Version 6.0
May 2013

XPAF Forms Generator
User Guide and Reference

Table of Contents

1. Introduction

This manual describes how to use the XPAF Forms Generator (XFG) to create an electronic version of a form. XFG compiles Forms Source Language (FSL) to generate a Xerox LCDS form (.FRM) and store it in the XPAF Centralized Forms Library. XPAF uses this library to manage and download forms to LPS and PCL printers and to PDF documents as required. Utility programs and JCL are provided to access this library and retrieve these FRMs outside of XPAF to allow sampling and/or private forms management.

About this manual

This manual is designed for users who are familiar with the MVS-z/OS operating systems and have experience using Xerox printers. For more information, refer to the "Overview" chapter.

This manual describes XFG commands and statements and how to plan and lay out your forms. Before using this manual, become familiar with its contents and conventions.

Contents

This manual contains the following sections:

Chapter 1   "Introduction" describes the purpose of this manual, contents and conventions.

Chapter 2   "Overview" describes the XFG features, operating environment and hardware and software requirements. It also lists supported Xerox laser printing systems and common forms you can create and manage with XFG.

Chapter 3   "Planning your forms" describes how to plan and design forms. It shows you how to determine page orientation and format, add colors, select paper size and use the design ruler.

Chapter 4   "Using XFG commands" describes the available commands and how to set up your forms. It also contains the command statement conventions and definitions you use to create the FSL and examples of JCL.

Chapter 5   "Creating a sample form" describes how to create a form using XFG.

Chapter 6   "Managing and printing sample forms" describes how to compile, store and download forms to the printer.

Chapter 7   "Messages" lists the error and informational messages that may appear during forms compilation.

Appendix A   "System considerations" describes the printer limitations you may encounter during the creation of complex forms.

Appendix B   "FMT grid format properties" lists the grid and origin substitutions you can use for the supported paper sizes.

Appendix C   "Approved abbreviations for reserved words" lists the abbreviations you can use for reserved words.

Appendix D   "Supported paper sizes" lists the paper sizes that XFG supports.

Conventions

This manual uses the following conventions:

In this manual, the word "character" refers to an "alpha-numeric character," except where noted.

"Supported paper sizes" appendix	References to chapters, sections and appendices appear in quotation marks.
ICATALOG	Commands, operands and parameters that you enter appear in bold.
XPAF Forms Generation Language 5.0 Creating Forms	Document names, library names, variable information and user-defined data you enter appears in italic.
LPI	System commands and parameters appear in uppercase letters.
Note	Notes are hints that help you perform a task or understand the text.

CAUTION: Cautions appear immediately before any action that may destroy the data stored on your network or damage your equipment. Make sure you understand the potential impact of the action before you perform it.

Related publications

You can find additional information related to XFG in the following publications.

Xerox publications

Xerox related publications include the following:

Publication	Number
Xerox LPS Forms Creation Guide	720P90081
Xerox LPS Operations Reference	720P90041
Xerox LPS PDL/DJDE Reference	720P90051
Xerox LPS Print Description Language Quick Reference Card	720P60681
Xerox LPS Standard Font Library User Guide	720P86174

IBM publications

IBM publications related to XFG include the following:

IBM z/OS JCL Reference

IBM z/OS User's Guide

2.   Overview

The XPAF Forms Generator (XFG) operates in a standard IBM z/OS environment that includes Xerox high–speed, high–volume centralized, PCL and/or PDF printers.

This chapter describes the XFG features, prerequisites, operating environment and the supported printers. Refer to the Xerox XPAF User Guide for more information.

XFG features

XFG provides the following features:

- Flexible language statement syntax. The syntax allows you to quickly and easily develop your form electronically. The syntax is similar to the source language that resides on the Xerox laser printing system (LPS) printer and that the Forms Description Language (FDL) compiler uses. XFG also allows additional command extensions, such as dot leaders, which the FDL compiler does not support.

- Additional flexibility. You have the ability to download host–generated forms, which were compiled in different formats, to specific laser printers.

- Increase in several FDL limits.

- Standard entries. You have the ability to enter standard XFG reserved words and identifiers in uppercase or lowercase letters.

Prerequisites

You should have a working knowledge of your MVS or z/OS operating system and be able to perform the following procedures:

- Create and edit a text file on your system.

- Use JCL to compile a job on your system.

- Use JCL to access a file through your resource management interface and download your selected file to your Xerox laser printing system.

Operating environment

The host computer, peripheral devices, printers and software make up your printing system. This section describes the hardware and software requirements to operate XFG.

Hardware requirements

XFG 5.0 operates on IBM S/370 and S/390 series computers running the MVS/ESA or z/OS operating system.

Software requirements

XFG operates in batch mode or interactively under ISPF panels within the IBM MVS/ESA and z/OS operating system. It does not use operating system features specific to any MVS release level. Any specific requirements for your site are provided during installation and described in the XPAF 5.0 Installation Guide.

Xerox laser printer support

XFG supports Xerox centralized and decentralized printers.

Centralized printers

Centralized printers are high–speed, high–volume printers generally used in central locations for printing data processing output from mainframe computers. They may be attached to the host computer by an S/370 channel or attached remotely by a TCP/IP connection.

Decentralized printers

XFG also supports Xerox decentralized printers through XPAF. These printers are generally in remote locations away from the mainframe. A TCP/IP communications link connects them to the host.

Contrasting XFG with a typical forms application

XFG allows you to create electronic versions of your forms, compile them, store them in your XPAF CFORMLIB and download them to your printer as needed. This eliminates the need to stock large quantities of preprinted forms.

Once you download your form to a Xerox LPS printer, the printer merges your form with variable data. This capability allows you to create, store and print any number of forms, in any order, on plain paper.

In a typical forms application that does not use XFG, you have to prepare and load preprinted forms (often printed in large quantities) on a printer connected to a computer. You can load only one set of forms at a time and must remove and replace any forms currently loaded with the preprinted forms you want. The typical forms application usually requires a stock of preprinted forms. If there are different forms, storage costs can be high. If you have to modify preprinted forms, you usually must discard old versions, making your costs even higher.

Forms you can create with XFG

The following list provides examples of general business forms you can create with XFG:

- Bar codes
- Billing statements and invoices
- Calendars
- Checklists
- Financial statements
- Floor plans
- Flow diagrams
- Marketing surveys
- Order forms
- Organizational charts
- Packing and shipping slips
- Production and inventory control forms
- Reorder lists
- Reports
- Routing forms
- Schedules
- Status report forms

Depending on your business, you may need to create forms to meet specific industry standards. XFG allows you to create the following types of specialized forms:

- Bank statements
- Government compliance forms
- Insurance forms
- Tax return forms

3. Planning your forms

This chapter describes the parameters you must determine and specify when designing your form.

Careful planning and design are critical to creating electronic versions of your forms. Form planning and design specify such things as page size, orientation, layout, fonts, inks, and highlight colors.

For additional information about designing complex forms and Xerox laser printer capabilities, refer to the "System considerations" appendix.

Determining format

Before you create a form with XFG command statements, you must plan its layout, content, and appearance according to the requirements described in the following sections.

Page orientation

The orientation of a page is either portrait or landscape. Figure 3–1 shows these orientations. Figure 3-1 **Page Orientation**
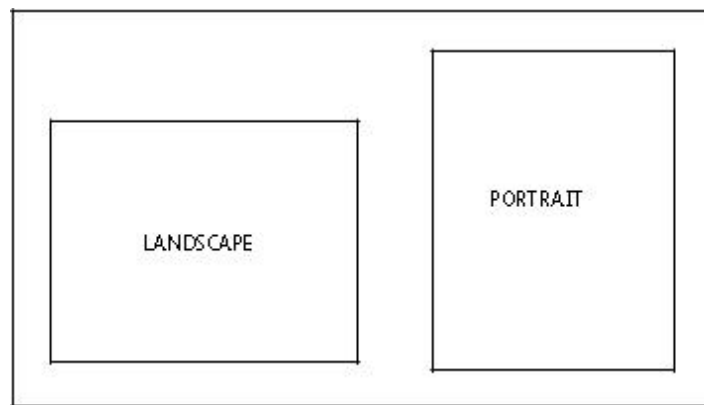
 The landscape orientation, sometimes called the horizontal format, refers to a page whose horizontal side is longer than its vertical side.

The portrait orientation, sometimes called a vertical format, refers to a page whose vertical side is longer than its horizontal side. Use the portrait orientation for text material such as letters, manuals, reports, and forms that need the tall vertical format.

Landscape format is ideal for forms, spreadsheets, and reports that require columns across the page. For example, the landscape orientation is appropriate for a Gantt chart that lists six or twelve months horizontally, and lists the tasks vertically.

Large paper size forms in short edge feed (SEF) require you to specify the **FONTS** command in the opposite orientation. For example, for a portrait orientation, use landscape fonts, and for a landscape orientation, use portrait fonts. Make sure you specify the orientation you want. For example, for a portrait form, specify **PORTRAIT** for the orientation, and for a landscape form, specify **LANDSCAPE**.

Page frames

The page frame is a set of boundaries associated with a page as a unit of printing or imaging. There are three types of page frames: system, physical, and virtual. Figure 3–2 shows the relationship between the system, physical, and virtual page. Figure 3-2 Page Frames

SYSTEM PAGE

PHYSICAL PAGE

VIRTUAL PAGE

Note: Corresponding edges of physical and system pages. This is important when considering edge-marking.
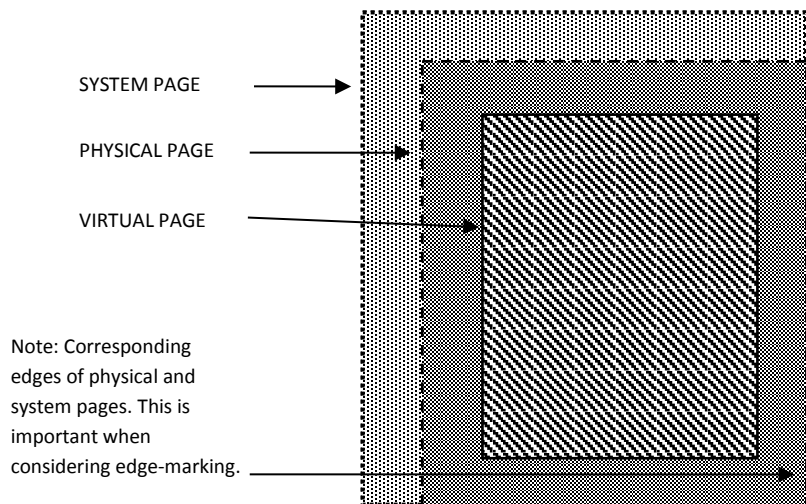
For additional information about page frames, refer to the Xerox LPS Forms Creation Guide.

System page

The system page is the maximum area the printer allows you to image graphics and text. The system page is slightly larger than the physical page. If the default paper sizes have been established for your printer, the system page has little bearing on your form design.

If you decide to place marks along the edge of the physical page (edge–marking) in your form, be familiar with the edge–marking limitations. To accommodate edge–marking, the system page must be larger than the physical page. Since the system page boundaries correspond to at least two edges of the physical page boundaries for all paper sizes, you can only use edge–marking on the two edges of the physical page that does not correspond to the system page.

The product guide for each printer contains specific information about edge–marking.

Physical page

The physical page is the actual paper size. On a Xerox LPS, you define the default size of the physical page during system generation (SYSGEN). During planning, you must consider the default paper size or other paper sizes available for your printer. See the "Supported paper sizes" appendix for more information.

Virtual page

The virtual page is the area you specify for printing. The virtual page can be as large as the physical page (actual paper size), but can never be larger than the physical page. If you do not specify the virtual page size, the defined (planned) area of the virtual page is the same as the physical page.

Form origin

All form data described by XFG is positioned on the page relative to a point, called the form origin. The form origin is a reference point from which you place all elements on a form using XFG commands. The form origin is a point that is offset from the upper left corner of the virtual page. You can specify this point during form description. If you do not define a virtual page and specify a form origin, the default form origin is the origin of the physical page.

Figure 3–3 shows the location of the form origin of the virtual page
in relation to the actual page size.

Figure 3–3. **Form origin location**
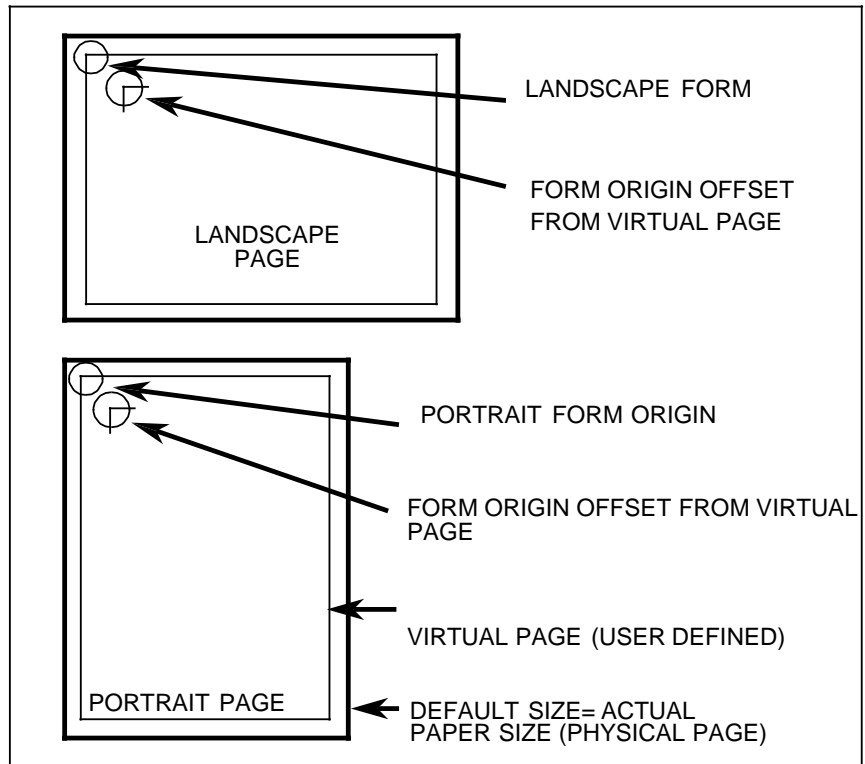
Positioning form elements

You can place the following elements on a form with XFG:

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

You locate an element by determining the horizontal (x) and vertical (y) distance from the upper–left corner (origin) of the form. These coordinates (x, y) comprise the origin of the document. The form origin (upper left corner of the page) is always (0,0).

Form element	Element origin
Vertical ruled line	Top of the line at the midpoint of the thickness of the line.
Horizontal ruled line	Left end of the line at the midpoint of the thickness of the line.
Box (outlined)	Upper–left corner of the box at the midpoint of the thickness of the outline.
Box (shaded)	Upper–left corner of the box.
Fixed text (single line)	Upper–left corner of the first character cell when the line appears upright.
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.
Graphics	Upper–left corner of the image.
Logos	Upper–left corner of the first character cell.
Sections (of a form)	Upper–left corner of the section.

You specify the location of a form element (alphanumeric character, line, or box) on a page in terms of its horizontal and vertical displacement from the form origin. You can use these units of measure to define this displacement:

- Characters per inch (cpi) horizontally
- Dots per inch (300 dpi)
- Linear units (inches or centimeters)
- Lines per inch (lpi) vertically
- Xdots (600 dpi)

X and Y coordinates

If you draw a horizontal and vertical line through the form origin to create x and y coordinates, the location of the form origin is expressed as y=0, x=0. When specifying both coordinates, provide the y value first. Figure 3–4 shows the correct placement for x and y coordinates.

Figure 3–4.   **X and Y coordinates**
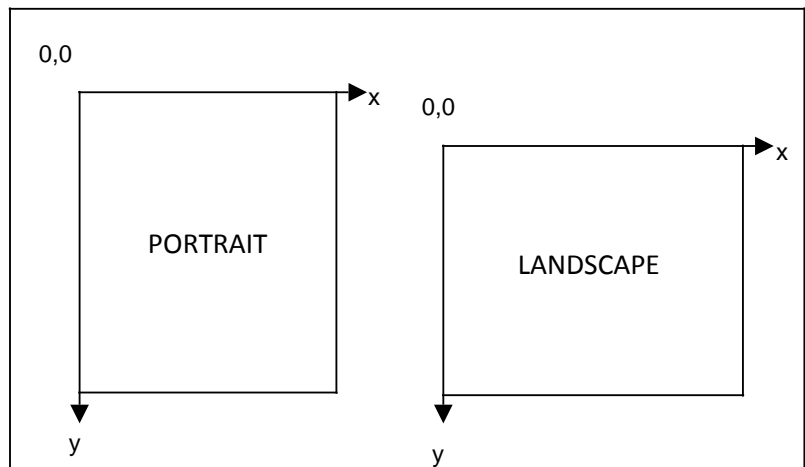
If you draw additional lines horizontally and vertically away from the form origin so that each additional line is one unit of measure apart, you create a grid. You can now position form elements by specifying a grid position.

Negative coordinates

You can use negative x and y values to place the form elements above and to the left of the form origin. You can only specify negative values within the system page boundaries.

Unit value

You can specify the unit of measure you want to use to position form elements on the page. However, predefined formats automatically select the grid values for you. The "FMT grid format properties" appendix lists the predefined grid format properties for various page sizes.

Data types

There are two types of data: forms data and variable data. Make sure you consider them when you create forms for printing on Xerox LPS printers.

Forms data

Forms data is the information on a form that remains the same, such as form titles and column titles. Typically, fixed–length forms data changes only when you modify the form. Forms data is the only data you enter in the design process.

When designing a column size you must consider the amount of variable data the column accommodates, not the size of the fixed–length column title.

Variable data

Variable data is the data you overlay in the space provided by the designed form. Variable data varies in length. An example of variable data is a name and address.

You must allow sufficient space in the columns to accommodate the variable data. Otherwise, variable data may be invalid or unusable. For example, if you do not provide sufficient space for an address, the zip code could be truncated. Spacing for variable data is extremely important and is more accurately calculated with fixed spacing of the font characters.

Fonts

Fonts are character sets, each having a unique typeface (Times, Roman or Helvetica), type size (points), and type style (normal, bold, or italic).

Fonts must reside in the host database and on the printer. Check with your System Administrator to identify the fonts available on your printer. If you specify a font that does not reside in the host database, you receive an error message when you compile the form.

Fixed and proportionally-spaced fonts

You can use fixed and proportionally spaced fonts on Xerox LPS Printers. Each character occupies an area of the form cell 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.

Usually, you use fixed fonts for variable data and proportional spacing for forms data. Figure 3–5 shows examples of fixed and proportional spacing.

Figure 3–5.   **Character spacing**
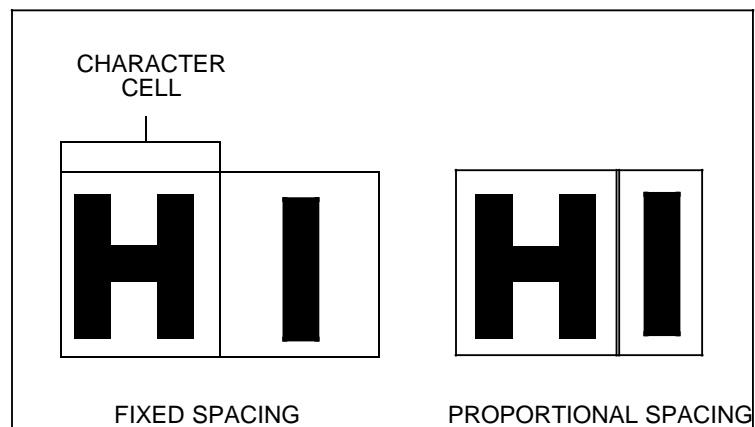
Font orientation

In addition to typeface, style, and size, you can also define fonts by their orientation:

- Landscape
- Portrait
- Inverse landscape
- Inverse portrait

Figure 3–6 shows the font orientations relative to the physical page.

