Print Services Facility/MVS

Application Programming Guide

Version 2 Release 2.0

S544-3673-03

Print Services Facility/MVS

Application Programming Guide

Version 2 Release 2.0

S544-3673-03

Note!

Before using this information and the product it supports, be sure to read the general information in “Notices” on page xi.

Fourth Edition (January 1997)

This edition applies to Version 2 Release 2 Modification 0 of the IBM Print Services Facility/MVS licensed program (5695-040) and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Be sure to use the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM Printing Systems Company welcomes your comments. For your convenience, a form for readers' comments is provided at the back of this publication. You can either send your comments by fax to 1-800-524-1519 or mail comments to:

INFORMATION DEVELOPMENT
THE IBM PRINTING SYSTEMS COMPANY
DEPARTMENT H7FE, BUILDING 003G
PO BOX 1900
BOULDER, COLORADO 80301-9191

If you prefer to send comments electronically, use one of the following methods:

- Internet: print_pubs@vnet.ibm.com
- Fax: 1-800-524-1519

Internet

Visit our home page at http://www.can.ibm.com/ibmprinters

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

References in this publication to products or services of IBM do not suggest or imply that IBM will make them available in all countries where IBM does business or that only products or services of IBM may be used. Noninfringing equivalents may be substituted, but the user must verify that such substitutes, unless expressly designated by IBM, work correctly. No license, expressed or implied, to patents or copyrights of IBM is granted by furnishing this document.

Trademarks

This publication uses the following terms, which are either trademarks or registered trademarks of the IBM Corporation:

Advanced Common Control Unit
Advanced Function Presentation
Advanced Function Printing
AFCCU
AFP
AIX
AIX/6000
Application System/400
AS/400
AT
Bar Code Object Content Architecture
BCOCA
BookManager
BookMaster
DisplayWrite
Extended Services
GDDM
IBM
Intelligent Printer Data Stream
IPDS
MO:DCA
MVS/ESA
MVS/SP
Operating System/2
OS/2
OS/390
OS/400
Personal System/2
Print Services Facility
PrintManager
PSF
PSF for AIX
PSF/2
RACF
SP
System/370
System/390
S/370
VM
VM
VSE
VTAM

Summary of Changes

This edition contains the following new or changed information:

- Support for a 4-digit year designation on separator pages.
- Support by the IBM 3130 for font capture of single-byte raster and outline fonts.
- Support for printer-resident double-byte outline (scalable) fonts.
- Use of the first inline form definition, page definition, or microfilm setup resource without having to specify them in your JCL.
- Replacement of several page definitions with page definitions that can handle characters with descenders on the last line of the page.
- Support for AFCCU continuous forms printers (3900 wide, 3900 wide duplex, and 3900 duplex), with the appropriate level of microcode, to support up to 32 511 overlays and page segments per sheet. For more information, see your Printing Systems marketing representative.
- Support for using a specific single-byte or double-byte font for a line or a field as well as for an entire page. To specify a specific font for a line or field, you must specify them in the font list in the page definition.
- Support for specifying a bar code object in a page definition. PPFA APAR PN79369 is required for this support.
- References to PPFA/370 have been changed to PPFA. PPFA is a single product that runs on AIX, MVS, OS/390, OS/400, VM, and VSE operating platforms.

PSF also has new features that might be useful to you:

- IP PrintWay, which transmits output data sets from the JES2 or JES3 spool to printers in TCP/IP networks. For additional information, refer to *IP PrintWay Guide*, S544-5379.
- MVS Download, which automatically transmits MVS system output to AIX operating systems in the TCP/IP network for printing or archiving by Print Services Facility for AIX and OnDemand for AIX. For more information, refer to *Print Services Facility/MVS: MVS Download Guide*, G544-5294.
- NetSpool, which you can use to print output from VTAM applications, such as Customer Information Control System (CICS) or Information Management System (IMS), on any printer supported by your MVS system, including printers supported by PSF/MVS. For more information, refer to *IBM NetSpool Guide*, G544-5301.

