacing (8 lpi)
Scale 4–12/inch	Can be used for 12 cpi or 12 lpi
Side 2	
Scale 5–13.6/inch	Default character spacing on Xerox laser printing systems
Scale 6–8.1/inch	Default line spacing on Xerox laser printing systems (66 lines per landscape page)
Scale 7–10.7/inch	Line spacing on Xerox laser printing systems for 88 lines per landscape page
Scale 8–15/inch	Can be used for 15 cpi or 15 lpi

The scales can be used for either character spacing or line spacing. For example, scale 1 can be used for either 10 characters per inch (cpi) or 10 lines per inch (lpi). Side 1 of the ruler is typically used for measuring impact-printed computer output reports in which the spacing is 10 cpi (scale 1) and either

6 lpi (scale 2) or 8 lpi (scale 3). Side 2 should be used for measuring output from your laser printing system.

The default spacings on the system are 13.6 cpi (scale 5) and 8.1 lpi (scale 6); they were chosen for optimal readability for the standard computer format of 132 columns by 66 lines. (Refer to table 2-1 for a complete list of standard formats and spacing.)

4890 palette	A predefined set of colors or inks provided with the 4890 HighLight Color LPS. Different versions are provided with the printer and with host- or PC-based application software.
A3	International paper size measuring 297 mm by 420 mm or 11.69 by 16.54 inches.
A4	International paper size measuring 210 by 297 mm or 8.27 by 11.69 inches.
additive primary color	There are five additive primary colors: red, green, blue, magenta and cyan. When light of these five colors is combined in equal amounts, the result is white light.
American Standard Code for Information Interchange (ASCII)	Standard 7-bit code that represents alphanumeric information. Each alphanumeric character and several nonprinting characters are assigned a binary number, covering 128 possible characters. It is used for information interchange among data processing systems, data communication systems, and associated equipment.
ANSI	American National Standards Institute.
application software	Host- or LPS-resident software that directs the computer to perform specific tasks or functions. Common business applications include payroll, accounting, and inventory.
argument	Independent variable of a function, such as a parameter.
ascender	Portion of a lowercase alphabetic character that extends above the main body of characters such as b, d, or h. See also <i>descender</i> .
asynchronous	Transmission in data communications controlled by start and stop characters; thus, time intervals between transmitted data blocks may be unequal in length.
ASCII	American Standard Code for Information Interchange. Standard 7 or 8-bit code that represents alphanumeric characters and several nonprinting characters by assigning each a binary number covering 128 possible characters. It is a common standard for information interchange among data processing systems, data communication systems, and associated equipment.
B4	International paper size measuring 250 mm by 353 mm or 9.84 by 13.9 inches.

batch processing	Process that allows for repetitive operations to be performed sequentially on batched data without much involvement from the computer operator.
background job	Low-priority job, usually batched, which is executed automatically as system resources become available.
baud	Measurement of data flow between devices in bits per second. Common baud rates are 110, 300, 1200, 2400, 4800, and 9600.
BCD	Binary coded decimal.
binary	Numbering system with a base of 2. All numbers are represented by combinations of 1 and 0.
binary synchronous communication	See <i>BSC</i> .
binary synchronous transmission	Data transmission where synchronization of characters is controlled by timing signals generated at the sending and receiving stations.
bit	Abbreviation for binary digit, the smallest unit of information recognized by a computer.
bitmap	Visual representation of graphic images in which a bit defines a picture element (pixel) and a matrix of bits defines an image. For example, if a bit is 1, the corresponding pixel is printed.
bitmapped	Display image generated bit by bit for each point or dot. A software-driven scanner is used to create characters and/or graphics.
bits per second	See <i>bps</i> .
block length	Number of characters or bytes contained in a block of data (the block is treated as a unit within the computer). Block length is usually invariable within a system and may be specified in units such as records, words, computer words, or characters.
blocking	Process of combining two or more records into a single block of data which can be moved, operated upon, stored, and so on, as a single unit by the computer.
BOF	Bottom of form.
boot	To load the initial instructions of a program into memory. These instructions direct the loading of the operating system and application software.
BOT	Beginning of tape.

bpi	Bits per inch.
bps	Bits per second. In serial communication, the instantaneous bit speed a device or channel transmits a character.
BSC	Binary synchronous communications. 1. Data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations. 2. Communication using binary synchronous line discipline. 3. Uniform procedure using a standardized set of control characters and control character sequences for synchronous transmission of binary-coded data between stations.
buffer	Area of memory in which data is stored during transfer from one device to another. Used for: 1. Accumulating data into blocks before storage or processing. 2. Adjusting differences of speed between devices, or between a device and a communicating facility.
byte	Fixed number of bits (in data processing, usually 8) processed as a single binary value.
cache memory	A fast, small memory used to enhance CPU performance, separate from the main processor memory.
CCID	Character code identifier. Code associated with the universal identifier "Xerox" to indicate the version of the Xerox character code standard used to code Interpress strings.
CCU	Customer Changeable Unit. The color housing that fits inside the printer.
central processing unit	Interprets and executes instructions, performs all operations and calculations, and controls input and output units and auxiliary attachments.
channel	1. In data communications, a path or line that enables two or more devices to communicate (sometimes called a circuit, facility, or link). 2. In computers, a path for communication between the central processing unit (CPU) and input and output units, or between the CPU and peripheral devices.
character cell	Area defined by the outside dimensions of a character plus all horizontal and vertical spacing. In electronic printing, the character cell is defined by a bitmap of dots.
character code identifier	See <i>CCID</i> .
character set	Set of all characters defined in a font, including alphabetic, numeric, and special characters such as symbols.
characters per inch	See <i>cpi</i> .