Figure 3–6. **Font orientation**
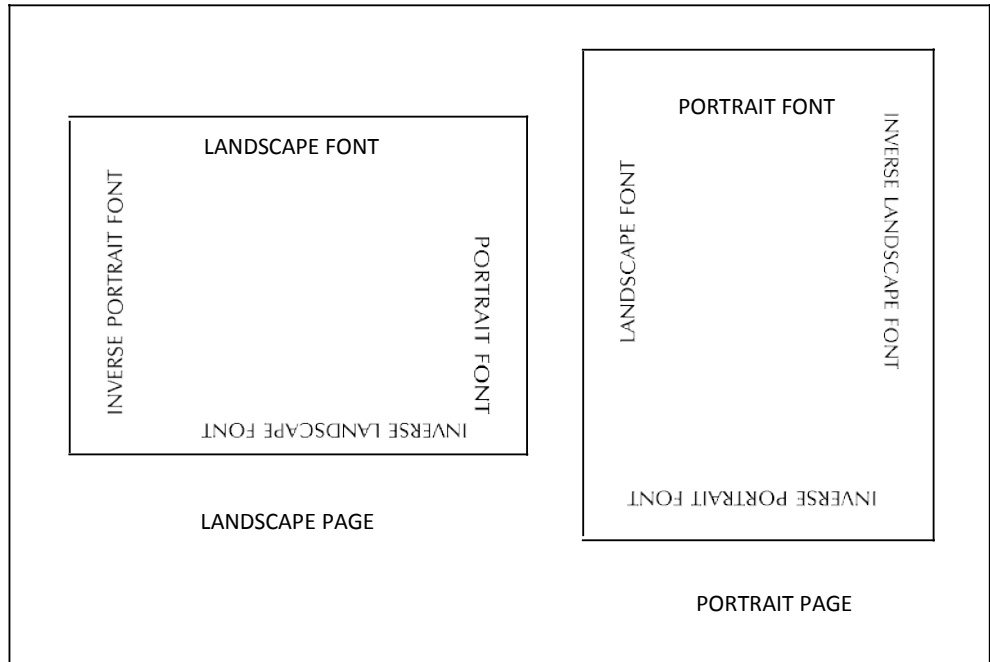
Font orientation when using short–edge paper feed

Normally, stock feeds into the printer moving in the direction of the page length. You assign each job a font with a landscape orientation (characters align along the long edge of the page) or a font with a portrait orientation (characters align along the short edge of the page).

Some printers, such as the Xerox 4135 LPS, support paper sizes up to 14.33 by 17 inches by allowing the stock to feed into the printer moving in the direction of the page width. This is the short edge feed (SEF) option. This reversal of the feed edge reverses the font orientation the job must use. When printing a landscape document using the SEF option, specify portrait fonts. When printing a portrait document using the SEF option, specify landscape fonts.

Adding color to forms

The Xerox highlight color LPS printers allow you to use color to emphasize important parts of your form. With a highlight color printer, you can use different shades of a single color to highlight parts of your form. The following examples show typical color enhancements for forms:

- Highlighted titles
- Highlighted columns representing totals or important results
- Lines setting off groups of columns or items or used as a border for your form
- Boxes filled with any of the colors and tints available in the ink catalog

You use the XFG command statements to control color. For more information, refer to the "Using XFG commands" chapter.

On a highlight color printer, text, lines, and box borders always print in the primary color of the printer, even if you specify a different color, tint, and so on.

Ink color availability and control

When planning colors, you should determine if your company has any regulations and controls for using colors on forms. You may need to use a specific ink list, which contains an ink catalog, palette, and ink for items such as logos and form titles. Contact your System Administrator to determine the available ink colors.

For additional information regarding default ink palettes, catalogs, color models, ink priority and substitutions, and modifying output, refer to the Xerox LPS PDL/DJDE Reference Manual.

Choosing paper size

Once you determine the actual page size (physical page) available for your printer, you can specify any size within the physical page size for your form.

Contact your System Administrator to determine available paper sizes. For a list of paper sizes for the family of Xerox laser printers, refer to the "Supported paper sizes" appendix.

4. Using XFG commands

You use the XPAF Forms Generator (XFG) keyword commands to translate your form design into a format for the host computer. The command set representing your form becomes the Forms Source Language (FSL), which the host compiles and stores. You can download the compiled form to a Xerox LPS printer. This chapter describes the XFG commands and their functions.

Creating forms with XFG commands

The XFG command language syntax is usually the same as the Forms Description Language (FDL) command language syntax. You may be familiar with the FDL commands and how to use them at the printer.

For more information on FDL commands and printer usage, refer to the Xerox Laser Printing Systems Forms Creation Guide.

Available specifications

The XFG commands allow you to specify the following features:
- **Form name** identifies the forms you are developing.

- **Page orientation** specifies a page as portrait or landscape.

- **Font selection** specifies the fonts you want to use.

- **Vertical and horizontal grid positions** specify grid position and position form elements in line printer measurements (line number and character position) or inches, centimeters, dots and xdots.

- **Highlight color** enhances graphs, charts, forms and other publications.

- **Shading** mixes color with black to produce shades of a primary color (red, green, or blue).

- **Caption position** specifies locations of captions on a page or selects automatic placement within a specified area.

- **Logo and graphic position** specifies the location of selected logos and graphics on a page.

Command format

Each XFG command consists of a command identifier, parameters   and reserved words associated with the function.

The following conventions apply:

- A semicolon (;) terminates a command.

- A comment line begins with the word COMMENT, followed by a comment and semicolon (;).

- A blank space separates reserved words and parameters.

- Commas are legal only as shown in the command syntax flow charts and may be optional.

Command information appears within the first 72 columns of each record. Commands can continue for multiple records; however, there is a limit of 4,096 characters allowed in one text block. Multiple commands can appear in one record if you separate them with semicolons. The system ignores records containing all blanks. You can use them to separate command statements.

Reserved words

Reserved words appear in bold uppercase text in the command syntax flow charts in this chapter. You can abbreviate many reserved words to the first three characters (except FMTn, where you must specify the integer n). Refer to the "Approved abbreviations for reserved words" appendix for the list of abbreviations you can use. Avoid using reserved words as section or form names.

Special characters

Use the following special characters to meet certain requirements within the command statements:

- A single quote (') encloses a text string, for example, 'This is a font count' produces "This is a font count" in the printed version. When you use a reserved word such as "font" in a string, it is not processed as a command.

- A pound symbol (#) within a text string shifts the text between uppercase and lowercase characters. Each string begins with an uppercase character. For example, F#IRST produces "First" in the printed version. To shift back to uppercase characters, you must enter another pound symbol (#) character in the text string. For example, R#EV#A produces "RevA" in the printed version.

- An exclamation point (!) specifies a character in the font character set, not the keyboard character. Enter an exclamation point (!) and the ASCII hexadecimal code point for the character you want to print. For example, 65!7B produces 65°. No mapping or translation is done to these characters.

- Special characters (' # !) print only if you type them twice in succession. For example, 'IT''S HERE!!' produces "IT'S HERE!" in the printed version.

Understanding the command syntax flow charts

Many XFG statements offer several options. This chapter represents XFG command syntax in the form of flow charts, which allow you to visualize the main command statement and the way options branch from that statement.

If you want to enter a command in its short form, without options, follow the main command line in the flow chart until you reach a semicolon. Otherwise, follow the branching below the main command line to include any parameters you want.

The main command line contains the required words and operands. Under the command line, or under a branching line, you can use optional words to clarify the command. Parameters that are vertically stacked (one above the other) under the main command line or under a branching line indicate choices for that particular parameter. Ellipses (...) indicate multiple occurrences of the preceding parameter.

Words, characters, or phrases shown in lowercase are variables. Replace these variables with names or values that you have selected.

Setting up your form

Before you enter commands to describe a form, you enter setup commands to specify the following:

- **FORM** specifies the name and resolution of the form.

- **PAPER** specifies the paper size.

- **LANDSCAPE/PORTRAIT** specifies page orientation and virtual page size.

- **GRID** specifies grid unit and form origin.

- **FONTS** specifies fonts to be used in form.

- **ICATALOG** and **PALETTE** specifies ink catalog and palette.

- **INK** specifies inks and colors valid only when using Xerox color printers.

- **IRESULT** specifies results when objects imaged with different inks overlap on Xerox highlight color printers only.

The name parameter of the **FORM** command is required. The other commands are optional. If you do not specify the optional commands, the defaults are used.
If you want to include text on the form, you must specify the **FONT** command.

FORM command

You must first specify the **FORM** command when you describe your form. You must also completely contain it in the first record of the form description.

Enter a form name between one and six characters. This is the only required parameter in the **FORM** command. **FORM** must be the first command of the form file and must precede any comments. Once the form is compiled, this name identifies the form stored on the printer.

Figure 4–1 shows the FORM command syntax flow.

Figure 4–1. **FORM command syntax**
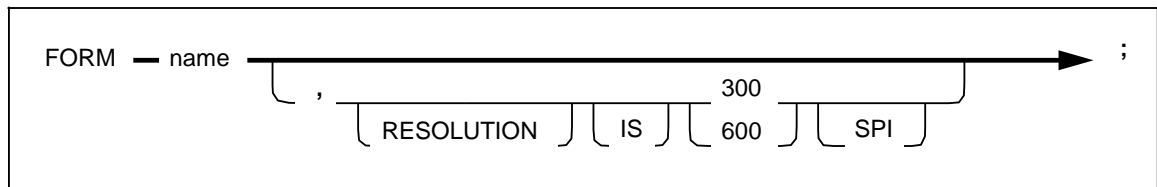
Parameters: Name identifies the name you want to use for the form. The form name must be from one to six characters. Name is a required parameter, but you can omit the remainder of the FORM command.

RESOLUTION: You can specify the resolution (300 or 600 SPI) or leave the resolution blank. IS and SPI are optional. The 600 SPI resolution is only valid on the Xerox 4650 LPS.

Default: 300 SPI if you do not specify SPI. There are no defaults for the form name.

Examples
Long form:
FORM NEWFOR, RESOLUTION IS 300 SPI; Short form:
FORM NEWFOR;

PAPER command

Use the **PAPER** command to specify the paper size for your form.

If you specified a predefined form using the **GRID** command, the paper size is determined automatically and you do not need to use the **PAPER** command.

If you use the **PAPER** command to specify a paper size at run time, make sure the paper size matches the size specified in the JDE or JDL for the variable data.

Before printing your form, verify that the printer contains the correct paper size. Use the printer status key on the printer console. An error message displays if a paper size mismatch occurs at run time. The job aborts and your form does not print.

Refer to the output statement **PAPER SIZE** command in the Xerox LPS PDL/DJDE Reference manual.

Figure 4–2 shows the command syntax flow.

Figure 4–2   **PAPER command syntax**
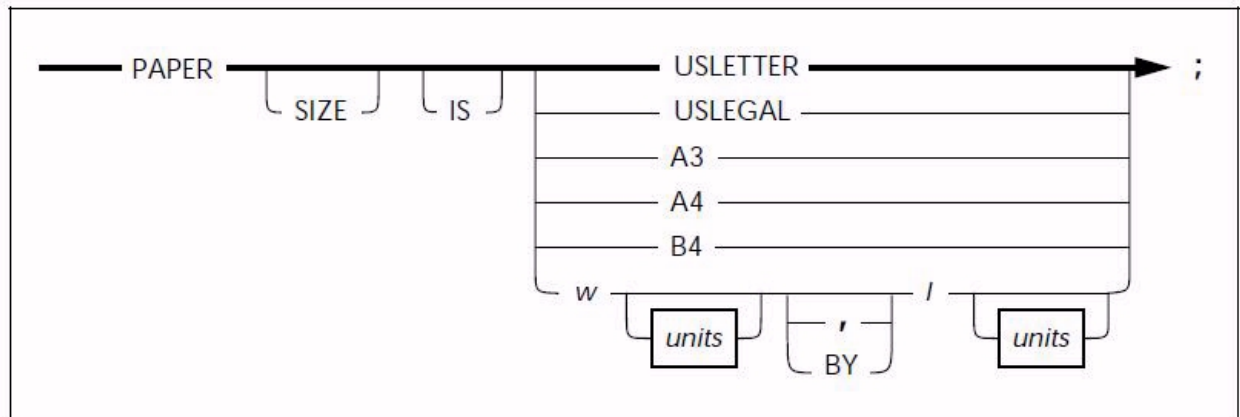
Parameters **SIZE** and **IS** are optional words you can use for clarity in the long form of the language coding.

USLETTER, **USLEGAL**, **A3**, **A4**, and **B4** are the standard paper sizes you can use with the Xerox LPS.

Width (w) and length (l) represent the decimal paper size for printers that allow you to print forms in nonstandard paper sizes.

units indicates the unit of measure you want to use when specifying the size of the page. The units you can choose are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

Refer to the "Supported paper sizes" appendix for more information.

Default **INCHES**, if you do not specify units.

If you do not specify the **PAPER** command and a predefined format is not defined in the **GRID** command, XFG assumes **USLETTER**.

Examples Long form:

PAPER SIZE IS USLETTER;

PAPER SIZE IS 8.5 INCHES BY 11 INCHES;

Short form:

PAPER USLETTER;

LANDSCAPE/PORTRAIT command

Use the **LANDSCAPE/PORTRAIT** command to specify the orientation of the form, and the size and origin of the virtual page.

If you specified the predefined format in the **GRID** command, the form orientation is established automatically and you do not need to use the **LANDSCAPE/PORTRAIT** command.

Figure 4–3 shows the command syntax flow.

Figure 4–3. **LANDSCAPE/PORTRAIT command syntax**

```
   ┌─ LANDSCAPE ─┐                                                        ;
───┤             ├──────────────────────────────────────────────────→
   └─ PORTRAIT ──┘   ┌─ PAGE ──┬──────┬──┬────┬── n ──┬────────┐
                               └ SIZE ┘  └ IS ┘       └ units ┘

     ┌─────────────────────────────────────────────┐
     │          ┌──────┐  n  ┌────────┐  ┌──────┐   │
     └──┬──────┬┴─ , ──┴─────┴ units ──┴──┴ HIGH ┘──┘
        └ WIDE ┘  └ BY ┘
```
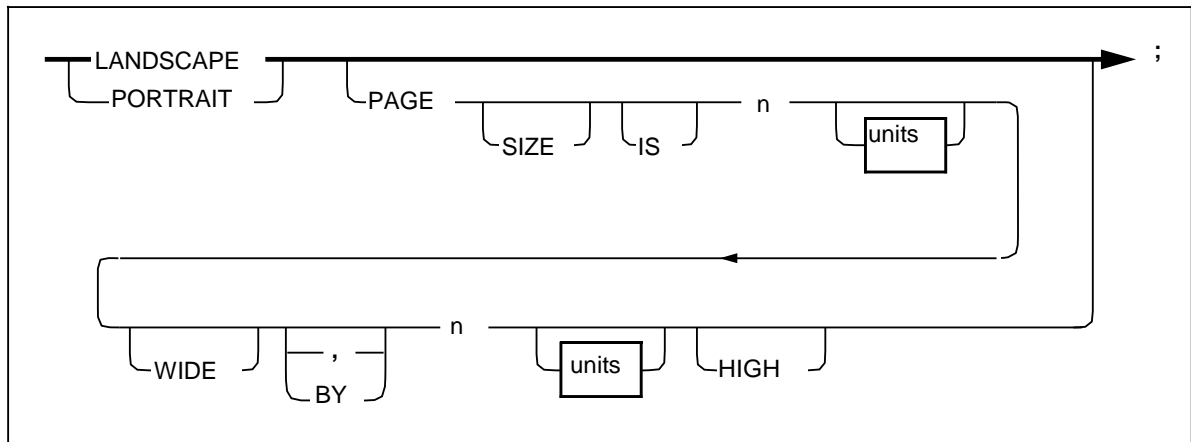
Parameters

LANDSCAPE specifies the landscape form orientation. If you only specify **LANDSCAPE**, grid units and form origin correspond to the standard grid format (FMT1).

PORTRAIT specifies the portrait form orientation. If you only specify **PORTRAIT**, grid units and form origin correspond to the standard grid format for the portrait page (FMT6), unless you change them with the **GRID** command.

Refer to the "FMT grid format properties" appendix for more information.

PAGE, **SIZE**, and **IS** specifies the width and height of the virtual page.

The value you use to specify the special page size is n. You must use positive numbers and you cannot exceed the size specified in the **PAPER SIZE** command. You can use two decimal places when you specify inches. You must express **DOTS** and **XDOTS** as integers.

units indicates the unit of measure you want to use when specifying the size of the virtual page. The units you can choose are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** as integers.

Defaults

If you do not use the **LANDSCAPE/PORTRAIT** command, XFG uses these defaults:

- **LANDSCAPE** (orientation)
- FMT1 (grid units and form origin)
- 8.5 by 11 inches (virtual page)
- **INCHES** (if you do not specify units)

Examples

Long form:

PORTRAIT PAGE SIZE IS 8.5 INCHES WIDE BY 11 INCHES HIGH;

Short form:

PORTRAIT;

It allows XFG to default to the FMT6 predefined format.

Use the **GRID** command to specify the dimensions of the grid unit and the location of the form origin with respect to the virtual page origin. All XFG description commands create and place form elements (lines and boxes) relative to the form origin.

Figure 4–4 shows the command syntax flow.

Figure 4–4.   **GRID command syntax**

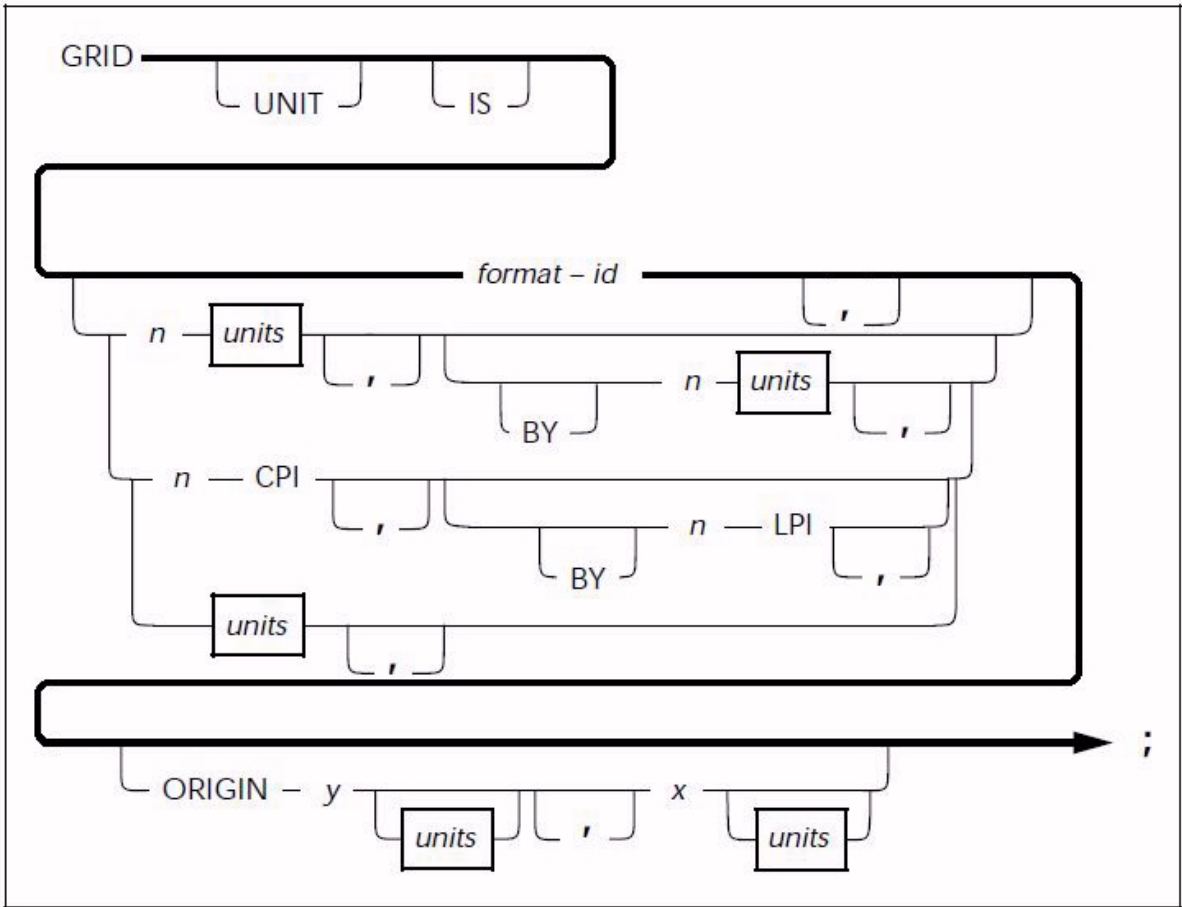|	
Parameters	**UNIT** and **IS** are optional words you can use as a lead–in to the format – id and units identifiers.
	The name of the predefined format that you have selected is format – id. Refer to the "FMT grid format properties" appendix for more information.
	The size of the grid unit is n, expressed in positive numbers of units (for example, 6 inches). If you specify **inches** or **centimeters**, the value represents one side of a square grid unit.
	If you only specify one **DOTS** or **XDOTS** parameter, the grid unit is the same number of dots both horizontally and vertically.
	CPI specifies the number of characters per inch for the horizontal size of the grid unit.
	LPI specifies the number of lines per inch for the vertical size of the grid unit.
	ORIGIN specifies the origin of the form relative to the virtual page origin. **ORIGIN** establishes the 0.0 X and Y coordinates.
	The **GRID** sizes must fall into the boundaries of the physical page. The smallest grid allowed is 1 **XDOT** by 1 **XDOT**. The largest grid allowed is 1 physical page by 1 physical page.
	Y specifies the form origin offset downward from the top–left corner of the virtual page.
	X specifies the form origin offset to the right of the left edge of the virtual page.
	units indicates the unit of measure you want when specifying the offset origin. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.
Defaults	Predefined format FMT1, if you do not specify the **GRID** or **LANDSCAPE/PORTRAIT** commands.
	Predefined format FMT6, if you specify **PORTRAIT** (**LANDSCAPE/PORTRAIT** command), but not **GRID**. If you specify a predefined format, all **GRID** format parameters are automatically provided, unless you change them with the **ORIGIN** parameters.
	INCHES, if you do not specify units.
	A value of 1, if you omit an **n** value when you specify **DOTS** or **XDOTS**.
Examples	Long form:
	GRID UNIT IS 12 CPI BY 8 LPI ORIGIN 0.18 INCH 0.66 INCH;
	Short form:
	GRID FMT1;

You cannot specify any text in your form until you specify the fonts you are going to use with the **FONT** command. Check with your System Administrator for the fonts available for your forms.