For more information about these features, consult your Printing Systems Company marketing representative.

Chapter 1. Introducing Print Services Facility/MVS

This publication provides information about using the Print Services Facility/MVS (PSF/MVS) (licensed program number 5695-040); hereafter referred to as PSF.

This publication is written on the assumption that you have experience with application programming and with Advanced Function Presentation (AFP) printers. In this publication, the word *printing* refers to presentation on paper, foils, labels, or microfilm.

What is PSF/MVS?

PSF is an IBM licensed program that provides a link between application programs and presentation devices such as paper or microfilm printers.

PSF can be used either in deferred-printing mode, as the output writer or in direct-printing mode, as the access method for processing an output data set. Printers attached by means of Systems Network Architecture (SNA) cannot be used in direct-printing mode.

A page printer can print at any addressable point on a page. This all-points-addressable (APA) printing enables you to mix a variety of type sizes and styles, as well as images and electronic forms, on a single page. Page printers differ from line printers in that a page printer receives an entire page of data before printing any data, whereas a line printer receives 1 line of data at a time, and prints each line as it is received. PSF, however, can process line data so that page printers can print it; therefore, a job prepared for printing on a line printer can be printed on a page printer with little or no change to the application. You can enhance the printing of line data by using electronic forms (overlays), typographic fonts, and other advanced-printing functions.

Understanding Advanced Function Presentation on Microfilm Devices

PSF/MVS 2.2.0 with APAR OW10067 adds support for sending Advanced Function Presentation data to microfilm[1] devices. You can use the same data set for printing jobs on paper or microfilm by specifying a parameter in your Job Control Language (JCL). For information on the JCL keywords, see “Specifying JCL Parameters for Microfilm Jobs” on page 118. One physical AFP file can have multiple output destinations. Job Entry Subsystem (JES) does not purge the spool file until all outputs are satisfied. Because of this, separate OUTPUT statements can be designated, for instance, 1 for paper and 1 for microfilm, for the same print job on the JES spool. Your system programmer will provide the system setup.

Note: You cannot send AFP jobs to a microfilm device attached through the Distributed Print Function (DPF) of PSF. A microfilm device cannot be driven in direct-printing mode.

[1] *Microfilm* can mean either microfiche or 16mm film.

PSF as an Output Writer: Deferred-Printing mode

In deferred-printing mode, PSF is the output writer that processes the spooled output from JES and sends a data stream to a page printer. All the printers supported by PSF can print in deferred mode, and PSF can process data for 1 or more page printers at a time.

In deferred-printing mode, you are not using PSF directly. Rather, JES sets up, starts, and controls each output writer and its system output devices. Output from an application program is spooled to the Job Entry Subsystem (JES), and printing is deferred until JES schedules PSF to print the spooled output data set. The components involved in generating formatted output on a system output printer are shown in Figure 1.

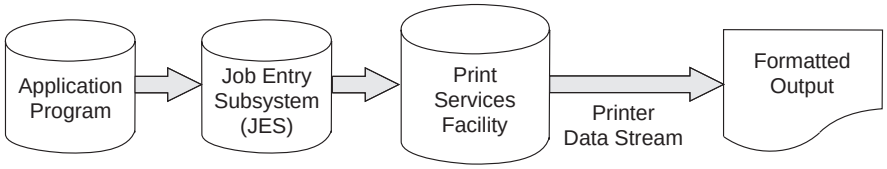


Figure 1. PSF as an Output Writer

Using information from JCL parameters and from the formatting specifications, PSF processes the print data set to generate the data stream from which the printer produces formatted output. For information about using JCL parameters to control the output, see Chapter 2, “Using JCL for Advanced Function Presentation.”

PSF as an Access Method (Direct-Printing Mode)

In direct-printing mode, PSF acts as an access method for the printer. PSF devotes the printer exclusively to the job, and the output is printed immediately. The components involved in direct-printing mode are shown in Figure 2.