clearing house service	Directory of registered users, services, and other resources, allowing relevant information about each item to be retrieved by name.
clocking	A method of synchronizing sending and receiving data communications devices. Clocking allows synchronous transmission at high speeds.
cluster	Group of related feeder trays, usually containing the same size and type of paper (stock). Each cluster has a name, consisting of one to six alphanumeric characters.
CME	Entry modifying the output printing characteristics of a report on a copy-to-copy basis.
CMT	Character mapping table.
CMYK	A printing industry standard color definition model where all colors are defined in terms of values for the four-color process primary colors: cyan, magenta, yellow, and black. Refer to color definition model.
Code	1. A set of symbols representing data or instructions to a computer. 2. To write a program or routine that instructs a computer to perform specified operations.
code conversion	Translation of one type of character or symbol code to another.
color definition model	A standardized format for defining colors using quantifiable values. The most popular such models are RGB, CMYK, and HSB.
color substitution	The 4890 HighLight Color LPS uses an algorithm to translate a color specified in a printing industry standard color definition model to a tone of the Highlight primary color using a 4890 palette.
command language	A language used to instruct an operating system.
communication line	Telecommunication line connecting devices at one location with devices at other locations in order to transmit and receive information.
communication link	Physical means connecting one location to another to transmit and receive information, such as a data link
compiler	Software that translates instructions written in high-level language into machine language for execution by a system.
composite character	Character that has multiple bitmap references combined into one and stored on disk in font memory.

continuous printing	Refers to Interpress job integrity under any of the following conditions: excessive graphics, forms, or font use problems.
continuous tone	A photograph, rendering, or other similar image that is made of blended gray tones or values that flow into each other gradually and without hard edges.
control program	An operating system program that manages job flow, input/output processing, and other overall system functions and resources.
Copy Modification Entry	See <i>CME</i> .
copy-sensitive	Job in which multiple copies of a report contain different data, such as paychecks and banking statements.
cpi	Characters per inch. Measure of the size of a fixed-pitch font expressed as the number of characters of the font that can be set in one horizontal inch. See also <i>pitch</i> .
CSI	Command status interface.
Data Capture Utility	See <i>DCU</i> .
data communications	Transmission and reception of encoded information over telecommunication lines.
data file	Collection of related data records organized in a specific manner so that each record is similarly structured. An example of this would be a payroll file set up with one record for each employee, last name first, indicating the rate of pay and all deductions.
data link	Physical means of connecting one location with another for communication. This might include the communications lines, modems, and controls that transmit information between two or more stations.
DCU	Data capture utility. LPS function that saves and prints all system controller activity and provides tracing and event logging facilities.
DDCMP	Digital Data Communication Message Protocol.
default	Value assigned to a field by the system if no input is received from the operator.
defined ink	The tone (defined in the ink catalog file and reproduced for your reference on the printed samples of 4890 HighLight Color LPS palettes) as it is applied to the page.

descender	Portion of the lowercase alphabetic character that extends below the main body of characters such as g, p, or y. See also <i>ascender</i> .
DEC	Digital Equipment Corporation.
device	Any piece of hardware other than the CPU (Central Processing Unit).
digitize	To express or represent data in digital (binary) form so that it can be processed electronically.
DJDE	Dynamic job descriptor entry. Command within an input data stream used to modify the printing environment dynamically.
DMA	Direct memory access.
DMR	Data mode ready. Command parameter that designates a Digital Equipment Corporation host system.
document	1. Data medium and the data recorded on it, usually permanent, which can be read by you or a computer. 2. Collection of information pertaining to a specific subject or related subjects.
dot	Picture element (pixel) imaged by a printer. The number of dots imaged per inch measures printer resolution, for example, 300 dots per inch (dpi). See also <i>spot</i> .
dpi	Dots per inch. Indicates the number of dots per inch displayed on a terminal screen or printed to form a character or graphic.
dry ink	Minute particles of resin and carbon black that can accept an electrical charge and create images. Resin and carbon black or color pigment toner are combined with developer to form the dry ink.
DSDD	Double sided double density.
DSSD	Double sided single density.
DSU	Digital signal unit.
duplex	1. Ability of a data communications system to send and receive information simultaneously. 2. In printing, duplex means printing on both sides of the paper.
dynamic Job descriptor entry	See <i>DJDE</i> .
EBCDIC	Extended binary coded decimal interchange code. Coded character set consisting of up to 256 8-bit coded characters.

edge marking	Use of graphic objects (usually lines or boxes) that bleed off the edge of the physical page. See also <i>physical page</i> .
electronic publishing	The integrated production of documents on demand, using digitally stored documents, computerized composition, and electronic printing systems.
elite	Smallest size standard typewriter type which is 12 characters per horizontal inch.
embedded blanks	Blank spaces within a command line.
EMT	Emulator trap. Language instruction.
enabler	Hardware devices or software packages that come with the printer and allow it to perform as specified.
ENET	Ethernet network.
EOF	End of file.
EOT	End of tape.
escapement	Lateral positioning of characters or font families.
Ethernet	Xerox developed local area network (LAN) that allows transmission of data by cable from one device to another. A modified version of the Ethernet specification has been approved as IEEE standard 802.3.
Extended binary coded decimal interchange code	See <i>EBCDIC</i> .
extended metrics	Measurements used in Interpress to alter the size of fonts, allowing more precision with character escapement. Used for rendered characters.
FCB	Forms control buffer. Controls the vertical format of printed output.
FCP	File control parameter.
FCG	Finishing Configuration Utility.
FCU	File Conversion Utility.
FDL	Forms description language. LPS-resident source language used to design electronic forms. See also <i>FSL</i> and <i>form</i> .