Figure 4–5 shows the command syntax flow.

Figure 4–5. **FONT command syntax**
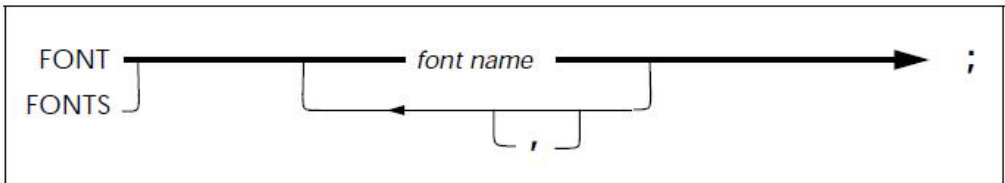
Parameters   Font name identifies the standard or custom font for your form. Font name is a maximum 6-character string. Use a blank space or a comma to separate multiple font names. There is a maximum of 32 fonts per form.

When you compile a form in LPS format, font name is converted to uppercase.

Defaults   None.

Examples   Long form:

FONT Font1;

There is no short form.

Use the **ICATALOG** command to specify the ink catalog to be used on highlight and full color printers.

The ink catalog is a file that contains color information. It resides on highlight color printers.

If you do not specify a catalog for a highlight color printer, the printer uses its default catalog.

Figure 4–6 shows the command syntax flow.

Figure 4–6. **ICATALOG command syntax**
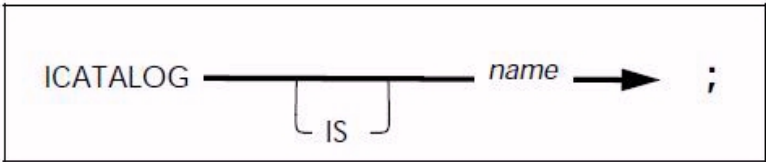
Parameter	name represents the ink catalog to use for color information. Your ink catalog name must not exceed six alphanumeric characters. You cannot use spaces.
Default	Printer default catalog and palette, if you do not specify the **ICATALOG** (highlight color only).
Examples	Long form:
	ICATALOG IS CAT1;
	Short form:
	ICATALOG CAT1;

PALETTE command

Use the **PALETTE** command to specify the ink palette to use for color information on highlight and full color printers. The ink palette is a subset of inks in the ink catalog.

If you do not specify a palette for a highlight color printer, the printer uses its default palette.

Figure 4–7 shows the command syntax flow.

Figure 4–7.   **PALETTE command syntax**
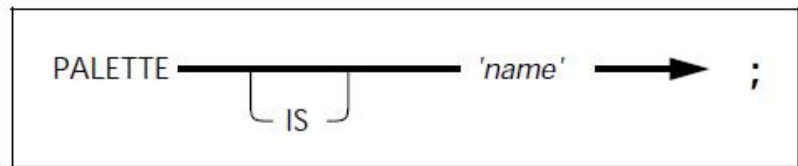
Parameter	name represents the ink palette you use for ink information. Your palette name must not exceed 32 alphanumeric characters, including spaces.
Default	Printer default catalog and palette, if you do not specify the **PALETTE** (highlight color only).
Examples	Long form:
	PALETTE IS 'PAL1';
	Short form:
	PALETTE 'PAL1';

INK command

The **INK** command is only valid on Xerox highlight color and full color printers. This command specifies all of the inks and colors you use when you create a form on Xerox highlight color and full color printers. The inks you specify with the **INK** command are referenced as an index number when you use the **LINE**, **TEXT** and **BOX** commands. They are not referenced as their identification name.

The **ICATALOG** and **PALETTE** commands must precede the **INK** command, as the printer uses the palette and catalog specified in these commands. If you do not specify a catalog or palette for a highlight color form, the printer uses its default catalog and palette.

Ink index numbers begin with 1 for the first ink you specify, 2 for the second, and so forth. This numbered list of ink names is the ink list you use for your form. The ink you selected for a specific description command is identified by the associated ink number found in the ink list. This command follows the **FONT** command. The maximum number of ink names that can be defined is 255.

Figure 4–8 shows the command syntax flow.

Figure 4–8.   **INK command syntax**
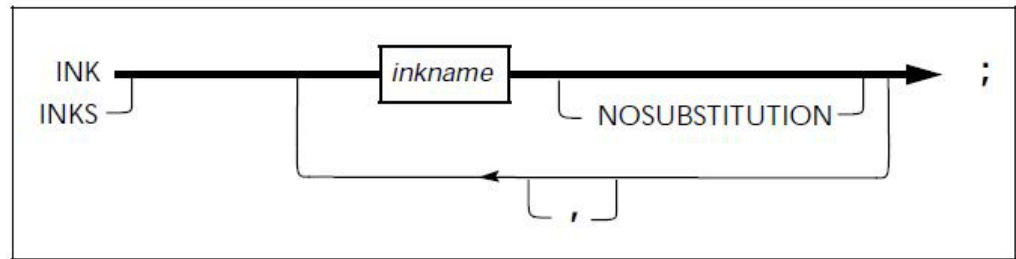
Parameters	Inkname identifies all of the inks you use to create your form. If you use a reserved word such as BLACK as an ink name, you must enclose it in single quotes: 'BLACK'. You must also enclose ink names with embedded blanks in single quotes.
	NOSUBSTITUTION specifies that ink substitutes are not allowed for this ink. If you do not specify this parameter, the printer may attempt to substitute another ink.
Default	The printer default ink, if you do not specify the **INK** (highlight color only).
Example	Long form:
	INKS RED, PINK;
	There is no short form.

Inkname format

This format describes a variety of ways you can specify an ink. Use this statement to describe forms you are compiling for Xerox highlight color or full color printers.

Figure 4–9 shows the inkname syntax.

Figure 4–9.   **Inkname syntax**
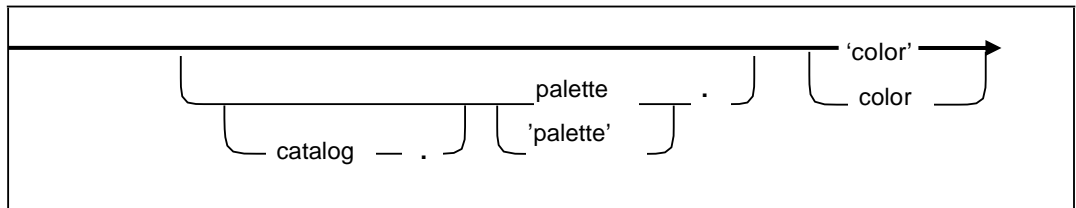
Once you establish an **INK** command parameter (catalog, palette, or ink color), it remains in effect until you change it or another parameter overrides it. A color or palette name containing one or more blank spaces requires quotes around the name.

Defaults   The user catalog specified in the ICATALOG command, if you do not specify a catalog in the INK command.

System default catalog, if you do not specify a default catalog (highlight color only).

The palette you specify in the PALETTE command, if you do not specify a palette in the INK command.

System default palette, if you do not specify a default palette (highlight color only).

When the ink color begins with a digit, you must specify it within quotes.

Examples   This example shows you how to use the single quotes in describing specific inks:

RED or 'RED'

PAL1.'RED', or 'PAL1'.'RED', or PAL1.RED

CAT1.PAL1.'RED', or CAT1.'PAL1'.'RED', or CAT1.'PAL1'.RED, or CAT1.PAL1.RED

The color result you receive from overlapping objects is either black or color for the entire page. The **IRESULT** command allows you to specify the result when objects imaged with different inks overlap. The **IRESULT** command appears in the XFG, FDL, and PDL OUTPUT statements and DJDE. You can use the **IRESULT** command only on a Xerox highlight color printer.

Figure 4–10 shows the command syntax flow.

Figure 4–10.   **IRESULT command syntax**
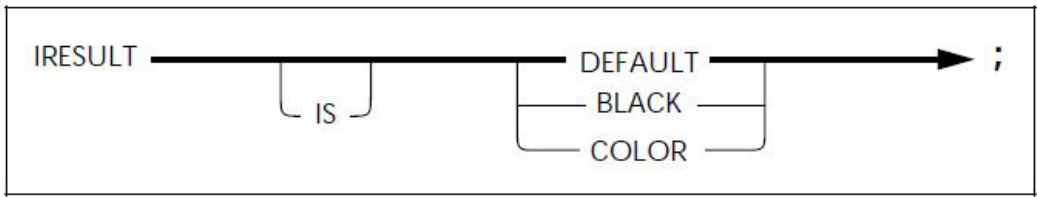
Parameters   **DEFAULT** uses the printer default **IRESULT,** which may be **BLACK** or
COLOR.

BLACK specifies a black pixel to print when you place two objects with different inks that overlap, and black and highlight pixels coincide.

COLOR specifies a color pixel to print when you place two objects with different inks that overlap, and black and highlight pixels coincide.

Default   **DEFAULT**, if you do not enter the command.

Examples   Long form:

IRESULT IS BLACK;

Short form:

IRESULT COLOR;

Description commands

Once you complete the setup commands, you can use the following commands in any sequence to enter descriptive data:

- LINE
- BOX
- TEXT AT
- TEXT IN BOX
- LOGO
- GRAPHIC
- SECTION
- DO SECTION
- COLUMN
- INTERWORD
- DOT LEADER
- LET
- COMMENT
- END

You must place coordinates within the boundaries of the physical page. If coordinates stay within the physical page, but extend beyond the virtual page, XFG issues a warning message. The size of the lines table, currently set at 1,500, determines the maximum number of lines and boxes. Text must fit on the physical page. You can enter up to 4,096 characters in a text block.

LINE command

Use the **LINE** command to specify one or more lines in the same direction on your form.

Figure 4–11 shows the **LINE** command syntax.

Figure 4–11.   **LINE command syntax**
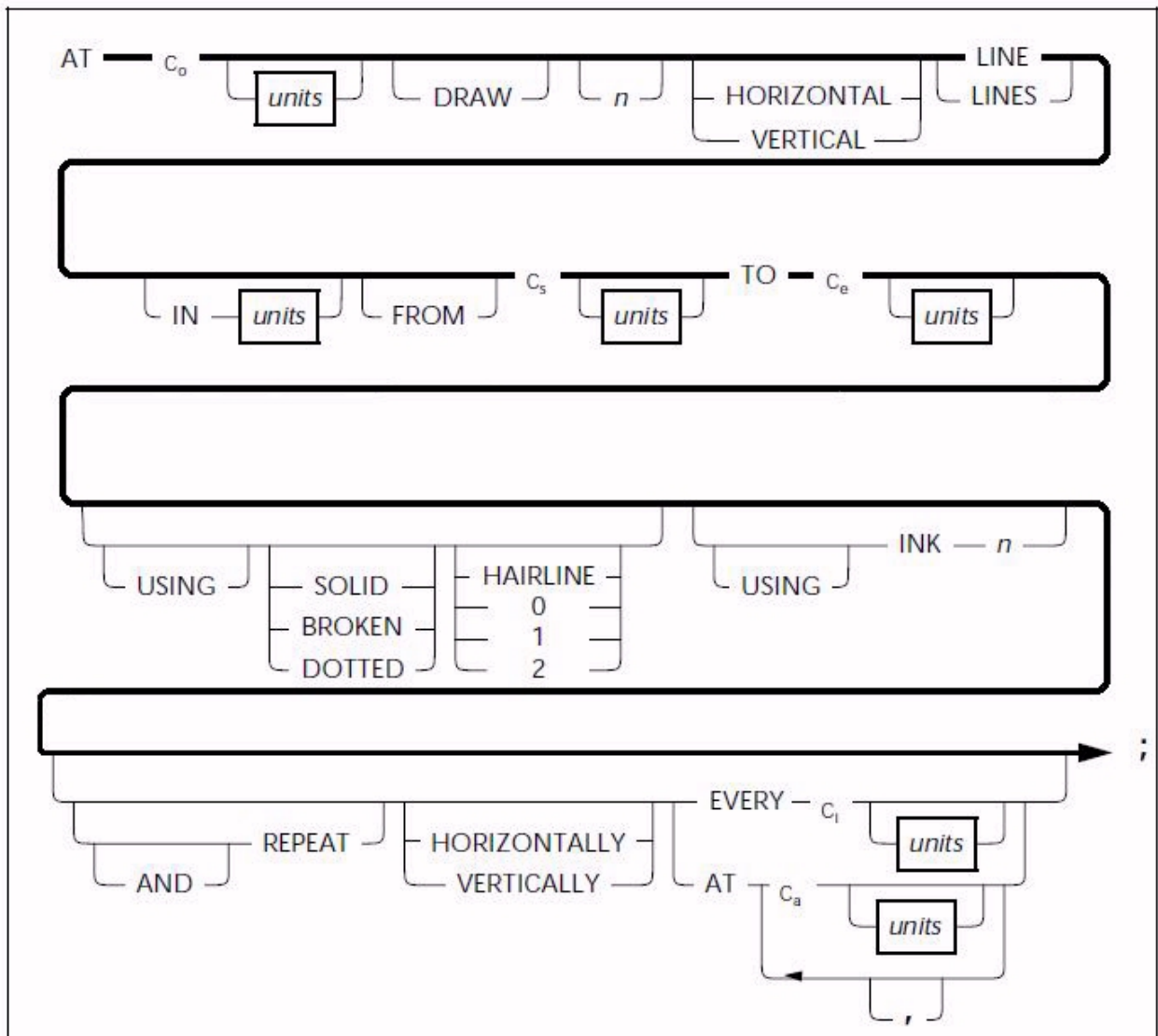
Parameters C_O allows you to specify a line origin location on the form grid, measured from the origin **AT** location, the point at which the line begins. If the line is horizontal, C_O represents the origin in the vertical direction (y–axis coordinate); if the line is vertical, C_O Represents the origin in the horizontal direction (x–axis coordinate).

units indicates the unit of measure for specifying the line origin. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

DRAW is an optional reserved word.

The number of lines you want drawn on your form is n. **HORIZONTAL** or **VERTICAL** specifies the direction of your lines. **LINES** specifies that this reserved word is required.

IN units is a shorthand method of specifying units for the entire command. **IN** units overrides all other units used. For example, if you want to draw a line using inches but specify grid FMT1, **IN** units uses inches as the grid for all units in the specific **LINE** statement.

FROM, **TO** specifies the coordinate at the coordinate start (C_s) point and the coordinate end (C_e) point.

SOLID, **BROKEN**, or **DOTTED** specifies the type of line you want to draw.

HAIRLINE, **0** point (invisible), **1** point, or **2** points specifies the thickness of the line you want to draw.

INK n specifies the ink color index from the ink list for a Xerox highlight or full color LPS printer.

REPEAT specifies that you want a selected line repeated according to the following commands.

HORIZONTALLY or **VERTICALLY** specifies the orientation of the lines you want repeated.

EVERY C_i indicates that you want a line repeated at the increment (C_i) you have specified.

AT C_a indicates that you want the line repeated at the absolute coordinate (C_a) you have specified.

Defaults **HORIZONTAL**, if you do not specify a line orientation.

SOLID, if you do not specify a line type.

1, if you do not specify a line weight.

VERTICALLY, if you do not specify the **REPEAT** orientation.

Examples Long form:

AT 3 INCHES DRAW 5 HORIZONTAL LINES IN INCHES FROM 4 INCHES TO 10 INCHES USING SOLID 1 USING INK 3 AND REPEAT HORIZONTALLY EVERY 0.50 INCHES;

Short form:

AT 3 5 LINES 4 TO 10 REPEAT EVERY 0.50;

BOX command

Use the **BOX** command to describe the size and shape of a box on your form. Similar to lines, you can repeat boxes at fixed intervals or at specified locations. The origin of the box is its upper left corner at the midpoint of the thickness of the outline. If you are using a HighLight Color printer, the **BOX** command allows you to outline the frame of the box in color. Figure 4–12 shows the **BOX** command syntax flow.

Figure 4–12. **BOX command syntax**
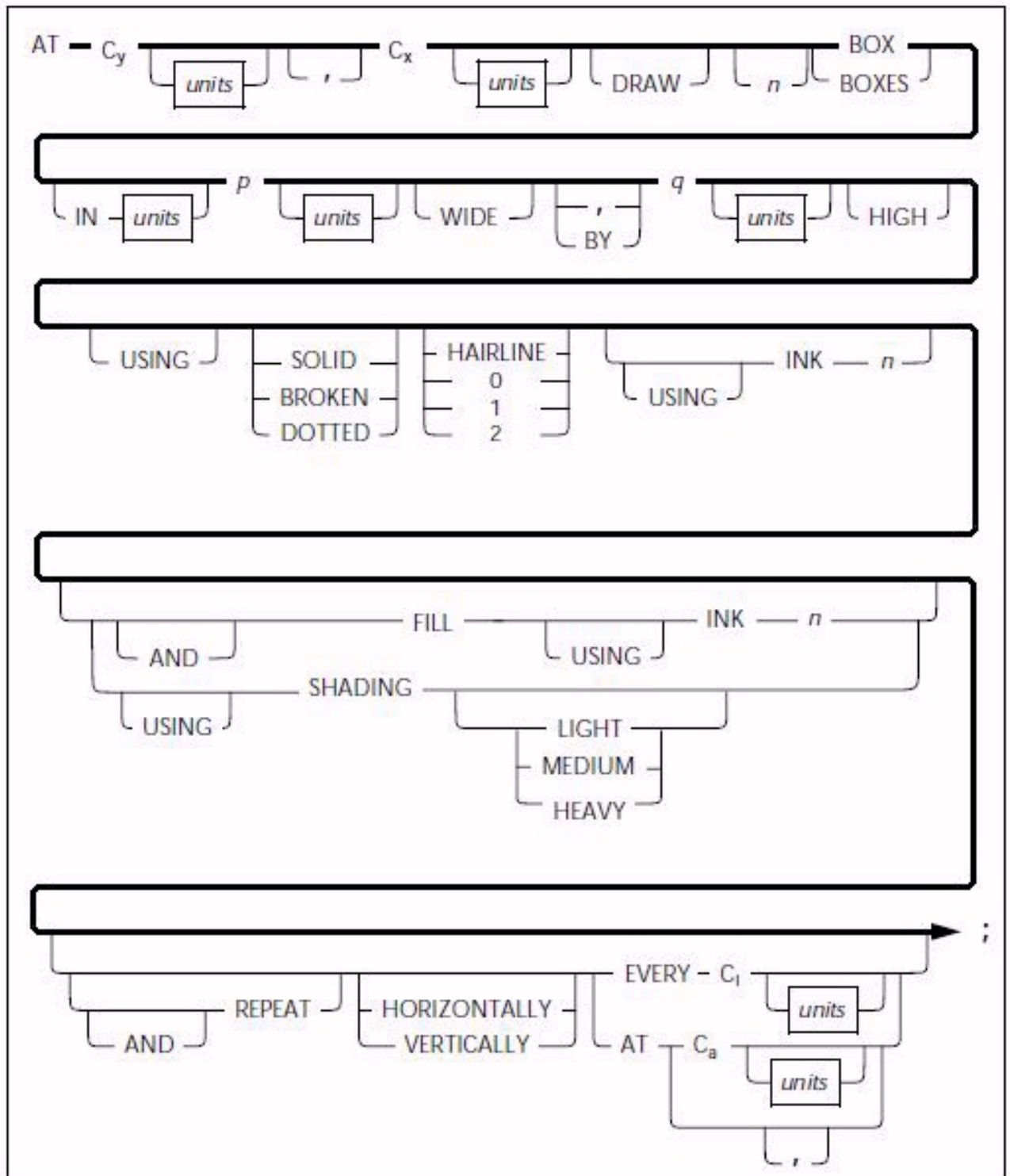
Parameters C_y specifies the y–axis coordinate for the box origin location.

C_x specifies the x–axis coordinate for the box origin location.

units indicates the unit of measure for specifying the box origin. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

The number of boxes you want to draw is n.

IN units is a shorthand method of specifying units for the entire command. **IN** units overrides all other units used. For example, if you want to draw a box using inches but specify grid FMT1, **IN INCHES** uses inches as the grid for all units in the specific **BOX** statement.

The width of the box you have specified is p.

The height of the box you have specified is q.

SOLID, **BROKEN**, or **DOTTED** specifies the type of line.

HAIRLINE, **0** point (invisible), **1** point, or **2** points specifies the thickness of the line for your box.

INK n specifies the **INK** number you selected from the ink list to draw the line in color for use on a Xerox highlight or full color LPS printer. You may specify the same, or different, ink numbers for the outline of the box.