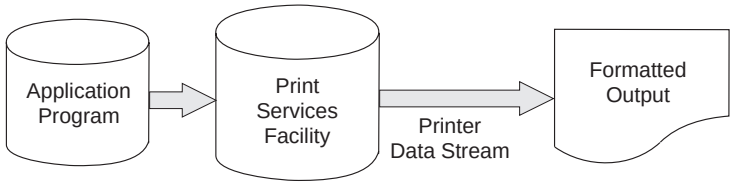


Figure 2. PSF in Direct-Printing Mode

Only channel-attached printers can print in direct-printing mode. (Microfilm devices cannot be attached in direct-printing mode.)

PSF uses JCL parameters for direct-printing mode, but these parameters are different from those used in deferred-printing mode. Direct-printing mode does not support the following deferred functions:

- User libraries
- System-assisted restart
- PSF repositioning
- Data-set checkpointing
- Multiple data-set processing
- JES operator commands to control the printer
- Job header and trailer pages
- Data-set header pages
- Carrier-strip marking
- Mandatory page labeling
- System Management Facilities (SMF) type 6 processing
- Notification of print completion
- Restartable abends
- Redirection of message data sets

For more information about direct-printing mode, refer to *Print Services Facility/MVS: System Programming Guide*.

Understanding Advanced Function Presentation (AFP) Output

AFP output can contain text data, image data, graphics, bar codes, or a combination of these data types on a single page.

Figure 3 shows a page printed on an AFP printer by using PSF. This page shows examples of the kinds of data described in the following sections.