FDR	File directory.
feedback	Portion of an output signal that is returned, directly or indirectly, to be compared to a reference signal to maintain the quality of the output signal.
FFM	Font file management.
firmware	Permanent programs stored in read-only memory (ROM).
fixed font	Font containing characters with fixed spacing.
fixed pitch	Font set with every character cell having the same width. In reference to character sets, this term describes typefaces with all character cells having equal width. Monospaced as opposed to proportional spaced.
fixed spacing	Arrangement of characters on a line so that all characters occupy the same amount of horizontal space.
flag	Small indicator marking the occurrence of an event or the existence of a certain condition while the program is executing.
floating accent	Nonspacing accent characters that can be combined with characters and printed as a composite.
font	Complete set of characters of a particular font family having the same point size, weight, stress, and orientation.
Font Interchange Standard (FIS)	A standard that defines the digital representation of fonts and character metrics for the generation of an entire series of Interpress fonts.
form	1. Compiled .FSL file. 2. Specific arrangement of lines, text, and graphics stored in an electronic version. Forms can be printed without variable data or merged with variable data during the printing process. See also <i>FDL</i> and <i>FSL</i> .
format	1. Layout of a document, including margins, page length, line spacing, and typeface. 2. In data storage, the way the surface of a disk is organized to store data. 3. To prepare the surface of a disk for acceptance of data.
format line	Line preceding the message text in a formatted message display.
form feed	Keyboard and printer control character that causes the printer to skip to the top of the next page.
forms control buffer	See <i>FCB</i> .

forms description language	See <i>FDL</i> .
forms source library	See <i>FSL</i> .
FPS	Formatting print service.
FSL	Forms source library. Uncompiled collection of user-created files containing FDL commands. See also <i>FDL</i> and <i>form</i> .
FST	Font specification table.
fuse	To affix dry ink to paper by heat or pressure or a combination of both.
GB	Gigabyte. Unit of approximately one billion bytes.
GCR	Group code recording.
GHO	Graphics handling option.
gigabyte	See <i>GB</i> .
group code recording (GCR) mode	Refers to the specific density of data (such as 6250 bpi) as it is recorded on tape, which is measured in bits per inch (bpi).
gsm	Grams per square meter.
GVG	Graphics video generator.
halftone screen	A tool used in offset printing, typesetting, and laser printing to convert a continuous tone (such as photographic) image to dots, which allows the image to be rendered accurately in these printing processes.
hardcopy	Machine output in permanent form, such as printed reports and listings. Output in a permanent form (usually on paper or paper tape) rather than in temporary form, as on a display. Contains readable printed copy of machine (such as computer) output.
hardware	Physical components (mechanical, magnetic, electronic, and so on) of a system, as opposed to programs, procedures, rules, and associated documentation. The hardware is operated by software and firmware.
HCF	High-capacity feeder.
HCSS	High-capacity stitcher/stacker; also referred to as stitcher/stacker.

hexadecimal	Numbering system with a base of 16. The numbers 10 through 15 are represented by A through F.
HFDL	Host forms description language.
hierarchy	Relative priority assigned to arithmetic or logical operations that must be performed.
high-level language	Programming language consisting of words and symbols that are close to normal English and, also readily understandable by the user. High-level source languages are used for most commercial programs.
highlight color	Printing with black plus another color. A range of colors, tints, and shades is printed by varying the percentage of black dots, colored dots, and the white space between the dots.
HIP	Host interface processor.
host	Computer accessed by users that serves as a source of high-speed data processing for workstations with less computer power. Commonly referred to as mainframe.
host interface	Connection between network and host computer.
HSB	A printing industry standard color definition model in which all colors are defined in terms of values for hue, saturation, and brightness. Refer to <i>color definition model</i> .
hue	The name that describes a color's general appearance, such as red, green, and blue.
identifier (id)	Character(s) used to identify or name data and possibly to indicate certain properties of that data image area on a physical page that may contain text or graphics.
IFU	Interpress font utility. A utility for managing Interpress fonts.
image area	Area on a physical page that may contain text or graphics.
initialize	1. To prepare a blank diskette so it can accept data. This is usually accomplished when a program is booted. 2. To set all information in a computer system to its starting values.
initial program load	See <i>IPL</i> .
ink catalog	File containing inks that is referenced by all software for color information. The ink catalog file has the extension <i>.ICT</i> .

ink referencing	Process by which inks within the ink catalog are referenced in PDL and FDL commands, and by page description language interpreters.
ink substitution	Option of aborting the printing process or continuing with a substitute ink if the referenced ink is not found.
input devices	Keyboards, magnetic media, or any device used to give a system information.
input/output	General term encompassing the flow of data into and out of a system. Also referred to as I/O.
interface	The device that connects two systems to allow them to communicate.
Interpress	Industry-standard page description language developed by Xerox. Interpress documents can be printed on any sufficiently powerful printer equipped with Interpress print software.
Interpress font utility (IFU)	Utility for managing fonts.
Interpress font utility (IFU) program	Program used to convert FIS fonts to LPS fonts.
Interpress master	File written according to the Interpress standard.
IPD	Interpress decomposer.
IPL	Initial program load. For the optional 9-track magnetic tape drive, the internal initialization sequence whereby certain functions are loaded into random access memory (RAM).
IPM	Interpress mapping.
ips	Inches per second.
JCB	Job control block.
JCL	Job control language.
JDE	Job descriptor entry. Collection of job descriptions.
JDL	Job description library. Collection of compiled job descriptions. See also <i>JSL</i> .
JID	Job identifier.
job descriptor entry	See <i>JDE</i> .