FILL and **INK** n specifies the ink you want to use to fill a box. The box fill can be a color or a tint of color. The ink you specify must be defined in the ink list of the **INK** command.

SHADING specifies shadowing: **LIGHT**, **MEDIUM**, or **HEAVY**.

REPEAT specifies that you want a box repeated according to the following commands.

HORIZONTALLY or **VERTICALLY** specifies the orientation of the repeated boxes.

EVERY C_i indicates that you want a box repeated at the increment (C_i) you have specified.

AT C_a indicates the absolute x or y coordinate where you want the box repeated in a horizontal or vertical direction.

Defaults   **SOLID**, if you do not specify the line type.

1, if you do not specify the line weight or shading.

0, if you do not specify the line weight but do specify shading.

MEDIUM, if you do not specify the degree of **SHADING.**

HORIZONTALLY, if you do not specify the **REPEAT** orientation.

When you specify a box and shade with one command and do not specify a line type for your box, your box will be shaded but you will not have a line around it.

Examples   Long form:

AT 4,6 DRAW 4 BOXES 14 WIDE BY 4 HIGH USING SOLID HAIRLINE INK 2 AND REPEAT EVERY 5;

Short form:

AT 4,6 BOX 14 WIDE BY 4 HIGH;

Use the **TEXT AT** command to incorporate text into your form. To include text in a form, you must specify at least one font. You use the **FONT** setup command to specify the font, and you can only specify one **FONT** command. The fonts you specify with the **FONT** command are referenced as an index number. They are not referenced as their font name. Font index numbers begin with 1 for the first font you specify, 2 for the second, and so on.

Once you specify a font index number in a **TEXT** command, it remains in effect until you specify a new font index. When you use multiple text segments within a **TEXT AT** command and they reference different font indexes, the last font you use remains in effect until you specify a new font index.

Figure 4–13 shows the **TEXT AT** command syntax flow.

Figure 4–13. **TEXT AT command syntax**
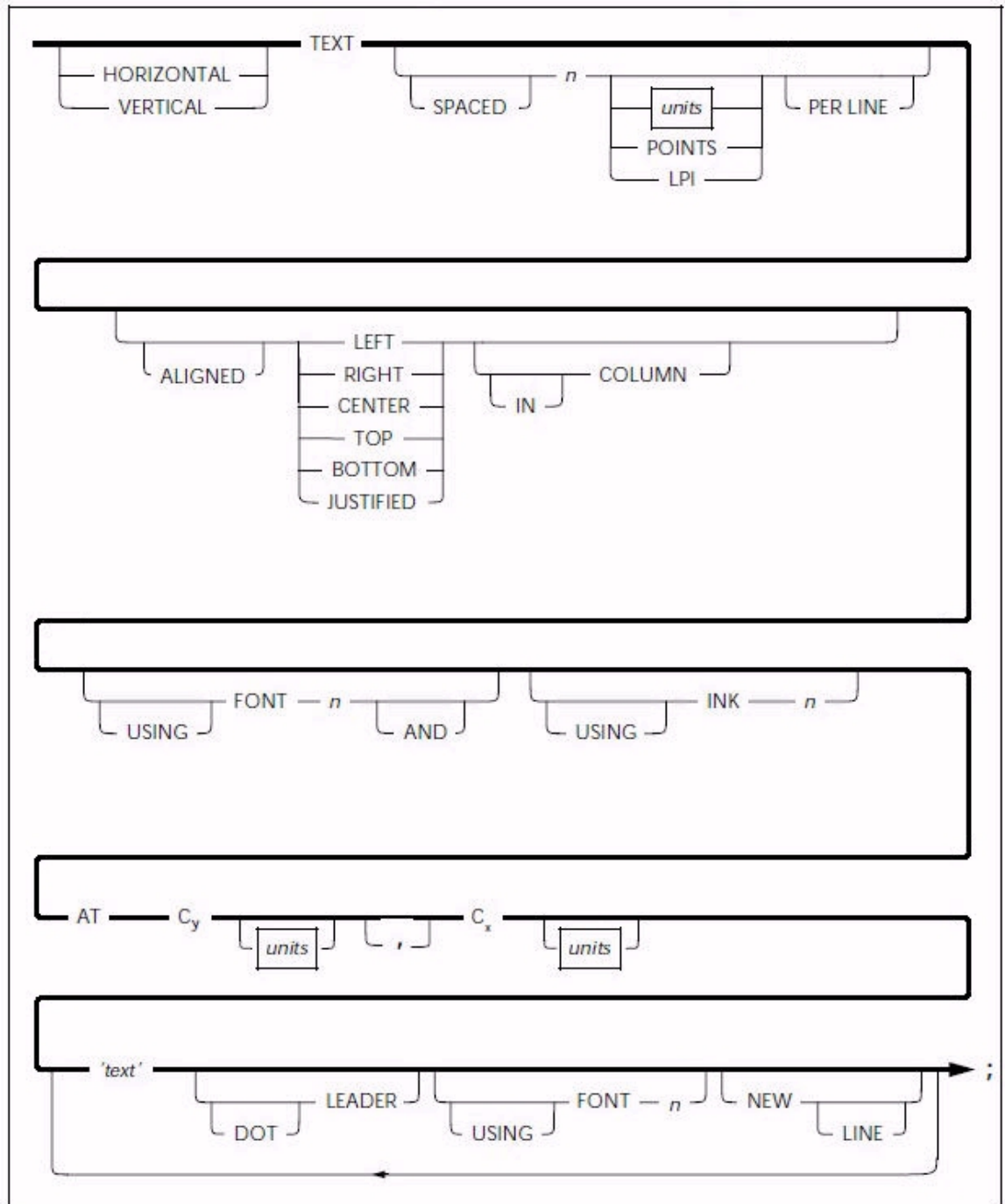
Parameters	**HORIZONTAL** or **VERTICAL** specifies the direction of the text relative to the page orientation.

SPACED defines the amount of vertical space (in n units) a line of text occupies. If you specify a value (not zero), it replaces the line spacing value of the font being used for this **TEXT AT** command. An **LPI** value specifies line height in terms of lines per inch. All other specifications are actual line height measurements. The n units value must be positive. The units are **INCHES, CENTIMETERS, CPI, LPI, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers. **POINTS** specifies that you want the **TEXT SPACED** in **n POINTS** (1/72 inch).

ALIGNED specifies the text justification alignment (**LEFT**, **RIGHT**, **CENTER**, **TOP**, **BOTTOM**, **JUSTIFIED**). See the Xerox Laser Printing Systems Forms Creation Guide for alignment examples. You can center multiple horizontal strings beneath each other or right– or left– justify them. You can center multiple vertical strings alongside each other or align them on the top or bottom.

The **JUSTIFIED** alignment mode allows you to justify text. In simple justification (with no **COLUMN**), you must specify where the lines break. You can justify lines by finding the longest line in a block of lines, then justify other text strings to this length. This method of justification is best for justifying small blocks of text or when you must justify the last line of a paragraph. You can use the **JUSTIFIED** parameter with the **INTERWORD SPACE** and the **COLUMN SIZE IS** commands. In this mode, text begins at the coordinates you specify and extends to the right column boundary. Line breaks occur automatically. Enter text in one continuous string without line breaks.

Example   **JUSTIFIED** parameters:

TEXT JUSTIFIED AT 1,1
    'This is a sample'
    'of justified text'
    'in block format.';

This produces the following text on the form:

This is a sample
of justified text
in block format.

You specify **IN COLUMN** for any alignment mode and use it with the **COLUMN SIZE IS** command. Line breaks occur automatically when you use **TEXT ALIGNED IN COLUMN** or **TEXT JUSTIFIED IN COLUMN**. You can enter a complete paragraph of text in one continuous string. A quote indicates the end of a paragraph rather than a line break. The system ignores multiple line breaks. Columns do not have a fixed location on the page. You place them relative to the **TEXT** command you specify. The horizontal coordinate you specify sets the left column boundary. You set the right column boundary to the sum of the column width and the left column boundary. You can include a maximum of 4096 characters in one continuous text string.

Example   Text justified in a column:

COLUMN 1 INCH WIDE;

TEXT JUSTIFIED IN COLUMN AT 1,1
'four four four five five five six six six'

This produces the following text on the form:

four four   four
five   five   five
six   six   six

FONT n specifies the index number of the font you selected. Refer to the **FONT** command for additional information.

INK n specifies the index number of the ink color you selected from the ink list to use with a Xerox highlight or full color LPS printer. The specified **INK** is in effect for this command and not for subsequent **TEXT** commands.

C_y defines the coordinate you selected on the y–axis where the text origin is located.

C_x defines the coordinate you selected on the x–axis where the text origin is located.

'text' specifies a text string that appears exactly as you enter it (except for special characters).

DOT LEADER specifies dot leaders you have selected to place between text or at a column boundary. Dot leaders extend the full width of the column you specify in the **COLUMN** command, unless interrupted by a **'text'** string. You can place dot leaders between two text strings and a right boundary previously defined by you in the **COLUMN** command. If you place dot leaders between two text strings, the right most text string aligns with the right column boundary.

Do not use **DOT LEADER** in the **JUSTIFIED** mode. **DOT LEADER** overrides any alignment mode and places the text flush with the column boundaries.

USING FONT n specifies a change of font for the next body of text which is not the dot leader string. When you specify dot leaders and a change of fonts together, the **DOT LEADER** keywords must precede the **FONT** n keywords.

NEW LINE specifies a new line to follow a text string. You can only use the **NEW LINE** keyword once between text strings.
NEW LINE is similar to a blank text string (**' '**). If you want to use multiple blank lines between text strings, use multiple blank text strings (**' ' ' ' '**), not **NEW LINE**. **NEW LINE** is ignored if it is not used with **DOT LEADER** or **FONT n**.

Defaults   **HORIZONTAL** and **CENTER**, if you do not specify the orientation and alignment for the **TEXT AT** command.

Use the font index of the last **TEXT** command, if **USING FONT** is not entered. If you omit **FONT** in all **TEXT** commands, FONT1 from the FONT list is used.

DOTS, if you do not specify units.

If a **TEXT** command containing a dot leader is not preceded by a **COLUMN** command, the system uses a default column width of 6.5 inches (portrait) or 9.5 inches (landscape).

Single dots repeated, if you do not use a **DOT LEADER** command to specify the **DOT LEADER** text string.

Examples   The following is an example of a **DOT LEADER** used between two text strings:

TEXT AT 1,1 'John Doe' DOT LEADER '987–6543';

This produces the following text on the form:

John Doe...987–6543

The following is an example of a **DOT LEADER** following a text string and ending at the column boundary:

TEXT AT 1,1 'Sales Tax' DOT LEADER NEW LINE 'Total Now Due' DOT LEADER;

This produces the following text on the form:

Sales Tax ....

Total Now Due ....

The following is an example of the long form of the **TEXT AT** command:

HORIZONTAL TEXT SPACED 12 POINTS PER LINE ALIGNED CENTER IN COLUMN USING FONT 1 AND USING INK 1 AT 1, 1 'Phone List' NEW LINE 'John Doe' DOT LEADER '987–6543';

The following is an example of the short form:

TEXT SPACED 12 POINTS PER LINE ALI CEN INK 1 AT 1,1 'Phone List' NEW LINE 'John Doe' DOT LEA '987–6543';

This produces the following text on the form:

                                    Phone List

John Doe 987–6543

TEXT IN BOX command

Use the **TEXT IN BOX** command to incorporate text into a box on your form. The command syntax format designates the proper type and placement of text in a box according to the planned form. This command format is similar to the **TEXT AT** command, except it includes those parameters necessary to designate the placement of text in a box.

You must define a box with matching coordinate values before you can reference it in a **TEXT IN BOX** command. You can create boxes and adjoining boxes with **BOX** commands, **LINE** commands, or both.

Figure 4–14. **TEXT IN BOX command syntax**
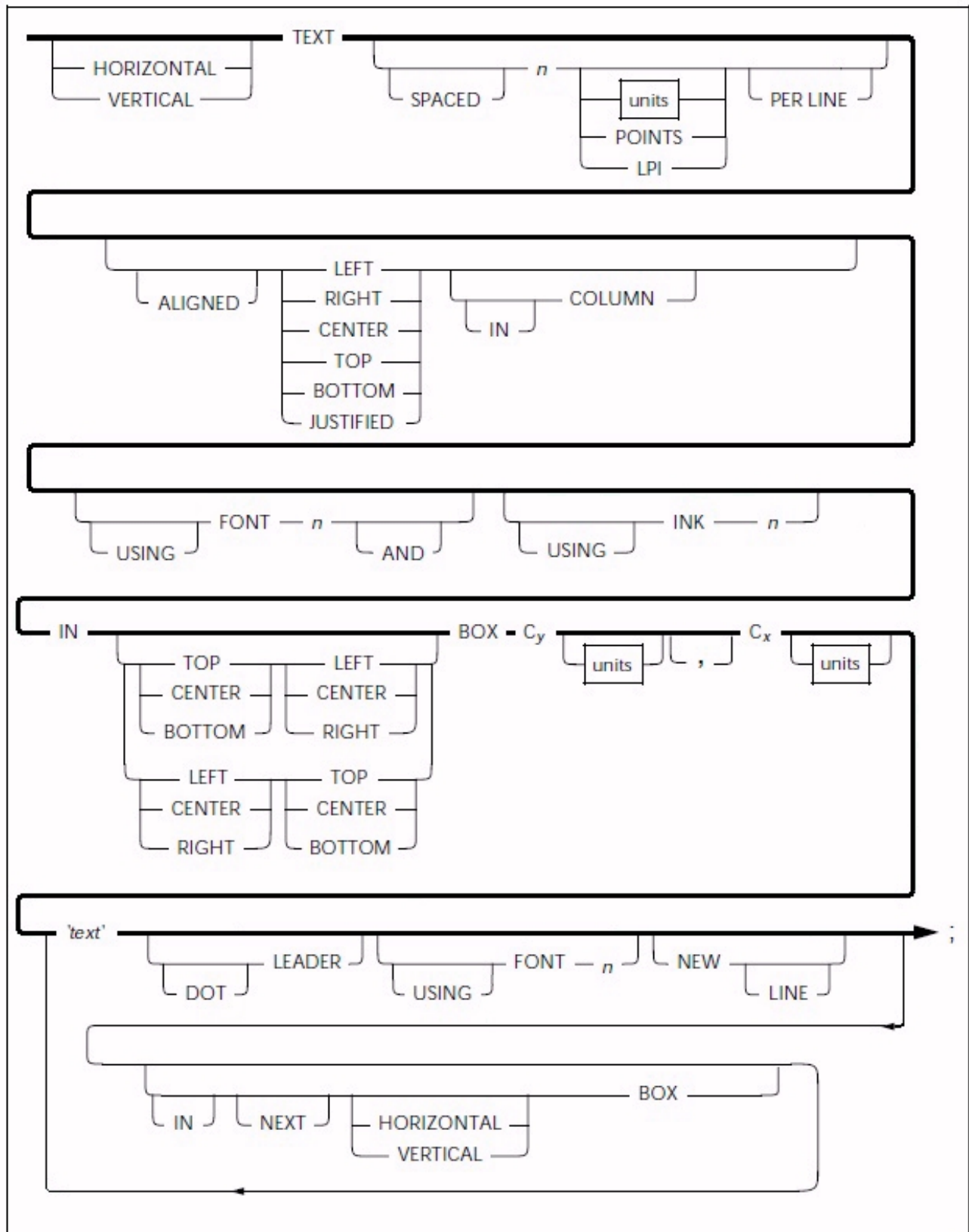
Parameters   **HORIZONTAL** or **VERTICAL** specifies the text orientation of the text within the box.

Do not confuse **TEXT** alignment with **TEXT** placement in the box. You can align horizontal text **LEFT** or **RIGHT**. You can align vertical text **TOP** or **BOTTOM.**

SPACED defines the amount of vertical space (in n units) you want a line of text to occupy. If you specify a value (not 0), it replaces the line spacing value of the font being used for this **TEXT AT** command. All specifications except **LPI** are actual line height measurements. An **LPI** value specifies line height in lines per inch. The n units value you specify must be positive. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You can express **DOTS** and **XDOTS** only in integers. **POINTS** specifies that you want the **TEXT SPACED** in n **POINTS** (1/72 inch).

ALIGNED specifies that you want the text aligned **LEFT**, **RIGHT**, **CENTER**, **TOP**, **BOTTOM**, **JUSTIFIED**. See the Xerox Laser Printing Systems Forms Creation Guide for alignment examples.

The **ALIGNED** and **JUSTIFIED** commands including **IN COLUMN**, are not recommended with the **TEXT IN BOX** command. These are more appropriate with the **TEXT AT** command. If you want to use **JUSTIFIED IN COLUMN,** refer to the "**TEXT AT** command" section for a description of how to use these parameters.

FONT n specifies the index number of the font you selected. Refer to the "**FONT** command" section for additional information.

INK n specifies the index number of the ink color you selected from the ink list to use with a Xerox highlight or full color LPS printer. The specified **INK** is in effect for this command and not for subsequent **TEXT IN BOX** commands.

IN specifies the position in a box where you want to place the text. There are nine positions as follows:

C_y defines the coordinate of the box origin on the y axis.

C_x defines the coordinate of the box origin on the x axis.

'text' specifies a text string that displays on the form exactly as you enter it (except for special characters).

DOT LEADER specifies dot leaders you have selected to place between text or to a column boundary. Dot leaders extend the full width of the column you specify in the **COLUMN** command, unless interrupted by a **'text'** string. You can place dot leaders between two text strings and a right boundary previously defined in the **COLUMN** command. If you place dot leaders between two text strings, the right most text string aligns with the right column boundary.

Do not use **DOT LEADER** in the **JUSTIFIED** mode. **DOT LEADER** overrides any alignment mode and places the text flush with the column boundaries.

USING FONT n specifies a change of font for the next body of text which is not the dot leader string. When you specify dot leaders and a change of fonts together, the **DOT LEADER** keywords must precede the **FONT** n keywords.

NEW LINE specifies a new line to follow a text string. You can only use the **NEW LINE** keyword once between text strings. **NEW LINE** is similar to a blank text string (**' '**). If you want to use multiple blank lines between text strings, use multiple blank text strings (**' ' ' ' '**), not **NEW LINE**. **NEW LINE** is ignored if it is not used with **DOT LEADER** or **FONT** n.

IN NEXT HORIZONTAL or **VERTICAL BOX** specifies the text to place in an adjacent box (horizontally or vertically). Adjacent boxes must have a common, congruent side. For example, **IN NEXT HORIZONTAL BOX...** produces the results shown in Figure 4–15.

Figure 4–15.   **IN NEXT HORIZONTAL BOX**
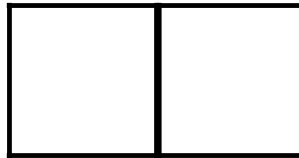
IN NEXT VERTICAL BOX... produces the results shown in Figure 4–16.

Figure 4–16.   **IN NEXT VERTICAL BOX**
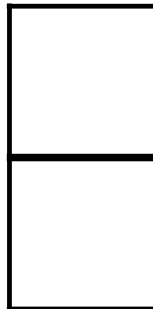
Defaults   **HORIZONTAL**, if you do not specify the box orientation. The same applies for **[IN] [NEXT] [HORIZONTAL/VERTICAL] [BOX...];**

The font index of the last **FONT** command, if **USING FONT** is not entered**.** If you omit **FONT** in all text commands, then **FONT 1** from the **FONT** list is used.

DOTS, if you do not specify units.

CENTER, if you do not specify the alignment. The same applies for the **IN [TOP, ...BOTTOM] BOX....** for text placement within a box.

Examples	Long form:
	HORIZONTAL TEXT SPACED 12 POINTS PER LINE USING FONT 1 AND USING INK 1 IN CENTER CENTER BOX 4 3 'FIRST' 'NATIONAL' 'TRUST' IN NEXT HORIZONTAL BOX 'FIRST' 'DEPOSIT' 'INSURANCE';
	Short form:
	TEXT 12 POINTS INK 1 IN BOX 4 3 'FIRST' 'NATIONAL' 'TRUST' BOX 'FIRST' 'DEPOSIT' 'INSURANCE';

LOGO command

Use the **LOGO** command to place logos in the form. The logo is a special font of one or more characters, which carries with it the relative position of the characters to make up the logo. Although a logo may comprise many characters, it is positioned as a single unit. There is an option that overrides the colors given to a color logo file. For example, if you reference a logo in a color other than blue on a blue form, you must specify **'USING INK** n**'** to override the existing color specification of the logo. If you do not use this option, the printer requests an ink substitution prior to printing the form. Logos you specify in a form you are printing on a Xerox highlight color LPS printer must be in color format. You receive a 'LOGO NOT FOUND' message if you reference a black and white logo in a highlight color (HC) form. The command format designates the proper placement of a logo according to the planned form.

Figure 4–17 shows the **LOGO** command syntax flow.