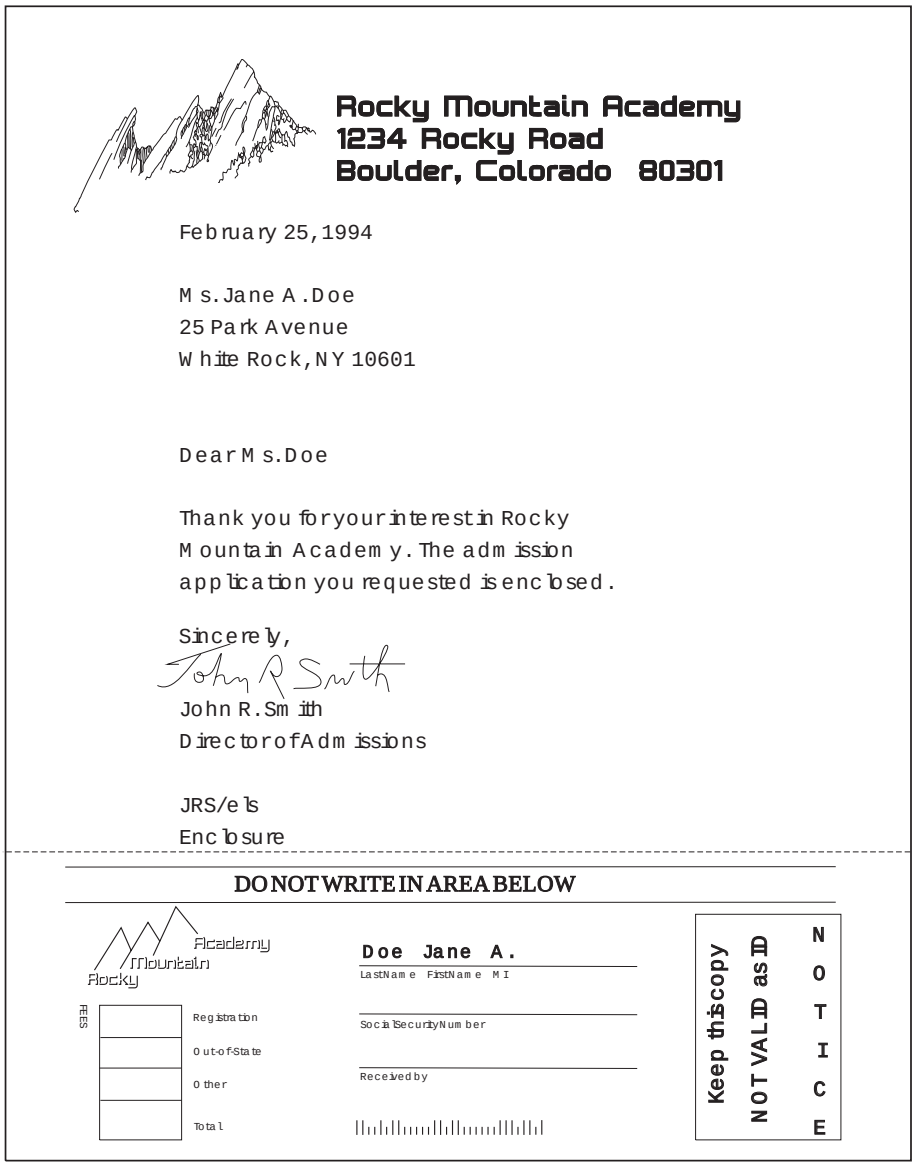


Figure 3. Sample Page Printed on an AFP Printer

All-Points Addressability

AFP printers can address all the picture elements (pels) on a page. The pel density is a major factor in determining the quality of a particular printer's output. The page printers supported by PSF have a variety of pel densities. To learn about the pel density[2] of your printer, refer to *Advanced Function Presentation: Printer Information*.

PSF sends the printer the address of a pel, along with the data to be placed there. All-points addressability enables you to send data to the page printer in any sequence and to position it anywhere on the printable area of the page.

For some print jobs, PSF must select addresses from formatting specifications. For others, addresses are already contained in the print jobs when PSF receives them.

Logical Unit Conversion

PSF accepts text, images, graphics, and bar code objects with L-unit base values of 14 400 units per 10 inches or 2400 units per 10 inches. PSF converts the input L-unit values to values supported by the printer, except where complexity and performance considerations prevent the conversion. This conversion can affect processing performance.

Text on a Page

AFP output can contain *text* data. Text is character data in the form of letters, numbers, punctuation marks, special characters, and ligatures. Character data is printed by use of *fonts*. A font is a collection of graphic characters sharing a single type family, type style, type weight, and type size. You can use several different fonts on a page. Two basic font technologies, raster and outline, are used to print text; they are described on page 21. IBM fonts are shipped on the AFP Font Collection CD-ROM. For more information about AFP fonts, refer to *IBM AFP Fonts: Font Summary*.

Figure 4 on page 6 shows the different fonts used on the sample page.

[2] Pel density is also called print-head resolution, or resolution.

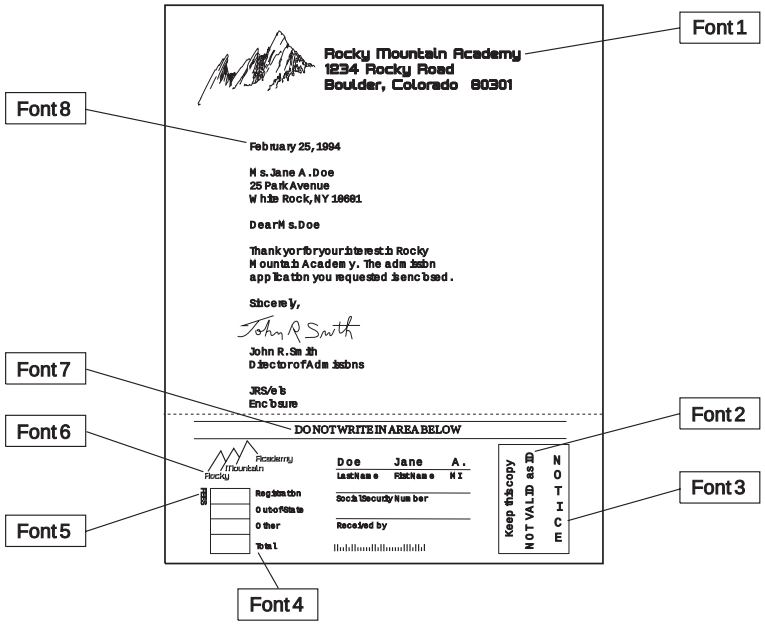


Figure 4. Sample Page of Text or Character Data Printed in Different Fonts

Images on a Page

AFP output can contain *image* data, which is a series of picture elements (pels) arranged in rows and columns. Another name for an image is a *raster pattern*.