job descriptor library	See <i>JDL</i> .
job source library	See <i>JSL</i> .
JSL	Job source library. Collection of uncompiled job descriptions. See also <i>JDE</i> and <i>JDL</i> .
keyword	Required part of a command.
label	Reference to a file saved on tape or disk, a record indicating the file name or date created, or other control information.
LAN	Local area network. LAN is the lower two layers of the network architecture: the physical layer and the data link layer
landscape	Orientation in which text and images are positioned parallel to the long edge of the paper.
laser printing	Technology that uses a laser to transfer character forms to a page by direct or indirect means.
latent image	Static charge present on the photo conductor before contact with dry ink particles.
leading	(pronounced <i>ledding</i>) Vertical distance between lines of type (also called line space), measured from a baseline of one line to the baseline of the next.
legal size	Paper size measuring 8.5 by 14 inches or 216 by 356 mm.
letter size	Paper size measuring 8.5 by 11 inches or 216 by 279 mm.
light emitting diode (LED)	Solid substance that glows when a current is passed through it. Often used for indicator lights on disk drives or modems, as well as for displays on other electronic equipment.
LF	Line feed character.
line feed	Control character that (unless set to be interpreted as a line end) causes the printing system to begin printing in the current character position of the next line.
literal	Alphanumeric character beginning with a letter, including an asterisk, period, colon, or slash, and not enclosed in single quotes.
logical page	In Xerox printing systems, a logical page is a formatted page that is smaller than the physical page. A logical page is defined by an origin, thus allowing more than one logical page to be placed on a physical page.

logo	Small illustration or design, usually simple, typically used to identify a company.
long-edge feed	See <i>LEF</i> .
lpi	Lines per inch.
LPS	Laser printing system.
LUN	Logical unit number. A number that identifies a peripheral device to the OSS.
magnetic media	Term for all storage devices (disks, tape, and so on) on which data is stored in magnetic form.
map	To establish a set of values having a defined correspondence with the quantities or values of another set.
mask	Selection of bits from a storage unit by use of an instruction that eliminates the other bits in the unit. In accessing files, a file name mask is used to reference one or more files with similar file-id (identifier) syntax. In Interpress, a mask serves as a template, indicating the shape and position of an object on a page.
master file	File serving as a general reference point for a particular application system and providing information to be used by the program. It is usually updated and maintained to reflect the results of current or daily processing operations.
MB	Megabyte. Unit of approximately one million bytes.
metacode	Method of controlling the image generator. The character dispatcher uses these codes to generate scan line information. This information is sent in the form of character specifications to the image generator, which uses it to compose the bit stream that modulates the laser. Also called native mode.
MHz	Megahertz. One million cycles per second. Used to measure electromagnetic waves.
mixed environment	Multiple printers or printing systems in the same location.
modem	Device that converts digital information into an analog signal suitable for sending over analog telecommunication lines. Also converts an analog signal from telecommunication lines into digital information.
monochrome	Printing in one color only.

nesting	Subroutine or set of data, such as a comment, contained sequentially within another set of data.
network	System of hardware and software that manages communication and sharing of common resources (such as printers) between computers and terminals for multiple users at once.
node	Station, terminal or computer operating in a network environment.
object file	Source file converted into machine language (binary code).
octal	System of representing numbers based on 8.
offline	Devices not under the active control of a central processing unit. For example, a computer makes output to a magnetic tape. The tape is then used by an offline printing system to produce printed data. Offline operations are much slower than online operations. Refer to <i>online</i> .
offset	To place printed output sets in slightly different positions from each other in an output bin for easy separation of collated sets.
offset printing	Widely-used method of commercial and corporate printing, in which ink is picked up by a metal or paper plate, passed to an offset drum, then passed to the paper.
online	Devices under the direct control of a central processing unit, such as a printing system in interactive communication with a mainframe. Refer to <i>offline</i> .
operand	That which is acted upon, for example, data, in an operation or process.
operating system	Software that controls the low-level tasks in a computer system, such as input or output and memory management. The operating system is always running when the computer is active.
orientation	In reference to image area, describes whether the printed lines are parallel to the long edge of the paper (landscape) or the short edge of the paper (portrait).
origin	In reference to image area, the upper left corner of a sheet.
output	1. Material produced by a peripheral device of a computer, such as a printout or a magnetic tape. 2. The result of completed operations.
overprint lines	Print lines whose carriage control specifies printing with no line spacing after the last printed line.

overprint ratio	Maximum number of variable data and form characters that can be intersected by a single scan line.
page end	Command character (form feed) to terminate the current page.
page orientation	Direction that data is printed on a report. Refer to <i>landscape page orientation</i> and <i>portrait page orientation</i> .
pagination	Process of separating text into pages.
palette	Predefined set of colors or inks. Different versions are provided with the printer and with some application software packages.
parameter	Part of a command, other than the keyword. See <i>keyword</i> .
pass-through job	On systems with XPAF, a job that is sent directly from a host to a Xerox printer through XPAF without undergoing XPAF processing.
parse	To read or interpret a command; to build up a parameter list from information within a command.
PCC	Printer carriage control.
PDE	Page description entry.
PDL	Print description language. Language used to describe printing jobs to a laser printing system. PDL describes the input (type, format, characteristics), performs the processing functions (logical processing), and describes the output (type, format, font selection, accounting options).
PE	Phase encoded.
physical page	Actual page size your printer uses to print a form.
pica	1. Unit of measurement equal to twelve points or approximately 1/6 inch. 2. A 10-pitch typeface having ten characters per inch and 12 points in height. See also <i>point</i> .
pitch	Width of a fixed-pitch font expressed in characters per horizontal inch.
pixel	Acronym for picture element. Smallest addressable point of a bitmapped screen that can be independently assigned color and intensity.
point	In Xerox laser printing systems, a unit of measurement equal to 0.0139 inch. Points are always used to express type, size, and leading. There are 12 points to a pica and about 72 points to an inch. See also <i>pica</i> .