Figure 4–17. **LOGO command syntax**
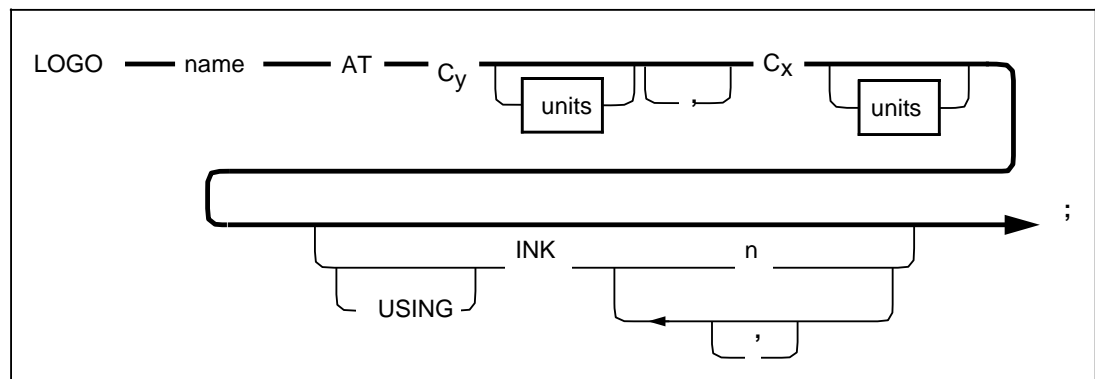
Parameters	name identifies the unique file name (or font identifier) for the logo. A logo name may consist of one to six characters.
	C_y defines the y–axis coordinate for the logo origin location. C_x defines the x–axis coordinate for the logo origin location. units indicates the unit of measure for placement of the logo. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

INK n specifies the ink number from the ink list that overrides the corresponding ink specified in the ink list of the logo file. If the logo has a **NOSUBSTITUTION** flag, then ink overrides are not allowed (highlight color only).

Default   None.

Examples   Long form:

LOGO NEW1 AT 1,1 USING INK 3;

There is no short form.

GRAPHIC command

A form may consist only of graphic images. You can have up to 16 images per page. Xerox highlight color LPS printers allow you to print colored graphics on your form. For highlight color forms, you can also override any ink specified in the ink list of a color graphic file.

You must specify each graphic associated with a form in the forms source language (FSL) file with the **GRAPHIC** command. Figure 4–18 shows the command syntax flow.

Figure 4–18.   **GRAPHIC command syntax**

Parameters   name identifies the IMG graphic file name you specify. This is the name of the graphic stored on the printer.

C_y defines the coordinate on the y–axis to locate the logo origin.

C_x defines the coordinate on the x–axis where you want the logo origin located.

units indicates the unit of measure for placement of the graphic. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

SCALE n specifies the scaling of the printed graphic that you have selected. The scaling factor value n must be an integer. You can only scale on a factor from 1 to 8.

INK n specifies the ink number from the ink list which overrides the corresponding ink list of the graphic (highlight color only).

Default   None.

Examples   Long form:

GRAPHIC SYMBOL AT 1,1 SCALE IS 4 USING INK 3;

There is no short form.

SECTION command

Use the **SECTION** command when a form consists of identical blocks of commands. You can save time by defining one section and then repeating it at the locations you want.

You can also use this command to define an entire form as a section. Within the section, you can use any other command, statement, or specification except another **SECTION** command, **DO SECTION** command, **FORM** command, or **END** command.

Figure 4–19 shows the **SECTION** command syntax flow.

Figure 4–19.   **SECTION command syntax**
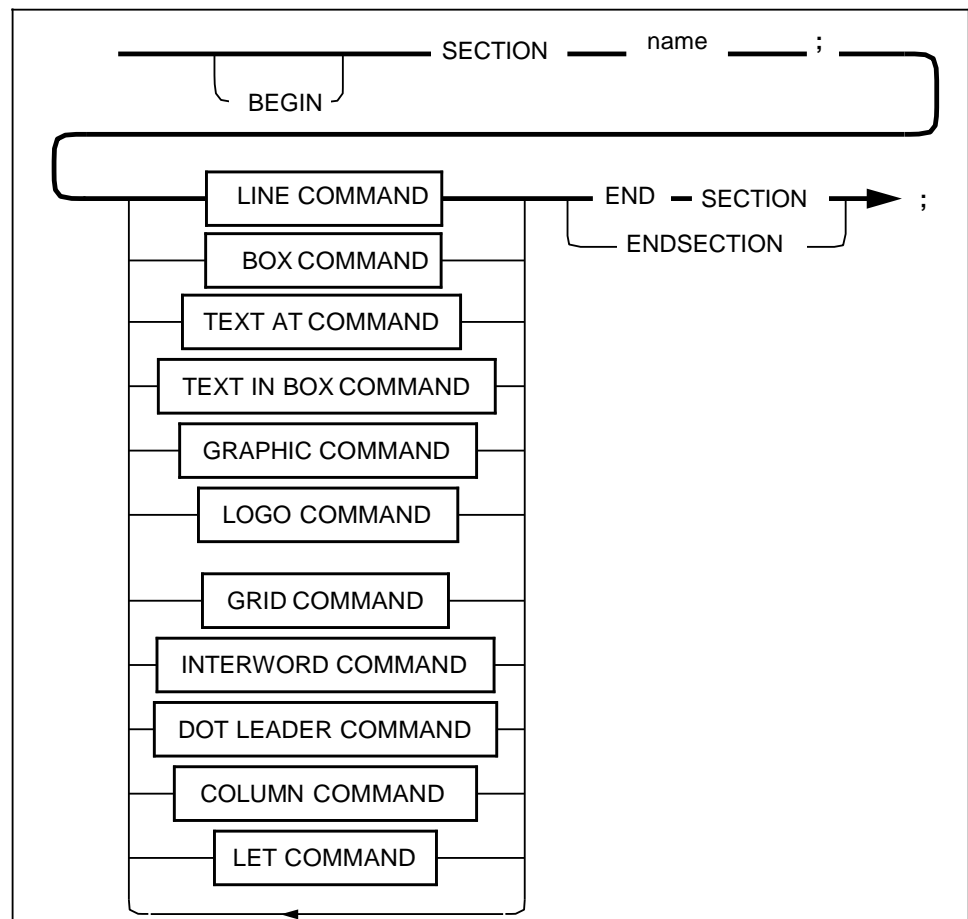
Parameter   name specifies the name of the **SECTION** you want to use to invoke the **SECTION** command. You may define and invoke more than one section, but you must terminate each section with the command **END SECTION** or **ENDSECTION** before you define another section. Do not use a reserved word or keyword as a section name. name is a maximum 16-character string.

Default   None.

	Examples	Long form:

Examples Long form:

BEGIN SECTION NUSEC; LOGO NEW0 AT 0,0 USING INK 2;
TEXT AT 1,1 USING FONT 3 AND INK 1 'Acme Weather Balloons' END
SECTION;

There is no short form.

SECTION definitions The **BEGIN SECTION** command informs the compiler that what
follows includes the **END SECTION** command and lines up to form a
logical collection of commands that may be invoked as needed to
build a complete form.

Most XFG commands may be found within a **SECTION** and these
commands are documented in the **SECTION** command description.
The key point to remember is that coordinates within the **SECTION**
definition are relative to 0, 0. Also, if the **GRID** command is specified,
then the **GRID** is only valid within the bounds of the defining
SECTION. The **GRID** command is a local variable and its scope is that
of the defining **SECTION**. Other statements within the section will
then use the local **GRID** command to determine the placement of
lines, boxes, text, etc. Remember that these placement values are still
relative to 0, 0.

The final placement of the elements is defined by the **DO**
S**ECTION** command.

SECTION invocation A **SECTION** is invoked by using the **DO SECTION** command.
This command names the desired **SECTION** as well as the origin of
the **SECTION**. The **AT** parameter takes on the role of the **ORIGIN**
command for the **SECTION** and is used to determine the final
placement of lines, boxes, text, etc. The final placement is computed
as follows:

$$xFinal = xOrigin + xValue * xGrid$$

and

$$yFinal = yOrigin + yValue * yGrid$$

xValue and **yValue** are taken from the selected command.
xGrid and **yGrid** are taken from the active **GRID** command.
xOrigin and **yOrigin** are taken from the **DO SECTION** command that
invokes the section.

If a **GRID** command has not been defined within the **SECTION**, the **GRID**
that is in effect for the **FORM** is used.

DO SECTION command

Use the **DO SECTION** command to invoke a defined section.
You must first define a section before you can invoke it. You can repeat
sections at absolute or incremental coordinates. You can also repeat a
section in the same location. This command is useful for sections
containing relative coordinates.

Figure 4–20 shows the **DO SECTION** command syntax flow.

Figure 4–20.   **DO SECTION command syntax**
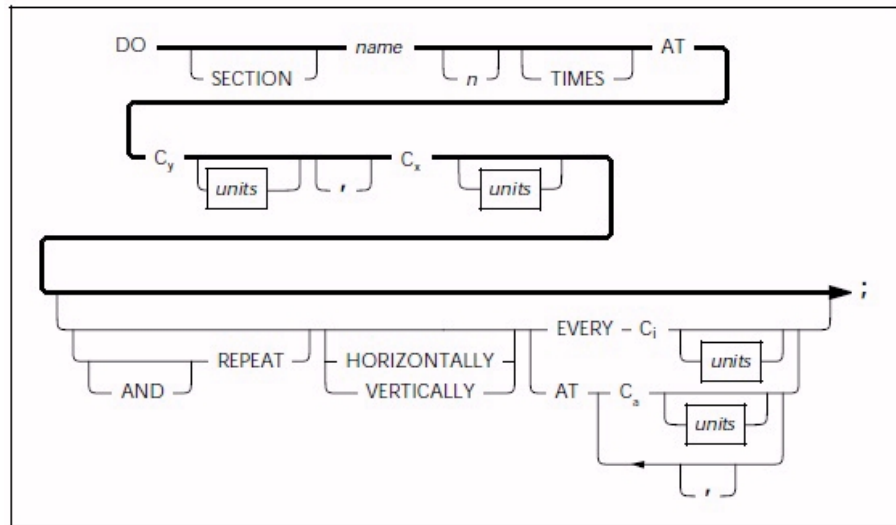
Parameters name specifies the name of the section you want to invoke.

n **TIMES** specifies the number of times you want the section repeated.

C_y specifies the y–axis where you want the section origin located.

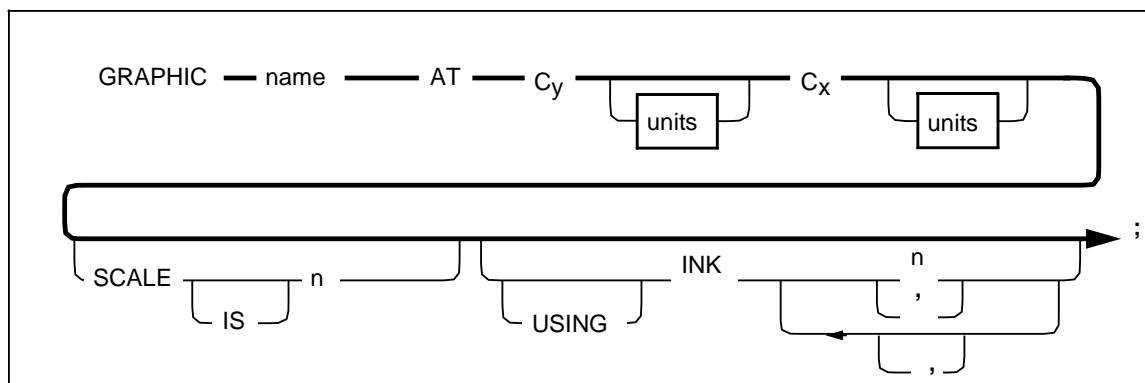C_x specifies the x–axis where you want the section origin located.

units indicates the unit of measure for placement of the section origin. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You must express **DOTS** and **XDOTS** in integers.

REPEAT specifies the number of times you want a section repeated in the orientation you specified.

HORIZONTALLY or **VERTICALLY** specifies the orientation of the lines you want repeated.

EVERY C_i indicates that you want to repeat a section at the increment you have specified.

AT C_a indicates that you want to repeat the section at the absolute coordinate you have specified.

Default **VERTICALLY**, if you specified **REPEAT**.

Examples Long form:

DO SECTION NUSEC 3 TIMES AT 1,1 AND REPEAT HORIZONTALLY EVERY 4;

Short form:

DO NUSEC 3 AT 1,1 REPEAT HORIZONTALLY EVERY 4;

COLUMN command

Use the **COLUMN** command to establish the column width when you specify **TEXT ALIGNED JUSTIFIED IN COLUMN**. You can also use the **COLUMN** command to establish a right boundary for the **DOT LEADER** command.

Parameters   units indicates the unit of measure for the column width. The units are **INCHES, CENTIMETERS, DOTS,** and **XDOTS**. You can use two decimal places when you specify inches and centimeters. You can express **DOTS** and **XDOTS** only in integers.

Figure 4–21 shows the **COLUMN** command syntax flow.
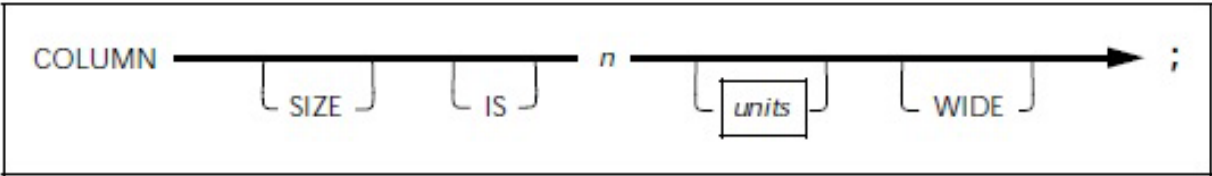
Defaults   The default units specified for the **GRID** command, if you do not specify units.

Column width of 6.5 inches (portrait) or 9 inches (landscape), if you do not specify column width using the **COLUMN** command.

Examples   Long form:

COLUMN SIZE IS 3.8 INCHES WIDE;

Short form:

COLUMN IS 3.8 IN WIDE;

INTERWORD command

The INTERWORD command specifies the interword multiplier used to determine the interword gap.

The interword gap formula

The interword gap is the blank space between words in printed text. It is determined in XFG with this formula:

interword gap = interword multiplier * base gap

where **base gap** is:

- the width of ASCII code point X'20', usually a "blank"
- based on the first named font

interword multiplier is one of these values:

- the value specified by the INTERWORD command
- 1.0, if no INTERWORD command is specified

XFG uses the interword gap you specify as the minimum value. For fully justified text, XFG adjusts the interword gap when needed.

Specifying the interword multiplier

The INTERWORD command specifies the interword multiplier in the formula above.

Figure 4–22 shows the syntax for the INTERWORD command.

Figure 4–22.   **INTERWORD command syntax**
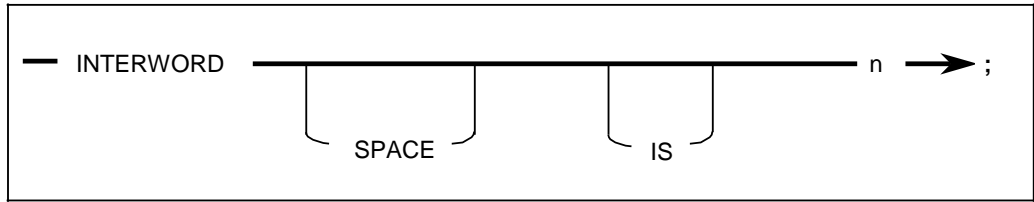
n is the interword multiplier used to determine the interword gap.
This value is multiplied by the base gap.

Examples:

INTERWORD SPACE IS 0.8;

INTERWORD 0.8;

DOT LEADER command

Use the **DOT LEADER** command to define the character string you
specify when you use **DOT LEADER** in the **TEXT AT** command.
The text string portion of the **TEXT AT** command allows for **DOT
LEADER** requests.

Figure 4–23 shows the command syntax flow.

Figure 4–23.   **DOT LEADER command syntax**
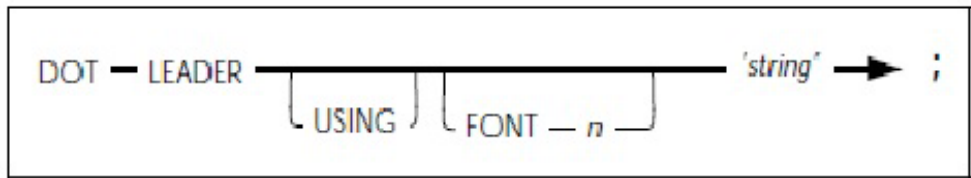
Parameters	'string' specifies the type of dot leader you want to use. Strings cannot exceed 12 characters.
	USING FONT n specifies the font you want to use for the dot leader characters.
Defaults	Single dots of the font currently in use, if you do not specify a font and you use **DOT LEADER** in the **TEXT** command.
Examples	Long form:

DOT LEADER USING FONT 1 '*';

        ****

Sample strings you can use are as follows:

```
' .' produces   . . . . . . . . . . . . . . . . . . .
'*'  produces  * * * * * * * * * * * * * * * * * * * * * * * * *
'-.' produces  -.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
'xo' produces  xoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxoxo
```

Use the **LET** command to define variable names. You can specify all x and y coordinates with either literal values or variable names. If you use variable names, you must first define them with the **LET** command. The **LET** command allows an identifier (variable) to equal an n value. Variable names must begin with an alphabetic character and may be up to 16 characters in length.

You can define up to 50 variables for each form (two to ten variables for each form is typical). You can define the maximum number of variables in one **LET** statement or in individual statements. You can place **LET** statements anywhere in the form following the **FORM** statement, and you can redefine variables any number of times.

You can only set variables to a specific numeric amount. You cannot specify units in the **LET** statement. However, you can specify units (if needed) when the variable is used as a coordinate. For example, if variable Y1 is set to 3 in a **LET** statement and a line is drawn at Y1 inches, it is drawn at 3 inches. If the grid units are dots and no units are specified, then the line Y1 is drawn at 3 dots.

Figure 4–24 shows the **LET** command syntax flow.

Figure 4–24.   **LET command syntax**
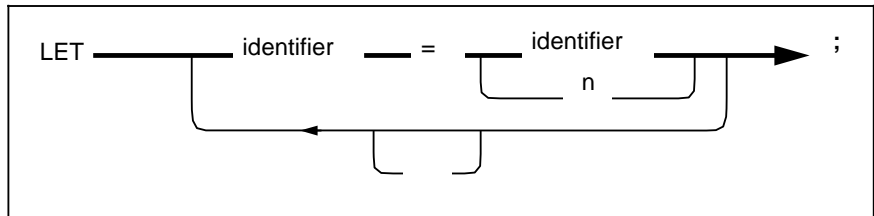
Parameters	identifier = identifier specifies the variable name and value you select for the **LET** command. The first identifier specifies the variable and the second identifier specifies the variable value n.
Default	None.
Examples	Long form:
	LET X1 = 5.5, X2 = 4.0, Y1 = 3.0;
	There is no short form.

Relative coordinate

You can specify all coordinates as literal values or variables. You must first define variables in the **LET** statement. Refer to the "LET command" section for additional information. If you use variables outside of the **LET** command, they are called relative coordinates. These coordinates can also have a positive or a negative value. This allows you to place text items relative to one another. If you want to shift the whole block of text up or have to change the initial values. You can shift text automatically when you add or delete text items.

XFG uses spaces to delimit keywords. Therefore, relative coordinates must not contain spaces between the variable name Y1 and the value **+**.5.
I.E. AT Y1+.5 will work while AT Y1 +.5 will not.

Figure 4–25 shows the variable syntax flow.

Figure 4–25.   **Relative coordinate syntax**
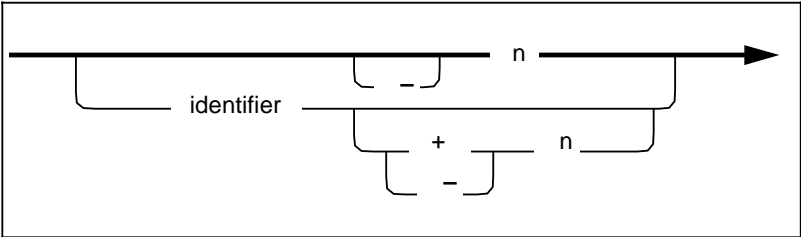
Parameters	identifier is the variable name you specify for the relative coordinate.
	+ or – specifies a positive or negative value that you can apply to the n value.
Default	None.
Examples	The following examples show the long form of the relative coordinate variables.
	In this example, you can define the Y coordinate as the value of Y1 plus 3 centimeters:
	AT Y1+3 CM

In this example, you can define the Y coordinate as the value of Y2 minus 4.5 grid units:

AT Y2–4.5

There is no short form.

In the examples above, the value of **Y1** and **Y2** change. When you refer to **Y1** in the FSL, it is 3 larger. Likewise, **Y2** is 4.5 smaller.

COMMENT command

Use the **COMMENT** command to retain a record of your comments in the source statement. Comments do not appear on the form, and the compiler ignores them. You do not need to enclose the **COMMENT** command text in single quotes.