Image data is created either by a scanning device or by a program, and is stored in a set of MO:DCA-P structured fields called an *image data object*. The image data can be included in a print data set, in an overlay resource, or in a page segment resource. PSF accesses resources and sends them to the printer with the print data set. For more information, see “PSF Resources” on page 17.

Figure 5 on page 7 shows the images printed on the sample page.

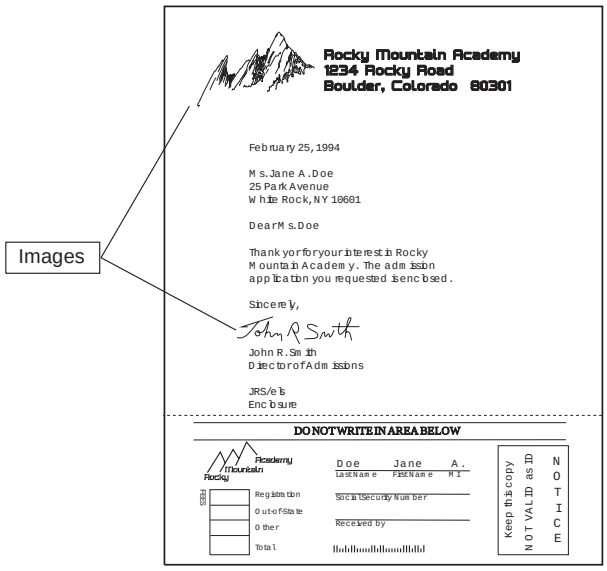


Figure 5. Images on the Sample Page

PSF supports image data objects stored in 2 different formats:

- IO image, which is also called an IOCA image, where IOCA stands for Image Object Content Architecture
- IM image, which is also called a basic image

For more information about coding IO and IM image data objects and the options that you can specify, refer to *Mixed Object Document Content Architecture Reference* or to *Image Object Content Architecture Reference*.

Note: When using images on a microfilm device, see Appendix D, “Microfilm Device Considerations.”

IO Images

An IO image data object specifies the content of a raster image and its placement on a page. The IO image format offers more function than the IM image format. The IO image format offers several options:

- An IO image can be placed in any position on the page.
- An IO image can be stored and transmitted to the printer in compressed form, saving storage and transmission time.
- An IO image can be resolution-independent, so that an image created with 1 resolution (that is, pel density) can be printed at the same size on a printer with a different resolution.
- An IO image can be scaled to any size (that is, enlarged or reduced) by the printer.
- An IO image can be rotated by the printer.
- An IO image can be clipped so that only part of it is printed.

PSF provides a MO:DCA-P document called IOCAMMR. IOCAMMR contains a compressed IOCA image compressed by use of the IBM MMR compression algorithm. You can print this image on any IPDS printer that supports IOCA. You can use this document as a test file or print it in hexadecimal to assist an image

application developer in understanding the structure of a MO:DCA-P document that contains an IOCA image object. If an image has been compressed by use of the CCITT Group4 compression algorithm, you would code it similarly.

Compatibility among Printers: Most of the newer printers support IO image data objects. To determine whether your printer does so, refer to *Advanced Function Presentation: Printer Information*.