point size	Height of character set from the top of its ascenders to the bottom of its descenders in units (points). Point size does not always include leading.
portrait	Orientation in which text and images are positioned parallel to the short edge of the paper.
PostScript	Proprietary page definition language, compatible with the Xerox 4890 HighLight Color LPS when a front-end conversion utility is installed.
ppm	Pages per minute.
print quality adjustment	See <i>PQA</i> .
primary color	A color that, when combined with one or more other primary colors in a color model system in varying quantities, produces the palette of colors described by that model.
print description language	See <i>PDL</i> .
print ratio	The maximum number of variable data and form characters that may be intersected by a single scan line.
printer subsystem controller	See <i>PSC</i> .
print file	Portion of the system disk memory (up to 4 MB) reserved for temporary storage of formatted pages for printing. Pages are retained until they are delivered to the output tray.
PQA	The process the 4890 HighLight Color LPS uses to maintain the electrostatic state of the xerographic subsystem to ensure optimal print quality.
printout	Informal expression referring to almost anything printed by a computer peripheral device.
PROM	Programmable read-only memory.
prompt	Message or symbol displayed on a system console requiring the operator to take action.
process color printing	In process color printing, every color in the spectrum is printed by combining various percentages of the four-color process primary colors—cyan, magenta, yellow, and black.
processor	Applies to any system that is capable of receiving and performing operations upon data and supplying the results of those operations.

protocol	Formal set of conventions governing format of data and control of information exchange between two communication devices.
proportional font	Font containing characters that vary in width.
proportional spacing	Text where each alphanumeric character is given a weighted amount of space. Such output has print-like appearance. Proportional spacing allows more space for wide characters and less space for narrow characters.
proportional type	Characters that vary in width.
PSC	Printer subsystem controller. Command and task that download the printer and/or raster image processor (RIP) software if selected.
query	Request for data or other information, entered by an operator while the system is processing.
queue	Managed database of documents waiting to be processed.
RAM	Random access memory.
raster data	Binary data, usually consisting of dots arranged in scan lines, according to the print order.
raster image processor	See <i>RIP</i> .
rasterization	The creation of a bitmap image on a page for printing.
read-only memory	See <i>ROM</i> .
read/write head	A small electromagnet that reads, writes, and erases data in the form of magnetic dots on the surface of an external storage medium, such as a magnetic disk.
real time	To process transactions as they are entered into a system. User and system interchange allow transactions to be processed and the results returned to the user.
remote access	Access to a central computer by terminals or devices geographically separated from that computer.
RGB	A printing industry standard color definition model where all colors are defined in terms of values for the three additive primary colors: red, green and blue. Refer to <i>color definition model</i> .
ROM	Solid-state memory for programs. It cannot be rewritten.

rendered character	Character whose bitmap is a combination of other bitmaps positioned relative to each other, such as accented characters and single characters consisting of two or more characters. A rendered character does not require additional disk space or font memory.
report	In setting a separation boundary, report refers to a subset of a job. A job may consist of one or more reports.
resolution	Number of dots per inch (dpi) or spots per inch (spi). The greater the number of dots, the higher the resolution and the clearer the image. The terms dots, spots, and pixels are synonymous.
RIP	Raster image processor. LPS option that supports the Interpress V3.0 Publications Set.
scale	To adjust font or image size according to given proportions.
sci	START command index.
scroll	Manipulation of a display to bring upper or lower portions of a document page into view when available space is insufficient to view the entire document at once.
SCSI	Small computer system interface. Common standard for connecting devices to computers.
SDI	System diagnostics interface. OSS task used by service personnel to diagnose system problems.
secondary storage	Form of storage external to a system, such as magnetic tapes or floppy disks.
security	1. Procedure for limiting access to the system's resources, programs, or files, to authorized personnel. 2. Protecting programs and files from unintentional or undesired modification.
SEF	See <i>short-edge feed</i> .
sequential	1. In numeric sequence, usually in ascending order. 2. A file structure in which records are written one after another and cannot be randomly accessed.
set	Multiple copies of the same report.
SFC	Status file converter.
SFS	Status file service.

shade	Color resulting from adding black to a pure hue. See also <i>tint</i> and <i>tone</i> .
short-edge feed	Movement of paper through the printer in the direction of the paper width (the shorter side of a sheet of paper).
SIF	Sequence insert file.
simplex printing	Printing on one side of the page.
small computer system interface	See <i>SCSI</i> .
SNA	System network architecture. Total description of logical structure, formats, and protocols of operation sequences for transmitting an information unit through the communication system.
special processing	Commands allowing the user to process special reports such as printing certain records, and printing on special paper
spi	Spots per inch. See <i>resolution</i> .
spooling	Process of releasing data from main memory and storing it temporarily until a peripheral device is ready to accept it, for example, storing print data before sending it to a printer.
spot	A picture element imaged by the printer. Synonymous with <i>dot</i> and <i>pixel</i> .
statement	Detailed instructions in a program step, written according to specific rules called syntax.
static data	Information usually found on preprinted forms or overlays.
station	1. In data communications, a terminal device connected to a data link. 2. In computer networks, any of the numerous terminal devices that form a network.
stock	User-defined name in the JSL that specifies a certain type of paper for printing a job.
stock descriptor	A stock reference or a stock name.
stock name	The name of a cluster as defined by a system command.
stock reference	An application specific pseudonym for a cluster.
stockset	Collection of stocks to be used on a print job. See also <i>stock</i> .