Figure 4–26 shows the **COMMENT** command syntax flow.

Figure 4–26. **COMMENT command syntax**

Parameter	text specifies the comment text string you want to include in the XFG command statement.
Default	None.
Examples	Long form:

COMMENT ***XFG IS ABLE TO DESCRIBE MANY TYPES OF FORMS***;

There is no short form.

Note that a comment is delimited by a semi-colon (;) and all text between the command COMMENT and the semi-colon is ignored. A comment may span multiple lines.

END command

Use the **END** command to terminate a form description.

Figure 4–27 shows the command syntax flow.

Figure 4–27. **END command syntax**

Parameter	**END** specifies an end–of–file condition from the source input disk file or from the input data stream.
Default	There is no default for the **END** command. It is an optional command.
Example	Long form:

FORM XFG1; ....; END; There is no short

form.

5. Creating a sample form

This chapter describes how to create a simple form electronically using the XPAF Forms Generator (XFG). The forms shown in the following figures are examples only and appear smaller than the actual forms.

The steps on the following pages include both long and short forms of commands. The long form of a command may include optional keywords and system default values that you may omit as you become familiar with the forms creation process.

To make the form description easier to read, you can enter a comment before each command. For example, before specifying column headings, you might enter the following:

COMMENT DEFINE TITLES;

The program does not process comments.

You can enter XFG commands on the same line or you can enter each command on a separate line to make your form description easier to read. You must end each command with a semicolon. Refer to the "Using XFG commands" chapter for complete information on XFG commands and their parameters and defaults.

Figure 5–1 shows the landscape form that you can create using the procedures in this chapter.

Figure 5–1.   **Sample electronic form**

COMPOSITE INVENTORY STOCK STATUS REPORT

COMPOSITE INVENTORY STOCK STATUS REPORT

XEROX2

Follow these steps to set up the basic elements of your form:

1.   Use the **FORM** command to specify both a unique form name that identifies the form to the system and a resolution.

     In this example, the form name is XEROX2 and the resolution is 300 SPI.

     FORM XEROX2, RESOLUTION IS 300 SPI;

2.   Specify the physical paper size.

     This example specifies U.S. letter size:

     PAPER SIZE IS USLETTER;

3.   Specify the grid unit and form origin. The form origin should coincide with variable data origin and form grid units should correspond with the character and line spacing of the variable data.

     The example below shows the long form of the command using FMT1 values:

     GRID UNIT IS 13.6 CPI 8.1 LPI ORIGIN .18 IN .66 IN; This

     example shows the short form of the command: GRID

     FMT1;

     You do not need to use the **LANDSCAPE/PORTRAIT** command to specify page orientation and virtual page size if you specify a predefined format in the **GRID** command.

4.   Specify the fonts you want to use.

     In this example, you specify a large font (UN214A) for the report title, a smaller font (UN107A) for the column headings, and a very small font (UN104A) for the form identifier:

     FONTS UN214A, UN107A, UN104A;

     When you refer to these fonts in later descriptive commands, the first font specified (UN214A) is identified as FONT 1, the second as FONT 2, and the third as FONT 3.

     This example shows entering all of these setup commands in their long form:

     FORM XEROX2 RESOLUTION IS 300 SPI;

     PAPER SIZE IS USLETTER;

     GRID UNIT IS 13.6 CPI 8.1 LPI ORIGIN .18 IN .66 IN;

     FONTS UN214A, UN107A, UN104A;

Since the sample form uses the default resolution, default physical page size, and default grid format, you can omit these commands. This example shows entering all of the setup commands required in their short form:

FORM XEROX2;

FONTS UN214A, UN107A, UN104A;

Using the sample description command

After you finish setting up the sample form, follow these steps to enter descriptive data:

1.   Draw 14 parallel horizontal lines as shown.

Physical and virtual page origin
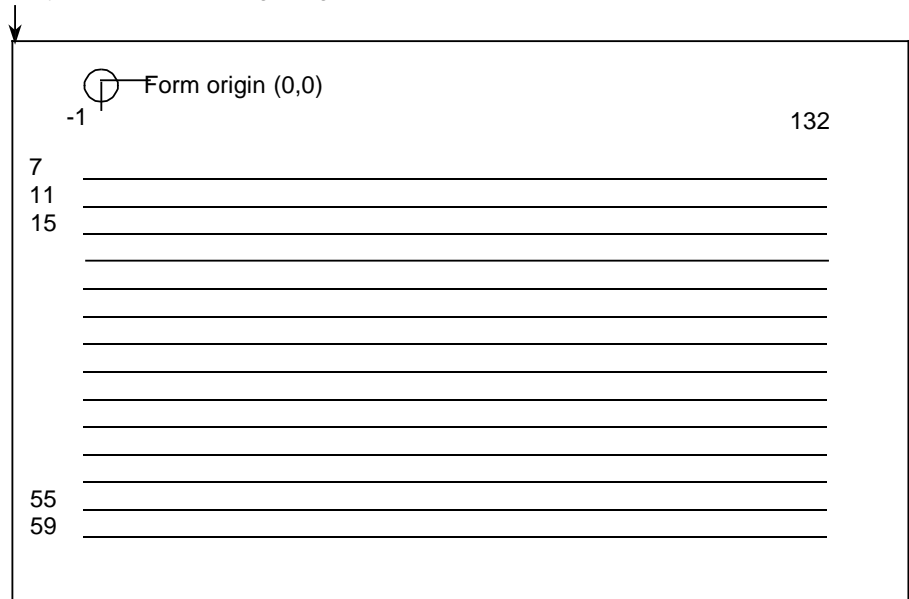
The following example shows the long form of the LINE command:

AT 7 DRAW 14 HORIZONTAL LINES FROM −1 TO 132
USING SOLID 1 AND REPEAT VERTICALLY EVERY 4;

The LINE command draws lines in a horizontal direction under character row 7 and every fourth row thereafter, beginning two character positions to the left of character position 1 and ending at character position 132.

SOLID 1 specifies the appearance of the lines, which are solid and one point wide. Make sure that you stay within the boundaries of the system page when you specify a negative character position.

VERTICALLY is the default direction of the repetition process when printing is horizontal, since the usual intent is to draw parallel lines.

If you omit optional keywords and system defaults, the short form of the command looks like this example:

AT 7 14 LINES –1 TO 132 EVERY 4;

2. Draw 15 parallel vertical lines as shown.

```
      -1  4  12      39  48  55  64  72  79  87  96  104  112  121  132
   7
  11
  15



















  55
  59
```

This example shows the long form of the **LINE** command:

AT –1 DRAW VERTICAL LINES FROM 7 TO 59 USING SOLID 1 AND REPEAT HORIZONTALLY AT 4, 12, 39, 48, 55, 64, 72, 79, 87, 96, 104, 112, 121, 132;

The LINE command draws lines in a vertical direction, beginning at character position –1 and repeated at the specified intervals, with each line extending from character row 7 to character row 59. HORIZONTALLY is the default direction of the repetition process when printing is vertical.

The following example, which omits optional keywords and system defaults, is the short form of the command:

AT –1 VERTICAL LINES 7 TO 59 AT 4, 12, 39, 48, 55, 64, 72, 79, 87, 96, 104, 112, 121, 132;

3. Use the **BOX** command to draw an invisible box and center it above the table.

    This example shows the long form of the command:

    AT 2, –1 DRAW 1 BOX 133 WIDE BY 5 HIGH USING SOLID 0;

The **BOX** command draws an invisible box 133 grid units wide by 5 grid units high under character row 2, beginning two character positions to the left of character position 1. Use SOLID 0 as the thickness of the box line when you want to draw an invisible box. Notice that the horizontal sides of the box begin and end at the same character positions as the horizontal lines of the table.

This example shows the short form of the command: AT

2, −1 BOX 133, 5 SOLID 0;

4. Position the title in the center of the invisible box.

This example shows the long form of the TEXT IN BOX command:

HORIZONTAL TEXT ALIGNED CENTER USING FONT 1 IN CENTER BOX 2, −1 'COMPOSITE STOCK INVENTORY STATUS REPORT';

The **TEXT IN BOX** command places the title COMPOSITE STOCK INVENTORY STATUS REPORT in the center of the box whose upper left corner begins at character row 2, at character position −1. FONT 1 is UN214A, the first font you specified with the FONT command; FONT 2 is UN107A, the second font specified.

5. Specify the column headings.

This example shows the long form of the command:

HORIZONTAL TEXT ALIGNED CENTER USING FONT 2

IN CENTER BOX 7, −1 'CAT' 'NO'

IN NEXT HORIZONTAL BOX 'ITEM' 'NO'

IN NEXT HORIZONTAL BOX 'DESCRIPTION' IN

NEXT HORIZONTAL BOX 'REOR' 'POINT' IN

NEXT HORIZONTAL BOX 'REOR' 'QTY'

IN NEXT HORIZONTAL BOX 'OPEN' 'STOCK'

IN NEXT HORIZONTAL BOX 'VENDOR' 'RECEIPTS' IN

NEXT HORIZONTAL BOX 'STOCK' 'ORDERS'

IN NEXT HORIZONTAL BOX 'COMM' 'STOCK' IN

NEXT HORIZONTAL BOX 'SALES' 'ISSUES' IN

NEXT HORIZONTAL BOX 'BACK' 'ORDERS'

IN NEXT HORIZONTAL BOX 'VENDOR' 'RETURNS' IN

NEXT HORIZONTAL BOX 'STOCK' 'TRANSFERS' IN

NEXT HORIZONTAL BOX 'CURRENT' 'STOCK';

This command places a column heading in each column of the table, using FONT 2. The system automatically centers column headings.

The **TEXT IN BOX** command simplifies the task of placing text in a series of adjacent boxes. You only need to specify the coordinates of the upper–left corner of the first box. Subsequently, when you enter a **BOX** command, the system automatically recognizes the coordinates of a box directly
to the right of the current box. The keywords IN NEXT HORIZONTAL are optional. It is also possible to specify the box directly below the current box, however, you must enter the keywords IN NEXT VERTICAL.

This example shows the short form of the command:

TEXT FONT 2 IN BOX 7, –1 'CAT' 'NO'

BOX 'ITEM' 'NO' BOX

'DESCRIPTION' BOX

'REOR' 'POINT' BOX

'REOR' 'QTY' BOX

'OPEN' 'STOCK'

BOX 'VENDOR' 'RECEIPTS'

BOX 'STOCK' 'ORDERS' BOX

'COMM' 'STOCK'

BOX 'SALES' 'ISSUES'

BOX 'BACK' 'ORDERS'

BOX 'VENDOR' 'RETURNS'

BOX 'STOCK' 'TRANSFERS'

BOX 'CURRENT' 'STOCK';

6. Apply shading to the form.

    This example shows the long form of the command:

    At 11, –1 DRAW 6 BOXES 133 WIDE BY 4 HIGH USING
    SHADING MEDIUM AND REPEAT VERTICALLY EVERY 8;

    This command draws six shaded boxes 133 grid units wide by 4
    grid units high and an identical series of shaded boxes every
    eighth row. The coordinates of the upper left corner of the first
    box are character row 11, character position –1.

    This example shows the short form of the command:

    AT 11, –1 6 BOXES 133, 4 SHADING VERTICALLY EVERY 8;

7. Place the form's identifier and date of issue at the bottom of
    the form.

    This example shows the long form of the command:

    HORIZONTAL TEXT USING FONT 3 AT 61, –1 'XEROX2
    (5/93)';

    This command places the words XEROX2 (5/93) at character row
    61, character position –1, or on the line immediately below the
    table, flush with the left edge of the table. Font
    3 is UN104A, the third font specified in the **FONT**
    command.

    This example shows the short form of the command:

    TEXT FONT 3 AT 61, –1 'XEROX2 (5/93)';

8. Terminate the form description with an **END** command, as
    follows:

    END;

6. Managing and printing sample forms

When you have created your form design using XPAF Forms Generator (XFG) commands, you can compile, store and print a sample of it. After this process completes, your form becomes available for use. This chapter describes the job control language (JCL) to invoke the XFG compiler.

JCL requirements

During installation, your System Administrator or system installer creates all the libraries necessary to run XPAF. If you are not familiar with the items you need to change in the JCL, contact your System Administrator.

The following sample JCL is provided in the XPFSAMP library created during the XPAF install procedure.

Note: The **XINSLSTR** member of your **XINPARM PDS** must contain a valid license string for XFG:

```
# XPAF_XFG feature on product XPAF 5.0, Exp.09-May-2012
 FEATURE=XPAF_XFG ExpDate=09-May-2012 Cpuid=FF01565A2066
 LicenseString=BF591B238607DA318800E23EFACB
```

Note: **The XINSXOAF** member of your **XINPARM PDS** must contain the following parameters for XFG:

```
XLOGDSN=your.hlq.XOAF.XLOG,
 XLOG=Y,
 SLOG=N,
 CFONTDSN=your.hlq.CFONTLIB,
 CLOGODSN=your.hlq.LOGOLIB,
 PAPERSIZE=dfltpapersize,
```

Figure 6–1 shows an example of the JCL you use to compile a form and store it in the database. This JCL may be found in the XPFSAMP library as member XOAFSAMP.

Figure 6–1. **Invoking the XFG compiler**

```
//user$XO  JOB (D498,340),'XOAF BATCH',CLASS=S,MSGCLASS=Y,
//             REGION=4000K,NOTIFY=&SYSUID
//*
//* THIS JOB EXECUTES A BATCH XOAF JOB
//*
//XOAFBAT  PROC
//*
//*       RUN XOAF
//XOAF     EXEC PGM=XOASUP00,REGION=2048K,PARM=&SYSUID
//STEPLIB   DD DISP=SHR,DSN=your.hlq.XPFLOAD
//XINPARM   DD DISP=SHR,DSN=your.hlq.XINPARM
//TABLELIB  DD DISP=SHR,DSN=your.hlq.TABLELIB
//XOAPRINT  DD SYSOUT=X,DCB=(RECFM=FB,LRECL=121,BLKSIZE=6050)
//UJLLIST   DD SYSOUT=X
//SYSPRINT  DD SYSOUT=X
//XPAFXLOG  DD DISP=SHR,DSN=your.hlq.xoaf.XLOG
//SYSABEND  DD SYSOUT=*
//*
//*         ***** FOLLOWING FOR XPDL *****
//CON       DD  DISP=SHR,DSN=your.hlq.XPDLCNF
//MSG       DD  DISP=SHR,DSN=your.hlq.XPDLMSG
//INVXLT    DD  DISP=SHR,DSN=your.hlq.INVXLT
//PCCFIL    DD  DISP=SHR,DSN=your.hlq.PCCFIL
//TYPFIL    DD  DISP=SHR,DSN=your.hlq.TYPFIL
//XPDLDFT   DD  DISP=SHR,DSN=your.hlq.XPDLDFT
//       PEND
//*
//*       ========>         RUN XOAF          <========
//*
//XOAF     EXEC XOAFBAT
//XOAF.CFONTLIB DD DISP=SHR,DSN=your.hlq.CFONTLIB
//XOAF.CLOGOLIB DD DISP=SHR,DSN=your.hlq.LOGOLIB
//XOAF.XOAIN DD *
COMPILE FORM('xxx.FSLLIB(BANF)') TO('your.hlq.CFORMLIB')
/*
```

Downloading and printing a sample

If you want to download and print a sample of the form, run the XFGUTIL job following a successful compile. Figure 6–2 shows an example of JCL you use to download and sample a form on the printer. This JCL is available in the XPFSAMP library as member XFGUTIL.

Figure 6–2. **Downloading and printing a sample**

```
//user$XF  JOB (D498,340),'XFG UTILITY',CLASS=S,MSGCLASS=Y,
//            REGION=4M,NOTIFY=&SYSUID
//*
//* THIS JOB EXECUTES THE XFG UTILITY PROGRAM TO DOWNLOAD AND
//*      SAMPLE A FORM FROM THE CFORMLIB
//*
//* THE FORM NAME IS GIVEN IN THE PARM= STATEMENT
//* THE IDEN PARAMETERS ARE GIVEN IN THE IDEN FILE
//* THE OUTPUT IS WRITTEN TO SYSPRINT
//*
//* THE FORM WILL BE DOWNLOADED WITH THE DELETE OPTION
//* IF ,D IS CODED AFTER THE FORM NAME IN THE PARM STATEMENT
//*
//XFGUTIL EXEC PGM=XFGUTIL,REGION=4M,PARM=(form,D)
//STEPLIB   DD DISP=SHR,DSN=your.hlq.XPFLOAD
//XINPARM   DD DISP=SHR,DSN=your.hlq.XINPARM
//CFORMLIB  DD DISP=SHR,DSN=your.hlq.CFORMLIB
//SYSPRINT  DD SYSOUT=X
//IDEN      DD *
PREFIX='@@@DJDE'
OFFSET=0
SKIP=8
JDE=DFLT
JDL=DFAULT
PRINTFORMNAME=Y
TEMPFORMNAME=TMPZZZ
/*
..
```

The parameters in the inline IDEN DD statement are as follows:

PREFIX	The text string to use as the prefix for the DJDE records generated
OFFSET	The offset of the DJDE prefix from the start of the record
SKIP	The byte position of start of the DJDE parameters in a DJDE record
JDE	If specified, a JDE= DJDE will be generated with the specified value
JDL	If specified, a JDL= DJDE will be generated with the specified value
PRINTFORMNAME	Indicates whether or not XFGUTIL should add variable text to the form sample that displays the form name
TEMPFORMNAME	Specifies a different form name with which the form will be downloaded and sampled. This can be used to avoid overwriting a production version of a form that you have updated and wish to sample before going live.

If you are using XPAF to print to your LPS printers you may use a simple IEBGENER job with Extended JCL to download and sample your forms. Figure 6–3 shows an example of JCL you use to download and sample a form on the printer using XPAF.

Figure 6–3. **Downloading and printing a sample with XPAF**

```
//user$SF  JOB (D498,340),'BANF FORM SAMPLE',CLASS=S,MSGCLASS=Y
//*
//*       SAMPLE FORM FROM CFORMLIB
//*
//SAMPFORM EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//*
//OUTP     OUTPUT LINECT=0,   PORTRAIT
//              CLASS=X,FORMS=PORT,BANSTYLE=NONE,
//              JDL=XPAF,JDE=LETTER,FEED=TRAY2,
//              XFORM1=BANF,REVFORM=BANF,
//              PMODE=POR,TOF=1,BOF=128,XDUPLEX=NO,
//              BEGIN1=(P3,IN,1,IN),FONT0=P0612B
//*
//SYSUT2   DD SYSOUT=(,),DCB=(RECFM=FA,BLKSIZE=80),
//              OUTPUT=(*.OUTP)
//SYSIN    DD DUMMY
//SYSUT1   DD *
1SAMPLE OF FORM BANF.FRM
/*
//
```

Editing an existing form

If you want to edit an existing form, select the forms source language (FSL) with your ISPF editor, make the necessary changes and then compile your edited form with XFG.

7. Messages

This chapter lists the error messages that appear when problems occur, an indication of the cause of the message and the action you can perform to correct the problem.

Message indicators

- **I** (informative) indicates the status or activity of the system where no action is needed by the operator. FRM will be generated.

- **W** (warning) indicates an error in the description of your form. It also indicates the action you may be required to take to correct the problem. FRM will be generated.

- **E** (error) indicates a syntax or similar error. FRM will not be generated.

- **F** (fatal) indicates a serious error that can prevent you from compiling your form or may abnormally end (ABEND) your session. FRM will not be generated.

XFG1001W   **Discarding excess FDL**

Indication	Excess FSL statements were found beyond the END statement.
Action	Verify that these statements do not belong in the form, then move or delete them.

XFG1002W   **Semicolon expected**

Indication	A semicolon was expected in the statement. The compiler inserted one and continued with the compile.
Action	None required.

XFG1003W   **Character not in FDL alphabet**

Indication	A character was found in the FSL statements that does not exist in the FSL alphabet. The character was ignored.
Action	Verify that the character does not belong in the FSL statements, then move or delete it.

XFG1004W   **DRAWN and REPEAT EVERY disagree, using 2**

Indication	The number of lines entered for the DRAW command must be greater than 1. The compiler substituted 2 for the invalid existing number.
Action	Enter the correct values and recompile the form.

XFG1005W   **Undefined Section Name <name>, command ignored**

	Indication	The section name used has not been defined.
	Action	Verify that the section name was spelled correctly or make sure the section is defined prior to invoking it with the **DO SECTION** command.

XFG1006W   **Invalid line weight, using HAIRLINE**

	Indication	An invalid line weight was entered. The compiler substituted a value of **HAIRLINE**.
	Action	If **HAIRLINE** is not the value you want, correct the FSL and recompile the form.

XFG1007W   **Invalid border weight, using HAIRLINE**

	Indication	An invalid border weight was entered for a box. The compiler substituted a value of **HAIRLINE**.
	Action	If **HAIRLINE** is not the value you want, enter a valid border weight.

XFG1008W   **Possible missing or unmatched quotes**

	Indication	The compiler cannot detect a closing quote.
	Action	Make sure that all quotes in the preceding commands in your FSL are matched.