Ordinarily, IO image data objects print only on printers that support IO image data objects. With PSF, however, you can print some IO image data objects even on a printer that supports only IM image data objects. The following restrictions apply:

- The resolution of the IO image must be the same as that of the printer.
- The IO image data must be stored in uncompressed format.
- The IO image data object must contain only 1 segment.
- The IO image data object must specify 1 of the following mapping options (these mapping options do not require scaling or correction of resolution):
  - Image point to pel
  - Image point to pel with double-dot
  - Scale-to-fit, with the input image space the same as that of the output space

When printed on a printer that supports IO image data objects, an IO image can be resolution-independent. That is, the printer prints the image at the same size, even if the resolution of the image is different from the resolution of the printer. Because of differences in the scaling algorithms used by various printers, however, exact fidelity of the image is not guaranteed. To be resolution-independent, an IO image data object must specify an option that causes the printer to correct the resolution. To learn about the mapping options that cause the resolution to be corrected, refer to *Advanced Function Presentation: Printer Information*.

IM Images

An IM image data object specifies the content of a raster image and its placement on a page. IM images offer several options:

- An IM image can be placed in any position on the page.
- An IM image can be enlarged to twice its size. This is the double-dot function.

Compatibility Among Printers: All the printers supported by PSF support IM image data objects. However, an IM image may or may not print at the same size on a printer with a different resolution; it depends on whether the printer supports IO image data objects, as follows:

- If the printer does not support IO image data objects, IM images created with 1 resolution will shrink or expand.
- If the printer does support IO image data objects, PSF transforms the IM image data object into an IO image data object, taking advantage of the resolution independence available with IO image data objects. Thus, the IM image will print at the correct size. Because of differences in the scaling algorithms used by various printers, however, exact fidelity of the image is not guaranteed.

Note: If an IM image data object specifies the double-dot option, PSF does not transform an IM image data object into an IO image data object; in this case, the image shrinks or expands when printed.

Graphics on a Page

AFP output can contain *graphics* data, which is also called vector representation or vector-graphics data. Graphics data contains commands to draw lines, arcs, and circles and can be used to represent something as complex as a three-dimensional engineering drawing.

Graphics data is created by a program and stored in a set of MO:DCA-P structured fields called a *graphics data object*.[3] A graphics data object can be included in a print data set, in an overlay resource, or in a page segment resource. PSF accesses resources and sends them to the printer with the print data set. For more information, see "PSF Resources" on page 17. For information on the data format for AFP graphics objects, refer to *Graphics Object Content Architecture Reference*.

Figure 6 shows the graphics data printed on the sample page. As the figure shows, a graphics data object can contain character data as well as graphics data. However, all fonts used must be active in the print data set or overlay that includes the graphics data object.

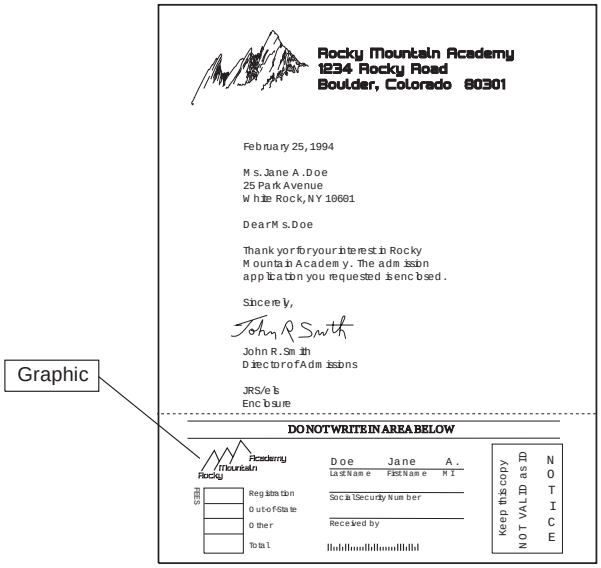


Figure 6. Graphic on the Sample Page

Note: When using graphics on a microfilm device, see Appendix D, "Microfilm Device Considerations."

[3] A graphics data object is also called a GOCA object, where GOCA stands for Graphics Object Content Architecture.

Graphics data objects offer several options:

- The graphics data can be placed in any position on the page.
- Graphics data can be scaled to any size; that is, enlarged or reduced, by the printer.
- Graphics data can be rotated by the printer.
- Graphics data can be clipped so that only part of the graphic is printed.
- A graphics data object can include MO:DCA text data.