string	Connected sequence of alphanumeric characters treated as one unit of data by a program.
symbiont	Device which shares a common purpose and close association with other devices. This term is used to identify the Xerox print software which resides in the Digital host system as part of the XPMF-VMS communications link.
symbol	Character used in a computer language to specify a particular function.
synchronous	Efficient encoding of data suitable for high-speed, block-oriented data transmission by using equal time elements.
syntax	Rules governing the structure of expressions in a programming language.
syntax error	System response to a mistake in a command's entry.
sysgen	Abbreviated form of system generation.
System	1. In data processing, a collection of computer components and procedures organized to accomplish a set of specific functions. 2. Assembly of components united by some form of regulated interaction to form an organized whole. 3. Operations or procedures used to accomplish a business activity.
System controller	The part of the LPS that provides interfacing capability, data handling, formatting, buffering, and operator control for the system. Also called the system controller.
system disk	Magnetic disk storage medium, usually of large capacity, that is not removable as opposed to floppy disk or disk packs.
system file	Master software program that keeps all components working together.
system generation	Process whereby the system is made ready to operate. Typically involves selecting the operative parameters and activating the relevant software.
System Network Architecture	See <i>SNA</i> .
system page	Maximum area in which text and graphics can be imaged on a printing system.
tape density	Expression of the format of a magnetic tape measured in number of bytes that can be stored per inch of tape.

tape drive	Input/output device that controls the movement of magnetic storage tape past the read/write head while data is accessed or stored.
telecommunications	The transfer of data through telephone lines.
teleprocessing	Data transfer through telecommunication lines for processing among various remote terminals and the central processing unit (CPU).
templates	1. Preset document formats, usually furnished along with application software, such as electronic spreadsheets or data base programs. 2. Also applies to keyboard overlays showing function keys for particular software packages.
temporary storage	Main memory locations reserved for intermediate results of processing, control values, or other information that needs to be kept on hand as a program proceeds.
terminal	Interface device connected to a computer or network. A terminal has no processing capability of its own.
throughput	Measure of the number of pages printed during a given unit of time, usually expressed as pages per minute.
tint	Color resulting from adding white to a pure hue. See also <i>shade</i> and <i>tone</i> .
TOF	Top of form.
toggle	In an LPS system, to switch (alternate) from one tray to another. The system will switch from an active feeder or stacker tray to an inactive one to allow continuous printing when the proper commands are invoked.
token	A coded character representing a word used in programs. For instance STOP is a word and the one-byte token for it in a BASIC dialect is 250.
tone	Primary color in its pure form, or mixed with black or white to create a lighter tint or darker shade of the primary color. See also <i>shade</i> and <i>tone</i> .
total xerographic convergence (TXC)	For the 4890 HighLight Color LPS, a function of the operating system software (OSS) that monitors the state of the color and black developer housings. If TXC detects a condition that may affect print quality, TXC requests that the operator initiate Print Quality Adjustment (PQA) to adjust the xerographic subsystem. Refer to <i>print quality adjustment</i> .
tpi	Tracks per inch.

transaction processing	Method of data processing that updates files and results are generated immediately after data entry.
translation	1. In data communications, the conversion of one code to another on a character-by-character basis. 2. In programming, the function of a language processor that converts a source program from one programming language to another.
transmission speed	In data communications, the rate data is passed through communication lines, usually measured in bits per inch (bpi).
Tri-level xerography	The feature of the 4890 HighLight Color LPS that enables the rasterization of both the black and the color images in a single pass by the creation of a third charge state.
truncated	Cut off before completion, as when data transfer from a host to a printer is cut off before all data has been transmitted.
two-up	Printing two logical pages on one side of a physical page.
TXC	Total xerographic convergence.
type style	Italic, condensed, bold, and other variations of typeface that form a type family.
typeface	Set of fonts having identical design features that give the type a uniform appearance.
UCS	Universal Character Set. Printer feature that permits the use of a variety of character arrays.
UCSB	Universal Character Set Buffer.
UI	User interface.
Universal Character Set	See <i>UCS</i> .
utility program	General-purpose program that performs activities, such as initializing a disk or sorting, which are not specific to any application.
validation	Process of testing a system's ability to meet performance objectives by measuring or monitoring its performance in a live environment.
variable data	Changeable information which is merged with a standard document to create specialized or personalized versions of that document. Variable data is not a part of a form design, but varies from page to page.

variable text	Text of changing nature (such as names and addresses) combined with a form letter to make a complete document.
verification	Process of testing a system's ability to meet performance objectives by running programs in a simulated environment.
virtual page	Page area selected by a forms designer for printing.
vpos	Vertical positioning.
WAN	Wide area network.
weight	Perceived blackness of a character affected by varying the width of the stroke. Weight is expressed in general terms as either bold or roman.
wildcard	Character (usually an asterisk *) which can be inserted into a command string to indicate that it may represent one or more characters in that position.
write protection	Data protection feature implemented on magnetic media (for example, floppy disk, 9-track tape) to prevent stored data from being modified, written over, or erased.
x axis	Horizontal axis on a forms grid.
x height	Height of lowercase letters without their ascenders or descenders (height of letter "x").
xdot	Unit of measurement representing a fraction of an inch. May also be referred to as a picture element (pixel) or spot; for example, 1/600 spots per inch (spi).
xerographic engine	Component of a printer that develops an image, transfers it to paper, and fuses it for output as hardcopy.
xerographic mode	Either of two possible printer configurations: 1. Black mode which allows printing with black dry ink only. 2. Highlight mode which enables both highlight color and black printing.
xerographic mode persistence (XMP)	For the 4890 HighLight Color LPS, a value specified during sysgen, in JDL or in DJDE code that specifies on what basis xerographic mode switching (XMS) may take place. Refer to <i>xerographic mode switching</i> .
xerographic mode switching (XMS)	A function of the operating system software (OSS) that controls the xerographic mode of the 4890 HighLight Color LPS. The operator or programmer controls XMS through the xerographic mode persistence (XMP) setting. Refer to <i>xerographic mode</i> .
XICS	Xerox Integrated Composition System.