XFG1009E   **600 spi cannot be used with highlight color form**

	Indication	There is no Xerox printer support for highlight color in 600 spi mode.
	Action	Use 300 spi for color forms or use black and white mode for 600 spi.

XFG1010W   **Box not found**

	Indication	A box with the origin specified, or the **NEXT BOX** could not be found.
	Action	Recalculate the box coordinates. To ensure the box is found, use dot coordinates for the box origin.

XFG1012W   **Unexpected fraction, using integer**

	Indication	A fraction was entered where an integer was expected. The integer part of the number is used.
	Action	Change fraction to an integer value and recompile the form.

XFG1014W   **Invalid resolution, using 300 spi**

	Indication	An invalid resolution was entered in the **RES** statement.
	Action	Change the resolution to 300 or 600 spi.

XFG1015W   **Unexpected end of text (terminating code expected)**

Indication	The terminating code character is missing at the end of the text string. If you entered the text correctly, this could indicate a damaged file.
Action	Reenter the text with the terminating code. If the problem persists, contact your System Administrator.

XFG1016W   **Text will not fit in column**

Indication	The entered text is too long to fit in the specified column.
Action	Reduce the length of the text string or increase the size of the column to fit the text. Otherwise the text will not be printed.

XFG1018W   **Text might not fit in box**

Indication	The entered text block exceeds or is equal to the dimensions of the box in which it is to be placed.
Action	Check the printed form. If text crosses the boundary of the box, reduce the length of the text string or increase the size of the box to fit the text.

XFG1020W   **Missing END statement**

Indication	End of file was found with no **END** statement.
Action	No action is required, since the **END** statement is optional. You may want to add the **END** statement and recompile.

XFG1021W   **Setup command out of sequence**

Indication	A setup command (**FORM/RESOLUTION**, **FONT**, **GRID**, **LANDSCAPE/PORTRAIT**, **INK**, **ICATALOG**, or **PALETTE**) was entered after a formatting command (such as **LINE**, **TEXT**, **BOX**).
Action	Place the setup command with the rest of the setup commands at the beginning of the form source and after the **FORM/RESOLUTION** command.

XFG1022W   **Identifier name too long, using <name>**

Indication	An identifier was used that exceeds the maximum 16 characters allowed.
Action	None required. The system truncates the identifier at 16 characters.

XFG1024W   **Section name too long, using <name>**

Indication	The section name used exceeds the maximum 16 characters allowed.
Action	None required. The system truncates the section name at 16 characters.

XFG1025W   **Logos not supported for XES forms**

Indication	You used the **LOGO** command while generating an XES form. In this release, logos are treated as fonts in XES forms.
Action	Do not use the **LOGO** command while generating XES forms or, if possible, enter the logo as a font.

XFG1026W   **Catalog name too long, using <six-character catalog name>**

	Indication	The catalog name used exceeds the maximum six characters allowed.
	Action	None required. The system truncates the catalog name at six characters.

XFG1027W   **Dot Leader too big for column**

	Indication	You specified a dot leader that exceeds the column setting.
	Action	Change the dot leader or the column setting.

XFG1028W   **Text String Longer than 256 characters**

	Indication	A text string contains more than the maximum 256 characters.
	Action	Reduce the size of the text string.

XFG1029W   **No substitution allowed for Logo <logo name>**

	Indication	You entered the **USING INK** parameter in the **LOGO** command when the NO SUBSTITUTION flag was set for the specified logo in the logo resource.
	Action	Do not use the **USING INK** parameter with this logo.

XFG1030W   **Can not close data base**

	Indication	The compiler is not able to close the database because it is being used by another user.
	Action	Try again in a few minutes. If the problem persists, note the return code number and contact your System Administrator.

XFG1031W   **Palatte name too long, using <thirty-two-character palette name>**

	Indication	The palatte name used exceeds the maximum thirty-two characters allowed.
	Action	None required. The system truncates the palatte name at thirty-two characters.

XFG1032W   **Column off right of physical page**

	Indication	The specified column width exceeds the physical page boundary.
	Action	Resize the column width to fit the physical page boundary.

XFG3001E   **Syntax error on <command segment>**

	Indication	The specified command segment contains a syntax error.
	Action	Correct the error and recompile the form.

XFG3002E   **DRAW N and REPEAT AT counts disagree**

Indication	The **DRAW** and **REPEAT AT** counts do not agree.
Action	Correct and recompile the form.

XFG3003E   **N TIMES and REPEAT AT counts disagree**

Indication	The number of **TIMES** and the **REPEAT AT** counts do not agree.
Action	Correct and recompile the form.

XFG3004E   **Horizontal text inconsistent with top/bottom**

Indication	Horizontal text is aligned **TOP** or **BOTTOM** instead of **LEFT** or **RIGHT**.
Action	Correct the text direction with the proper alignment.

XFG3005E   **Vertical text inconsistent with left/right**

Indication	Vertical text is aligned **LEFT** or **RIGHT** instead of **TOP** or **BOTTOM**.
Action	Correct the text direction with the proper **ALIGNMENT**.

XFG3006E   **Identifier exceeds 6 characters**

Indication	An identifier name, such as **FORM** name, exceeded six characters.
Action	Correct the identifier name and recompile the form.

XFG3007E   **Stacked text and justify conflict**

Indication	Stacked text cannot be justified.
Action	Remove the **JUSTIFY** and recompile the form.

XFG3008E   **Setup command not allowed within section**

Indication	One of the setup commands such as **FONT** or **INK** was found within a section.
Action	Move the setup command to the beginning part of the FSL.

XFG3009E   **Code point <code point> in font <font name> has a zero width**

Indication	Some of the characters in this font have a zero width value.
Action	Reload the font.

XFG3010E   **DO SECTION not allowed within section**

Indication	A **DO SECTION** command was found within a section.
Action	Remove the **DO SECTION** command and recompile the form.

XFG3011W   **Ambiguous positioning of <position>, <position> given. <position>, <position> used**

Indication	Ambiguous positioning of text in a box such as Left, Left instead of Left with Bottom, Center, or Top.
Action	Correct the ambiguous position.

XFG3012E BEGIN SECTION not allowed within section

Indication	A **BEGIN SECTION** command was found within a section.
Action	Remove the command or end the current section.

XFG3014E   **Value out of range**

Indication	A zero value or a value that exceeds page limits was encountered.
Action	Correct the value to one within the valid limits.

XFG3015E   **Keyword out of context/sequence**

Indication	You sequenced a keyword incorrectly.
Action	Enter the keyword in the correct order.

XFG3016E   **FONT command not defined**

Indication	Text is referenced in the form and no **FONT** command was specified.
Action	Enter the **FONT** command with the valid fonts you want to use in the form.

XFG3017E   **Invalid font index**

Indication	The font index in the **TEXT** command is outside the range of those defined in the **FONT** command.
Action	Correct the font number to a value within the range of fonts defined, or define a new font if the font is missing from the font list.

XFG3018E   **INK command not defined**

Indication	Inks are referenced in the form and there is no **INK** command specified.
Action	Enter the **INK** command with the valid inks you want to use in the form.

XFG3019E	**Invalid ink index**	
	Indication	The ink index referenced in a **LINE**, **TEXT**, **BOX**, **GRAPHIC**, or **LOGO** command is outside the range of those defined in the **INK** command.
	Action	Correct the ink number so that it is within the range of inks defined or define a new ink if the ink is missing from the ink list.

XFG3020E	**Line off (top/bottom/left/right) of physical page**	
	Indication	The line went off either the top, bottom, left, or right of the physical page.
	Action	Correct the line coordinates so that it does not extend beyond the physical page.

XFG3021E	**Font name <font name> exceeds 20 characters**	
	Indication	The font name exceeded the 20 character maximum length.
	Action	Use a valid font name, which should consist of 20 characters or less.

XFG3024E	**Line end coordinate <= start coordinate**	
	Indication	The line ending coordinate is less than or equal to the line starting coordinate.
	Action	Correct the line coordinates and recompile the form.

XFG3030E	**Box off (top/bottom/left/right) of physical page**	
	Indication	The box went off either the top, bottom, left, or right of the physical page.
	Action	Correct the box coordinates so the box does not extend beyond the physical page.

XFG3034E	**Box width is zero**	
	Indication	The width of the box was zero.
	Action	Correct the box width and recompile the form.

XFG3035E	**Box height is zero**	
	Indication	The height of the box was zero.
	Action	Correct the box height and recompile the form.

XFG3040W	**Text off (top/bottom/left/right) of physical page**	
	Indication	The text went off either the top, bottom, left, or right of the physical page.
	Action	Correct the text coordinates so the text does not extend beyond the physical page.

XFG3050W	**Logo off (top/bottom/left/right) of physical page**	
	Indication	The logo went off either the top, bottom, left, or right of the physical page.
	Action	Correct the logo coordinates so the logo does not extend beyond the physical page.
XFG3585W	**nn DAYS LEFT IN GRACE PERIOD**	
	Indication	The expiration date has passed for the XPAF Forms Generator. A grace period has been entered and there are nn days left in the grace period before the product will cease to function.
	Action	Contact Xerox to obtain a valid license.
XFG3589W	***** WARNING *** WARNING *** WARNING *** WARNING *****	
	Indication	This message is issued in conjunction with other messages.
	Action	Refer to the actions for the messages issued with XFG3589W.
XFG3593E	**NO VALID LICENSE STRING FOUND FOR XFG**	
	Indication	A valid license for the XPAF Forms Generator could not be found.
	Action	Contact Xerox to obtain a valid license.
XFG3594E	**XFG LICENSE HAS EXPIRED**	
	Indication	The XPAF Forms Generator license has expired.
	Action	Contact Xerox to obtain a valid license.
XFG3595W	**XFG LICENSE WILL EXPIRE IN nn DAYS**	
	Indication	The XFG License has less than 30 days before it will expire.
	Action	Contact Xerox to renew the license string.
XFG4001F	**Unexpected end of file**	
	Indication	The **END** statement was encountered before any description commands. There is nothing to compile.
	Action	Move the **END** command to the last line of the FSL and recompile the form
XFG4006F	**Cannot open FSL**	
	Indication	There is a serious system error. The compiler was not able to open the input FSL file.
	Action	Verify that the file name is correct and rerun the job. If the problem persists, contact your System Administrator.
XFG4009F	**Cannot open IFF**	

	Indication	There is a serious system error. The compiler was not able to open the immediate IFF file created by the parser.
	Action	Contact your System Administrator.

XFG4010F **Grid format not defined <format name>**

	Indication	The format specified in the **GRID** command does not exist.
	Action	Correct and recompile the form.

XFG4012F **Too many fonts specified**

	Indication	The **FONT** command specified more than the limit of 32 fonts.
	Action	Delete the number of fonts used so the maximum of 32 is not exceeded and recompile the form.

XFG4014F **Cannot open scratch file <file name>**

	Indication	There is a serious system error. The compiler was not able to obtain a scratch file used during compilation.
	Action	Verify that the file name is correct and rerun the job. If the problem persists, contact your System Administrator.

XFG4020F **Font not found <font name>**

	Indication	The font specified in the FONT command was not found in the database.
	Action	Load the necessary font into the font database or correct the FONT command with the correct name and recompile the form.

XFG4022W **Logo not found <logo name>**

	Indication	The logo specified in the **LOGO** command was not found in the database.
	Action	Correct the name of the logo or load the required logo into the database and recompile the form.

XFG4023F **Undefined font index**

	Indication	A font index was referenced in a **TEXT** command that was not defined in the font list.
	Action	Correct the **TEXT** command to the correct reference from the font list.

XFG4034F **Invalid graphic scale factor**

	Indication	An incorrect graphic scaling factor was entered. The scaling factor must be an integer in the range 1 to 8.
	Action	Correct the scaling factor and recompile the form.

XFG4040F **Text string exceeds buffer limits**

	Indication	The text string entered exceeded the text buffer limits of 4096

characters.

	Action	Shorten the text string by breaking the text up into two or more **TEXT** commands.

XFG4053F   **Maximum number of inks allowed (255) exceeded**

	Indication	Your FSL uses more than the maximum 255 inks (colors) allowed.
	Action	Reduce the number of inks in your FSL to 255 or less.

XFG4054F   **Cannot reassign identifier to itself**

	Indication	You made an invalid use of the **LET** command. An identifier cannot be on both the right–hand and left–hand side of the assignment statement.
	Action	Remove the identifier from the right–hand side of the assignment statement.

XFG4062F   **Invalid LPS tape label**

	Indication	There is a serious system error. The compiler was not able to process the tape label of the compiled form it just created.
	Action	Contact your System Administrator.

XFG6020I   **There are N forms compiled**

	Indication	There are n forms compiled in this job, either through multi-form or multi-member selection.
	Action	None

A. System Considerations

Consider the following items when designing and planning your form:

- Superimposed edges when adjoining boxes
- Text size in boxes
- Lines with sections

These items are especially important when designing complex forms. You can avoid error messages and save the time it takes to correct the problems by being aware of the system considerations described in this appendix.

Boxes and their image

You can draw boxes with just a single box statement or with line–by–line statements. The box statement may not work correctly if it specifies two boxes adjoined by a single side. This causes the adjoining side lines of the boxes to become superimposed. Since the two lines use the same pair of coordinates, a line density problem may occur.

Text in boxes

If you are using a **TEXT IN BOX** command, XFG determines if the given text can fit in the box. If your text font size exceeds the maximum allowable size, XFG generates an error message (XFG1018W). If you see this error, you have the following choices:

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

Refer to the "Messages" chapter for more information.

Using lines within sections

When you create a form by invoking defined sections, make sure you do not begin and end lines at section boundaries. It is more efficient to allow the line to continue through a series of sections.

B. FMT Grid Format Properties

This appendix lists the **FMT** parameters you use in the **LANDSCAPE/PORTRAIT** and **GRID** commands. It includes the FMT properties for 8.5 by 11–inch paper, 8.5 by 14–inch paper, and A4 paper. The **FMT** defaults are **FMT1** (landscape) and **FMT6** (portrait). Table B–1 lists the grid and origin substitutions you can use when you print on 8.5 by 11–inch paper.

Table B–1. **FMT properties for 8.5 by 11–inch 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 0.22 INCH, 0.50 INCH;

Table B–2 lists the grid and origin substitutions you can use when you print on 8.5 by 14–inch paper.

Table B–2.   **FMT properties for 8.5 by 14–inch 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;

Table B–3 lists the grid and origin substitutions you can use when you print on A4 paper.

Table B–3.   **FMT properties for A4 paper**

Specification	Substitution
FMT1A	LANDSCAPE; GRID UNIT IS 12.5 CPI, 8.3 LPI, ORIGIN 0.18 INCH, 0.57 INCH;
FMT2A	LANDSCAPE; GRID UNIT IS 14.3 CPI, 8.3 LPI, ORIGIN 0.18 INCH, 0.60 INCH;
FMT3A	LANDSCAPE; GRID UNIT IS 12.5 CPI, 11.1 LPI, ORIGIN 0.18 INCH, 0.57 INCH;
FMT4A	LANDSCAPE; GRID UNIT IS 14.3 CPI, 11.1 LPI, ORIGIN 0.18 INCH, 0.60 INCH;
FMT5A	LANDSCAPE; GRID UNIT IS 10.0 CPI, 6.0 LPI, ORIGIN 0.22 INCH, 0.85 INCH;
FMT6A	PORTRAIT; GRID UNIT IS 13.6 CPI, 8.1 LPI, ORIGIN 0.91 INCH, 0.46 INCH;
FMT7A	PORTRAIT; GRID UNIT IS 12.0 CPI, 6.0 LPI, ORIGIN 0.85 INCH, 0.39 INCH;
FMT8A	PORTRAIT; GRID UNIT IS 10.0 CPI, 6.0 LPI, ORIGIN 0.85 INCH, 0.39 INCH;
FMT9A	LANDSCAPE; GRID UNIT IS 20.0 CPI, 10.0 LPI, ORIGIN 0.14 INCH, 0.85 INCH;
FMT10A	PORTRAIT; GRID UNIT IS 17.6 CPI, 12.5 LPI, ORIGIN 0.57 INCH, 0.39 INCH;
FMT11A	PORTRAIT; GRID UNIT IS 20.0 CPI, 12.5 LPI, ORIGIN 0.57 INCH, 0.39 INCH;

C. Approved abbreviations for reserved words

Abbreviations can be helpful in reducing the number of characters or lines you need to describe a function or command. However, while abbreviations are helpful, they can create problems if you do not use them properly. If you use abbreviated reserved words, make sure they are identical to those in the approved list.

The following list contains approved abbreviations for reserved words. The approved abbreviation is shown in capital letters or in parentheses beside the word. If an abbreviation is not shown, you cannot abbreviate the reserved word.

ALIgned	GRAphic	page
and	grid	PALette
A3	HAIrline	PAPer
A4	HEAvy	per
BEGin	HIGh	POInts
BLAck	HORizontal	PORtrait
BOTtom	ICAtalog	pts
BOXes	in	REPeat
BROken	INch	RESolution
by	INKs	RIGht
B4	INTerword	SCAle
CENter	IREsult	SECtion
CENTImeters (cm)	is	SHAding
color	JUStified	SHIft
COLumn	LANdscape	SIZe
COMment	LEAder	SOLid
characters per inch (CPI)	LEFt	SPAced
DEFault	let	spots per inch (SPI)
do	LIGht	TEXt
dot	LINes	TIMes
dots	logo	to
DOTTed	lines per inch (LPI)	top
DRAw	LITeral	UNIt
end	MEDium	USIng
ENDSection	MEMber	USLEGal
EVEry	new	USLETter
FONTs	NEXt	VERtical
FORm	NOSubstitution	WIDe
FROm	ORIgin	XDOts

D.   Supported paper sizes

You can design forms for a variety of paper sizes, however, you must be aware of the paper sizes your printer supports. Even though XFG supports a paper size, it may not be available or supported by your printer. If you are not sure of the size of paper your printer supports, contact your System Administrator.

Table D–1 lists the paper sizes that XFG supports.

Table D–1.   **Paper sizes available for XFG**

Paper size (inches)	XFG designation
8.5 by 11	USLETTER
8.5 by 14	USLEGAL
16.54 by 11.69	A3
8.27 by 11.69	A4
10.12 by 14.33	B4
8 by 10.2	(None)
8 by 10.5	(None)
8 by 13	(None)
8.27 by 10.63	(None)
8.27 by 13	(None)
8.37 by 10.78	(None)
8.46 by 10.83	(None)
8.46 by 12.4	(None)
8.46 by 14.02	(None)
8.5 by 10.75	(None)
8.5 by 13	(None)
11 by 17	(None)

E.   Glossary

abort	To prematurely terminate the printing of a job, execution of a program, or execution of a nonprinting task.
algorithm	A step by step procedure for solving a problem.
alignment	Positioning of an image on a page for printing.
alphanumeric	Set of letters and whole numbers.
ASCII	American Standard Code for Information Interchange. Standard 7–bit code that represents alphanumeric information. Each alphanumeric character and several nonprinting characters are assigned a binary number covering 128 possible characters. It is used for information interchange among data processing systems, data communication systems, and associated equipment. See also EBCDIC.
application	Use to which a computer program or system is applied (for example, sorting employee records).
application software	Host– or LPS–resident software that directs the computer to perform specific tasks or functions. Common business applications include payroll, accounting, inventory, and so on.
argument	Independent variable of a function, such as a parameter.
ascender	Portion of the lowercase letter that extends above the main body of the character (such as b, d, or h).
band	Unit in a printed page that optimally handles image processing in sequence.
batch	Body of data accumulated and grouped by kind before processing.
batch processing	Process that allows repetitive operations to be performed sequentially on batched data without much involvement from the computer operator.
binary	Numbering system with a base of 2. All numbers are represented by combinations of 1 and 0.

bit	Binary integer. Smallest unit of information recognized by a computer. Numeric integer of 1 or 0 represent the binary scale of notation. In computers and related devices, the binary numeric values can be represented by a positive or negative electrical pulse. As a general rule, 1 equals ON, TRUE, or YES; and 0 (zero) equals OFF, FALSE, or NO. A number of bits strung together can represent a character to the computer.