XJCF	Xerox Job Control Facility.
XNS	Xerox Network Systems is the network architecture and protocol used with Xerox equipment.
XPAF, XPF	Xerox Printer Access Facility.
XPPI	Xerox Pen Plotter Interface.
XPS (Xerox Print Service)	Xerox Print Service (XPS) Manager is a printer front end to other hosts. XPS Manager server communicates with the printer via a layered communication protocol, based on the XNS protocol suite but enhanced with "online" capability.
XPS (Xerox Publishing System)	Xerox Publishing System.
XPMF-VMS	Xerox Print Management Facility-VMS Version.
y axis	Vertical axis on a forms grid.

Numerals

- 0 lines, 3-12
 - for positioning text in boxes, 3-24
- 300 spots per inch
 - density restrictions, 5-3

A

- advantages of FDL, 1-1
- aligning text, 3-22
 - avoiding imperfect alignment, 5-13
 - in boxes, 5-9
 - on a landscape page, 5-4
 - on a portrait page, 5-4
 - problems with rounding variable data, 5-13

B

- basic concepts, 2-1
- BEGIN value, 3-6
- BEGIN SECTION command, 3-31
- BOX command, 3-2, 3-17, 6-4, 6-6, A-2, A-7
 - conditions for use, 5-6
- boxes, 3-2
 - adjacent, 3-26
 - placing text in, 3-24
 - centering text, 5-9
 - drawing, 3-16, 6-4
 - locating the closest, 5-7
 - positioning, 2-5
 - shading, 5-10
 - suggested techniques, 5-14
 - text in, 3-23, 5-7, 6-5
 - error message, 5-7
- broken line, 3-12

C

- capacity limits, FDL, E-1
- centering text, 3-22
 - in boxes, 5-9
- character
 - cell, 2-2, 2-6
 - spacing
 - fixed, 2-1
 - proportional, 2-1
- coding sequence, recommended, 5-14
- color option, 3-7
- command
 - format, 3-1
 - summary, FDL, A-1
- COMMENT command, 3-2, 3-33, 6-2, A-2
- comments
 - inserting COMMENTS, 3-33
- compiling forms, 3-1, 4-2, 6-8

- invoking compiler commands, 4-2
- PROOF option, 4-3
- secured files, 4-5
- storing compiled forms, 4-4
- complexity, image, 5-1
- converting
 - other unit values to dots, 5-12
 - preprinted forms, 5-14
- coordinates, x and y, 2-7, 3-12, 3-17, 3-20, 3-23
 - BOX command, 3-17, 5-6, 6-4, A-2
 - LINE command, 3-12, 6-2, A-4
 - TEXT AT command, 3-19, 6-5, A-6
- cpi value, 2-7

D

- data types
 - forms, 2-9
 - variable, 2-9
- DEBUG option
 - for compiling forms, 4-2
- default
 - paper size, 3-4
 - summary, system, C-1
- density
 - line, error messages, 5-3
- description commands, 3-12
- designing new forms, 5-14
- directory
 - storing forms in, 4-4
- DO SECTION command, 3-2, 3-30
- dotted line, 3-16, A-2, A-4
- duplex
 - specifying for summary sheet, 4-3

E

- edgemarking
 - system page limitations, 2-3
- elements
 - form, 2-5
- END command, 3-33, 6-7, A-2
- END SECTION command, 3-30, A-5
- error list
 - failed compilation, 4-3
- error messages
 - list of all FDL, B-1
- errors
 - line density, 5-3
 - local density, 5-5
- page generation, 5-5
- page setup, 5-5
- text lines per box

correcting, 5-7

F

FDL

- capacity limits, E-1
- command (compiler), 4-2
- command summary, A-1
- generated messages, B-1

fixed fonts, 2-2

font

- specification command, 2-1, 3-6

FONT command, 3-7, 3-19, 3-23, 3-25

fonts

- definition, 2-2
- fixed, 2-2
- limits, E-1
- orientation
 - inverse, 2-2
- proportional, 2-2
- provided with system 2-3
- size restrictions, 5-4
- specifying, 3-6, 3-19, 3-23, 3-25

forms

- compiler
 - invoking, 4-2, 6-8
- creation process overview, 1-1
- creation, 1-1, 3-1
- data, 2-9
- design ruler, F-1
- elements, 2-5
- layout, 5-1
- origin, 2-4

formats, predefined, 2-8

FORM command, 3-3, A-2

form id

- suggested techniques, 5-14

frames, page, 2-3

FSL file

- creating, 4-1

G

GRAPHIC command, 3-2, 3-29, A-2

graphic image

- command, 3-2

graphics

- placing, 3-29
- positioning, 2-6

grid

- command, 3-2, 3-5, 3-11, A-1, A-3
- coordinates, 2-7
- positioning form elements, 2-5

GRID command, 3-2, 3-5, A-1, A-3

- suggested techniques, 5-14
- UNIT value, 2-7

grid units, 2-7, 3-5, 3-15, A-1, A-3

H

hairline, 1-1, 3-12, 3-16, A-4

highlight color, 1-1, 3-29

horizontal line, 2-6, 3-12, 5-1, 5-4, 5-8, 5-13, 6-3, A-4

I

ICATALOG command, 3-7

Images

- placing, GRAPHIC command
- positioning, 2-6, 3-28

INK command, 3-9

inks, substituting, 3-11

invisible

- box, 3-17, A-2
- line, 1-1, 3-12, A-4

IRESULT command, 3-31

L

landscape orientation

- fonts
 - inverse, 2-2
 - page, 2-1

landscape pages

- character density, 5-4
- font size restrictions, 5-4
- restrictions, 5-4

LANDSCAPE/PORTRAIT command, 3-2, 3-4, A-4

- virtual page orientation, 2-4

laser printing technology, 1-1

LINE command, 3-2, 3-13, A-4

- defaults, 3-13

line density

- errors, 5-3

LOGO command, 3-28, A-5

logos

- limits, E-1
- placing, 3-28
- positioning, 2-5
 - command, 3-2

lpi value, 2-7

M

measurements

- rounding, 5-12
 - variable data, 5-13
- using forms design ruler, F-1

messages

- list of FDL-generated, B-1

N

negative coordinates, 2-8

- for placing elements, 2-5

NOPRINT option

- for compiling forms, 4-2, 4-6, A-7

NOSOURCE option

- for compiling forms, 4-2, A-7

O

orientation

- font, 2-2
- landscape, 2-1
- page, 2-1

- specifying, 3-4
- portrait, 2-1
- specification command, 3-4
- origin
 - form, 2-5
 - text block, 2-6
 - virtual page, 2-4
- ORIGIN parameter, 3-5
- overview, FDL, 1-1

P

- page
 - frames, 2-3
 - setup errors
 - error message, 5-5
 - troubleshooting, 5-5
- PAGE SIZE parameter, 3-4, A-4
- PALETTE command, 3-9
- PAPER command, 3-3, A-5
- paper size
 - command, 3-3, A-5
 - default, 2-3, 3-4
 - parameters, 3-3
 - specifying, 3-3, A-5
- paper specifications
 - sizes, 2-4, 3-3
- physical page
 - paper size, 2-4
- points, value, 2-8
- portrait orientation
 - fonts
 - inverse, 2-2
 - page, 2-1
- portrait pages, 5-4
- positioning
 - form elements, 2-5
 - on grids, 2-7
 - text in a box, 3-24
- predefined formats
 - paper size specification, 3-3, A-5
- preprinted forms
 - converting, 5-14
- PROOF option
 - for compiling forms, 4-2, 4-5
- proportional fonts, 2-1

R

- REPEAT command, 3-12, 5-15
 - example, 3-16
- rounding measurements, 5-12
 - variable data
 - alignment problems, 5-13

S

- sample tray
 - destination command, 4-2
- scan line density, 5-3
- SECTION command, 3-31
- sections
 - defining, 3-31

- definition command, 3-2
- ending command, 3-32
- form, 2-5
- form image buffer limitations, 5-11
- invoking, 3-32
- limits, E-1
- line density problems, 5-3
- suggested techniques, 5-3
- secured files
 - overwriting during compilation, 4-5
- semicolon
 - use in commands, 3-1, 3-31
- sequence of coding
 - suggestions, 5-14
- setup commands, 3-2
- shading
 - boundary line density problems, 5-10
 - early termination, 5-11
 - landscape, 5-10
 - portrait, 5-10
 - suggested techniques, 5-3, 5-14
- signatures
 - positioning, 3-25
 - command, 3-25
- simplex
 - specifying for summary sheet, 4-3
- SIMPLEX option
 - for compiling forms, 4-3
- solid line, 3-13
- SPACED parameter, 3-20
- spacing text, 3-20, 5-15
- standard formats, 2-8
- statistics
 - summary, 5-6
- storing forms
 - compiled, 4-4
- summary sheet, 4-3
- syntax ambiguities, 5-16
- system default summary, C-1
- system page, 2-3

T

- tape, magnetic
 - input to the forms compiler, 4-1
- techniques, coding
 - coding sequence, 5-14
 - converting preprinted forms, 5-14
 - new forms, designing, 5-14
- terminating a form description, 3-32
- text
 - positioning in box, 3-25, 6-6
- TEXT AT command, 3-20, 6-5
 - space allocation, E-1
- text coding
 - suggested techniques, 5-14
- TEXT commands, 3-19
 - aligning text, 3-20
 - defaults, 3-20
 - font requirement, 3-20

- spacing, 3-19
- text, fixed
 - positioning, 2-5
- TEXT IN BOX command, 3-24, 6-6
 - adjacent boxes, 3-26
 - defaults, 3-26
 - space allocation, E-1
- TEXT SPACED AT command, 5-19
- tools, F-1
- TRAY option
 - for compiling forms, 4-2
- troubleshooting page setup errors, 5-5

U

- UNIT
 - parameter, 3-5
 - value, 2-8
- USLEGAL option, 3-3
- USLETTER option, 3-3

V

- values
 - converting to dots, 5-12
 - rounding, 5-13
- variable data, 2-9
 - rounding, 5-13,
- vertical line, 3-12
- vertical printing, 3-17
- virtual page
 - orientation, 2-4
 - origin, 2-4
 - specification command, 2-4
 - specifying size, 2-6

X

- x and y coordinates, 2-7
- XPAN option
 - for compiling forms, 4-2