Multipliers of the bit are as follows:

- 1 byte = 8 bits
- 1 kilobyte (KB) or 1,024 bytes = 8,192 bits
- 1 megabyte (MB) or 1,048,576 bytes = 8,388,608 bits

bitmap	Visual representation of graphic images in which a bit defines a picture element (pixel). For example, if a bit is 1, the corresponding pixel is printed.
bitmapped	Display image generated bit by bit for each point or dot. A software–driven scanner is used to create characters and graphics.
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.
block length	Number of characters or bytes contained in a block of data. The block is treated as a unit in the computer. Block lengths may be specified in units such as records, words, computer words, or characters.
boot	To load initial operating instructions for a program into the memory of the computer. These instructions direct the loading of the rest of the program. Booting may require the entry of a few commands or keystrokes (soft boot) or the turning on of a switch to begin the process (hard boot).
buffer	Compartment of memory in which data is stored during transfer from one device to another.
byte	Fixed number of bits operated upon as a unit (in data processing, usually 8). A byte may represent a character, a machine instruction, or some other logical unit of information.
carriage return	Control character that causes the printing system to start printing at the left margin of the current line (unless set to be interrupted as a line end).
centralized printer	Class of Xerox printers that includes the 4050, 4090, 4135, 4650, 4850, 4890, 8790, and 9790 printers. Centralized printers are frequently placed in a central location to handle a large volume of output.
channel	1. In data communications, path or line that allows two or more devices to communicate. 2. In computers, a path for communication between the central processing unit (CPU) and peripheral devices.
character	Single printable letter (A to Z), numeral (0 to 9), symbol (& % #), or punctuation marks (, . ! ?) used to represent data. Characters can also be nonprinting, such as a space, tab, or carriage return.

character cell	Digitized space containing a single character within a font set.
character code	Code representing alphanumeric information, such as ASCII.
character map	Grid of blocks, where each block corresponds to a letter, number, punctuation mark, or special character.
character set	Number of different characters used by a particular device, including alphabetic, numeric, and special characters such as symbols.
CLPS	Centralized laser printing system. Technology that uses a laser to transfer character forms to a page. See also LPS.
cpi	Characters per inch. Designates the number of characters per inch for a particular typeface. See also pitch.
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.
code	1. Set of symbols representing data or instruction to a computer. 2. To write a list of software instructions that cause the product or system to perform specified instructions.
code conversion	Translation of one type of character or symbol code to another.
collate	To arrange or assemble into ordered sets.
column	Vertical arrangement of characters.
command	User instruction to a computer. Commands are words, mnemonics, or characters that cause a computer to perform predefined operations.
command language	Set of commands that can be used to instruct a system on how to perform a task.
comment	Explanation added to program instructions. They are ignored by the computer.
compatibility	Characteristics of computer equipment permitting one device to use the same information or programs as another device without conversion or code modification.
compiler	Software that translates instructions written in high–level language into machine language for execution by a system.
composite character	Character that has multiple bitmap references combined into one bitmap and stored on disk in font memory.
computer language	Computer–oriented language consisting solely of computer instructions. See also machine language.
computer program	Sequence of instructions suitable for processing by a computer.
concatenate	To connect or link in a series, such as files grouped together for faster processing.
condition code	Code returned by a job that is used to verify successful execution of the job.
configuration	One or more computers and related devices (terminals, printers, and so forth) interconnected and programmed to operate as a system.
console	Functional unit containing devices used by an operator to communicate with an operating system. It may consist of a display, keyboard, and certain switches or other controls.
consumable supplies	Supplies such as paper and dry ink that are depleted during the course of normal operation.

control program	Program that supports the operating system by monitoring the flow of jobs, tasks, processing, and so forth, within the system.
coordinate	Point on the x–axis and y–axis that determines a grid position.
copy	To duplicate data in a new location or on an additional storage medium. For example, to copy files from disk to tape.
copy–sensitive	Term used to indicate jobs in which multiple copies of a report will contain different data, such as paychecks and banking statements.
data	1. In general, facts, numbers, letters, symbols, and so forth, that can be processed or produced by a computer. 2. In data processing, the source data, or raw data, entered for processing, as opposed to the results obtained by processing.
database	Information to meet specific processing and retrieval needs. Generally applies to integrated files of data, arranged for access by many subsystems.
data file	Collection of related data records organized in a specific manner so that each record is similarly structured, for example, a payroll file set up with one record for each employee, structured with the last name first, followed by the rate of pay and all deductions.
data processing	Operations carried out on data by means of algorithms and programs to produce information or a specific result. The rearrangement of data into a suitable form for further use.
data set	Collection of logically related records stored in a database. See also file.
data storage	Preservation of data on various media, such as tape and disks.
debug	To detect and correct errors in a program.
decentralized printer	Class of Xerox printers that includes any remotely connected printers. Decentralized printers are frequently placed in a remote location to handle a lower volume of output.
default	Value assigned to a field by the system. Default fields may be used for such items as document formats, menu selections, input fields, font selection, and paper or image size. You can change the default value of a field.
device	Any piece of hardware other than the central processing unit (CPU).
digitize	To express or represent data in digital (binary) format so it can be processed electronically.
display	Viewing device (monitor) used to visually display the results of operator interaction with the system.
DJDE	Dynamic job descriptor entry. Command within an input data stream used to modify the printing environment dynamically.
dot	Unit of measurement representing a fraction of an inch; for example, 300 dots per inch (dpi). Dot may also refer to a picture element (pixel) or spot.
dpi	Dots per inch. Indicates the number of dots per inch displayed on a terminal screen or printed to form a character or graphic.
download	To load files from the host onto LPS system disks.
dry ink	Minute, dry particles of resin and carbon black used to create images. Dry ink can accept an electrical charge.

duplex	Printing or copying on both sides of the paper. See also simplex.
easy values	Font point sizes, orientations, and names explicitly defined in Font Interchange Standard (FIS).
elite	Smallest size standard typewriter type: 12 characters per horizontal inch. See also pica.
embedded blanks	Blank spaces within a command line.
enabler	Hardware devices or software packages that allow the printer to perform as specified or better.
escapement	Lateral positioning of characters or font families.
EBCDIC	Extended binary coded decimal interchange code. Coded character set consisting of 8–bit coded characters. It can accommodate 256 characters. See also ASCII.
FCB	Forms Control Buffer. Buffer for controlling the vertical format of printed output.
field	1. Part of a record that serves a similar function in all records of the group (for example, name and address field). 2. Area or setting of practical activity or application.
file	Set of records or text that can be stored and retrieved. An organized, named collection of records treated as a unit. For offline, it is the data between the two tape marks. For online, it is the data between banner pages.
fixed font	Font containing characters with fixed spacing.
fixed pitch	Font set in which every character cell has the same width. In reference to character sets, this term describes typefaces in which all character cells are of equal width. Characters are monospaced as opposed to proportionally 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	Non-spacing accent characters that can be combined with characters and printed as a composite.
font	Data file for a particular electronic printer that prints a character set of a specific typestyle (letter, form, and shape). Each font has its own unique characteristics, including the following:

- Size—expressed in points
- Weight—expressed as light, medium, or bold
- Stress—Roman or italic
- Width—regular, condensed, or expanded
- Orientation—the print direction of the characters, relative to the page

In addition to alphanumeric characters, a font can contain various symbols or logos.

font white space	Spacing character in the font.

form	Electronic version of a form that can include lines, boxes, text, shading, logos, and embedded text. Forms can be printed without variable data or merged with variable data during the printing process.
form feed	Keyboard printer control character that causes the printer to skip to the top of the next page.
format	1. Layout of a document, including margins, page length, line spacing, typeface, and so forth. 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 the acceptance of data.
FDL	Forms description language. Keyword oriented language used to define the appearance of an electronic form including horizontal and vertical lines, boxes, descriptive text, and so forth. The source language is referred to as forms source language (FSL) and the object language (after compilation) is referred to as .FRM. See also FSL.
FSL	Forms source language. Uncompiled collection of user–created files containing FDL commands. See also FDL.
graphics	Use of lines and figures to display data, as opposed to using text.
grid	Imaginary pattern of evenly spaced horizontal and vertical lines on a page.
grid unit	Smallest rectangle enclosed by horizontal and vertical lines on a grid. The size of a grid unit is expressed as the length of one side of the rectangle.
hardcopy	Machine output in permanent form, such as printed reports, listings, and so forth.
hard values	Non-optimal adjustment of particular Font Interchange Standard (FIS) fonts in terms of point size and orientation.
hardware	Physical components of a system (processors, printers, terminals, and keyboards).
hexadecimal	Numbering system with a base of 16. The numbers 10 through 15 are represented by A through F, respectively.
hierarchy	Relative priority assigned to rank and order of items.
high–level language	Computer programming language consisting of symbols and English–like command statements that are easily understood and read by the user. High–level languages are considered the source code and are usually compiled. FDL and XFG are high–level languages.
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 a mainframe.
host interface	Connection between the network and the host computer.
IDEN	Parameter that notifies the system that DJDE records may be part of the input data stream.
identifier (ID)	Characters used to identify or name data and possibly to indicate certain properties of that data.
image area	Area on a physical page that may contain text or graphics.
input	Data or text introduced into a computer–based system.

input devices	Keyboards, magnetic media, or any device used to give a system information.
input output (I/O)	General term encompassing the flow of data into and out of a computer system.
input processing	Formatting control for the pages of a report.
insert	To add text or graphics to a document.
interface	Device by which two systems connect and communicate with each other.
Interpress	Xerox standard that defines digital representation of lines for printing. You can print Interpress documents on any sufficiently powerful printer equipped with Interpress print software.
JCL	Job control language. High–level language used with the MVS or DOS operating systems to identify job requirements such as user name, program name, CPU processing time, input/output devices, and files needed for batch processing.
JDE	Job descriptor entry. Collection of job descriptions. See also job.
JDL	Job descriptor library. Collection of compiled job descriptions. See also job source library (JSL).
job	Set of instructions (JDEs) defining a unit of work for the system.
jobcard	First JCL statement in a job, identifying the beginning of the job, the user, the job name, and similar information.
job control	Program called into storage to prepare each job or job step to be run.
job management	Collective functions of job scheduling and command processing.
JSL	Job source library. Collection of uncompiled job descriptions. See also JDE; JDL.
keyword	Reserved word that is essential to the meaning and structure of a statement. Required part of a command that must be entered in the command language syntax exactly as it is written. In XPRM, keywords can be written in uppercase or lowercase letters.
label	1. In data storage, a reference to a file saved on tape or disk, a record indicating the file name or date created, as well as other control information. 2. In programming, a name assigned to a particular instruction or portion of a program as a location reference. The computer translates the label into a locatable address.
landscape	Orientation of print lines or top of an illustration parallel to the long edge of the paper. See also orientation.
language	Defined set of characters and symbols combined together by specific rules. Used as an operator interface with the computers system software to perform certain tasks. See also high–level language; low–level language.
laser printing	Technology that uses a laser to scan and transfer character images onto a page by direct or indirect means.
leading	1. Vertical distance between lines (also called line space) measured from the baseline of one line to the baseline of the next. 2. Extra spacing between lines of type. 3. In typography, spacing between lines and paragraphs.
legal size	Page the standard size of legal briefs, 8.5 by 14 inches.

library	In data storage, a collection of related files or programs.
line	One horizontal flow of characters.
line feed	Control character that causes the printing system to begin printing in the current character position of the next line (unless set to be interpreted as an end of line).
line printer	High–speed printer that prints an entire line of characters at the same time.
line tables	Internal data structures that provide a record in memory of the line to be drawn on a page.
listing	Printout or display of the statements in a program, usually used as a convenience in examining or editing programs.
literal	Any alphanumeric character or symbol used to define itself rather than something it might represent.
load	To enter data into storage or working registers.
logo	Small illustration, signature, or design, typically used to identify a company.
log off	Procedure used to end a computer session.
log on	Procedure used to begin a computer session.
low–level language	Computer language that is close to machine language so each written instruction in a program can be converted to machine language on a one–to–one basis.
LPS	Laser printing system. Technology that uses a laser to transfer character forms to a page.
machine language	Binary code that can be read directly by the central processing unit (CPU) of a computer. All programming language is eventually converted into machine language for execution by the computer. See also computer language.
mainframe	The central processing unit (CPU) of a large computer. Most often used to denote any large computer that is used as the central source of computing for remote dumb terminals or interconnected micro– or mini–computers.
margins	White space on the top, bottom, and each side of the printed text.
media	Vehicles or devices on which information is stored or transmitted.
MB (megabyte)	1,048,576 bytes.
menu	List of available functions, commands, or options.
merge	Operation combining two or more items into a predetermined order. For example, merging two separate alphabetically ordered lists into one large alphabetically ordered list.
mnemonic	Word that possesses qualities helpful to human memory. Software commands are often rooted in mnemonic terms or abbreviations to make it easy to associate the functions with the name or keys pressed to implement the function.
mode	Manner in which an activity or process is carried out.
module	Cohesive unit within a program.
monitor	1. Program used to supervise, control, or verify the operations of a system. 2. A monochrome or color cathode ray tube (CRT) used to display computer information to the operator.

MVS	Multiple Virtual Storage. Operating system that supports the IBM System 370 hardware. MVS is designed to handle large scale production batch applications.
national use characters	Includes the @, $, and # sign. The system recognizes the hexadecimal representations of the U.S. national characters: @ as X'7C'; $ as X'5B'; and # as X'7B'. In some countries, the keyboard–generated characters may generate different hexadecimal representation and cause errors.
nonimpact printer	Printer that forms characters or symbols on a medium (usually paper) without striking or impacting the medium. Laser printers are nonimpact printers.
object file	Source file converted into machine language (binary code) that is usable by the CPU.
operand	Data that is acted upon (for example, in a computer operation or process).
operating system	Basic host– or LPS–resident controlling program that governs the operations of a computer, such as job entry, input output (I/O), and data management.
orientation	Direction in which text or images are positioned on a page. When text and images are positioned with the lines of text parallel to the shorter sides of the paper, the orientation is portrait. When text and images are positioned with the lines of text parallel to the longer sides of the paper, the orientation landscape.
origin	References a point on the image area of a page that is usually in the upper left corner of the page.
OSS	Operating system software. See also operating system.
overprinting	Printing more than one character at the same position.
overstrike	Printing characters over each other.
page	1. In computer programming, a block of instructions, data, or both, that can be located in main or auxiliary storage. 2. In word processing or form generation, a defined section of a document.
page end	Command character (form feed) that terminates the current page.
pagination	Process of separating text or data into pages.
parameter	Part of a command, other than the keyword. See also keyword.
PARM	Keyword parameter in the JCL that is coded on the EXEC statement of the JCL after the program or procedure name and only applies to the step being executed.
parsing	To read or interpret a command. To build up a parameter list from information within a command.
physical page	Area on which printing is done.
pica	1. Unit of measurement equal to 0.166 inch. Picas are often used to express line measurements or column width. 2. A type size that measures ten characters to the inch. See also elite; pitch; point.
pitch	Horizontal character spacing; 10–pitch spacing is called pica (10 characters per inch); 12–pitch spacing is called elite (12 characters per inch). See also cpi.

pixel	Acronym for picture element. Smallest addressable point of a bitmapped screen that can be independently assigned color and intensity. Pixels are definable locations on a display used to form images. For graphics displays, more pixels generally provide higher resolution. Spots, dots, and pixels are used interchangeably.
point	Unit of measurement equal to 0.0139 inch. Points are always used to express type, size, and leading. There are 12 points to a pica and about 72 points to an inch. See also pica.
point size	Height of character set from the top of its ascender to the bottom of its descenders in units (points). Point size does not always include leading.
portrait	Orientation of print lines or the top of an illustration parallel to the short edge of the paper. See also orientation.
PDL	Print description language. Language used to describe printing jobs to an LPS. PDL describes the input (type, format, and characteristics), performs the processing functions (logical processing), and describes the output (type format, font selection, and accounting options).
printer	Output device that produces hardcopy printouts.
printout	Informal expression referring to almost anything printed by a computer peripheral device.
process	1. To perform a systematic sequence of operations, such as add, edit, or delete. 2. To produce a specific result by manipulating data.
processor	Applies to any system that is capable of receiving and performing operations upon data and supplying the results to those operations.
program	Complete set of instructions in a language compatible with the device to be used. A program directs a system to perform each operation at the right time in the proper sequence.
programmer	Person involved in designing, writing, and testing computer programs.
prompt	Message or symbol displayed on a system console requiring the operator to take action.
proportional font	Font containing characters that vary in width.
proportional spacing	Text in which each alphanumeric character is given a weighted amount of space is proportionally spaced. Such output has print–like appearance. Proportional spacing allows more space for wide characters and less space for narrow characters.
proportional type	Characters that vary in width.
purge	To delete data from a system.
query	Request for data or other information entered by an operator while the system is processing.
queue	List of documents waiting to be processed.
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.
record	Collection of data or words treated as a unit.

recovery	Act of overcoming a problem or error during processing. Typically, a specialized software recovery routine gains control and attempts to resolve the error without crashing the system.
remote access	Access to a central computer by terminals or devices geographically separated from the computer.
replace	Process of exchanging one portion of text or symbols with another. This process encompasses two functions: deleting the old and inserting the new.
resolution	Number of dots per inch (dpi). The LPS imaging system converts a character from digitized data into a printed image composed of these tiny dots. The greater the number of dots per inch, the higher the resolution and the clearer the image. The terms dots, spots, and pixels are synonymous.
resource	File that is stored in a library (in machine–readable format) and used during the printing of a document. Each file holds a different type of resource; for example, forms or fonts.
RSTACK	Parameter that defines a series of reports in a single file. **save**
	To store programs, data, or text for retrieval at a later time. **scale** To
	adjust font size according to given proportions.
simplex	Printing on one side of a page. See also duplex.
software	Programs, including operating systems, procedures, utilities, and application programs written for a computer system. Software can be supplied by the hardware manufacturer or other firms but does not include programs written by the user.
sort	To rearrange data records according to a particular order. This order is generally selected by character (alphabetic, numeric, or special symbol) and can be ascending or descending.
source file	File containing source language statements or commands.
source language	High–level or low–level language that is used or created by a programmer. A source language must be converted by a compiler into machine language (object language) for instruction to be executed.
source program	Program written in a source language, generally a high–level programming language.
space	Blank area between words, recognized as a character by word and data processing systems.
spooling	Process of releasing data from main memory and storing it temporarily until it is ready to be accepted by the peripheral device. For example, storing text before sending it to a printer.
spot	Unit of measurement representing a fraction of an inch, for example, 300 spots per inch (spi). Spots may also be referred to as a picture element (pixel) or dot.
statement	Detailed written instructions in a program step. A statement is written according to specific rules called syntax.
string	Connected sequence of alphanumeric characters treated as one unit of data by a program.
submit	Term used when sending a JCL to the computer.
symbol	Characters used in a computer language to specify a particular function.

syntax	Rules governing the structure of expression in a programming language.
syntax error	System response to a mistake in structure of expression.
system	1. In data processing, a collection of parts and procedures organized to accomplish a set of specific functions. 2. Assembly of components united by some form of regulated interaction to form an organized whole. 3. Operation or procedures through which a business activity is accomplished.
system disk	Magnetic disk storage medium, usually of large capacity, which is not removable, as opposed to removable floppy disks or disk packs.
sysgen (system generation)	Process whereby the system is made ready to operate. Typically involves selecting the operative parameters and activating the relevant software.
system page	Maximum area in which text and graphics can be imaged on a printing system.
system software	Software programs that support and control system functions by governing hardware operation and input and output process. This includes interpreting source programs and converting them into machine language, as well as distributing tasks among various processors.
tape	Recording media for data or computer programs. Tape can be in a permanent form, such as perforated paper tape, or a reusable magnetic form. The magnetic form has a higher storage capacity than disk storage, but it takes longer to write or recover data from tape than from disk.
task	1. Any major job performed by a computer. 2. One of several programs being executed by a system.
terminal	Device equipped with a keyboard and connected to a computer or network.
testing	1. Process of running a program for the express purpose of discovering errors. 2. For computer–oriented systems, the process of verifying the ability of a system to meet performance objectives in a simulated environment, or validating its actual performance in a real–time environment.
text	Portion of a message, program, or input that contains the information you want to convey.
text string	Consecutive series of characters that are output exactly as specified in the command or program.
translation	Function of a language processor that converts certain source programs, such as Basic, into a machine–readable language.
transmit	To send data from one place to another.
truncate	Cut off before completion, as when data transferred from a host to a printer is cut off before all data is transmitted.
TSO	Time Sharing Option. Host software that allows multiple terminals to communicate efficiently with a host. TSO command processors can edit data sets, submit jobs for batch execution, and perform data set manipulation.
typeface	1. Type of a single design. 2. Set of character with design features that make them similar to one another.

type size	Height of a typeface expressed in points (measured from the bottom of its descenders to the top of its ascenders).
type style	Italic, condensed, bold, and other variations of typeface that form a type family.
UCS	Universal Character Set. Printer feature that permits the use of a variety of character arrays.
variable	Information of a changeable nature that is merged with a standard or a repetitive document to create a specialized or personalized version of that document.
variable data	Data that varies from page to page. Variable data is not a part of form design.
variable text	Text that varies from page to page. An example would be various names and addresses combined with a form letter to make a complete document.
virtual page	Page area selected by a forms designer for printing.
x–axis	Horizontal axis on a form grid.
xdot	Unit of measurement representing a fraction of an inch. Xdots may also be referred to as a picture element (pixel) or spot.
XDPM	Xerox Decentralized Print Mode or Xerox Escape Sequence (XES). See also XES.
XES	Xerox Escape Sequence. See also XDPM.
XPPM	Xerox Production Print Mode.
XPRM	Host–resident utility providing an interface between printer resources and the printer. XPRM allows you to store and manage resources generated by multiple applications in a central repository.
y–axis	Vertical axis on a form grid.

F. Index