For more information about how to code graphics data objects and what options can be specified, refer to *Mixed Object Document Content Architecture Reference*.

Compatibility Among Printers: Most of the newer printers support graphics data objects. To determine whether your printer does so, refer to *Advanced Function Presentation: Printer Information*.

Because graphics data is stored in vector representation, graphics data is always resolution-independent. That is, graphics data can be printed at the same size on any printer that supports graphics data objects, regardless of the resolution (that is, the pel density) of the printer.

Bar Codes

Bar codes represent characters by sets of parallel bars of differing thickness and separation that can be read optically by transverse scanning. These codes can represent, for example, product numbers, part numbers, and manual numbers.

Figure 7 shows the bar code printed on the sample page.

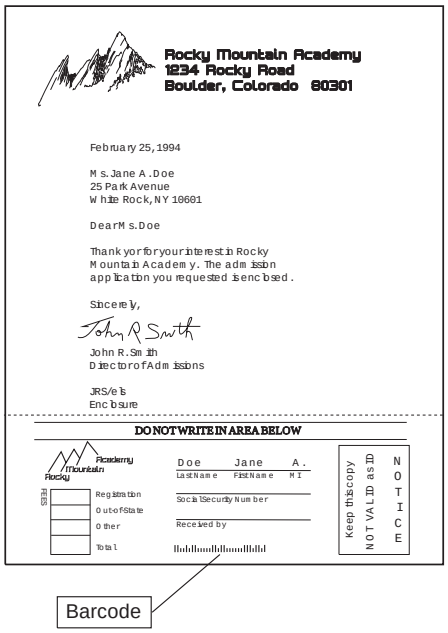


Figure 7. Bar Codes on the Sample Page

Many different kinds of bar code coding arrangement, or *symbologies*, have been developed for specific applications. Some of these codes are:

- Code 39 (Code 3 of 9)
- EAN 8 and EAN 13
- 2 of 5 Codes: Industrial, Interleaved, Matrix
- MSI/Plessey
- UPC A and UPC E
- Codabar
- Postnet

Bar codes can be printed on a PSF-supported printer in several ways:

- By using a font that contains symbols for printing bar codes
- By using MO:DCA-P structured fields to draw horizontal or vertical rules to represent the bar codes
- By using a MO:DCA-P bar code data object to instruct the printer to create the bar code

Using Fonts to Produce Bar Codes

Just as you can use AFP fonts to print symbols for alphanumeric characters, you can use them to print symbols for bar codes. IBM supplies several font products that print bar codes, including the following products:

- 5688-021: Bar Code/OCR licensed program, which uses a PL/I subroutine to print different bar code symbologies
- 5799-DGX: an RPQ font for printing the Postnet symbology

The products are designed for 240-pel printers that support the downloading of host fonts.

Using MO:DCA-P Text to Produce Bar Codes

The MO:DCA-P Presentation Text Data (PTX) structured field contains codes that instruct the printer to draw horizontal or vertical rules of different lengths and thicknesses. If you know the layout of the bar code you are trying to print, you can write a routine that instructs the printer to draw rules of the correct dimensions and placement to represent the bar code. This method can be used on any printer supported by PSF.

The IBM Document Composition Facility (DCF) program uses this method to format bar codes. DCF can format several different bar code symbologies, enabling you to control characteristics such as the height, width, and spacing of the bar code elements. For more information about using DCF to produce bar codes, refer to *Document Composition Facility Bar Code User's Guide*.

Using MO:DCA-P Bar Code Objects

By using a set of structured fields called a *bar code data object*,[4] you can direct some PSF-supported printers to produce bar codes. A bar code data object specifies the type of bar code (the symbology), its size, and positioning information. Many different bar code symbologies can be produced by use of bar code data objects.

[4] A bar code data object is also called a BCOCA object, where BCOCA stands for Bar Code Object Content Architecture.

A bar code data object can be included in a print data set or in an overlay resource, or it can be created from bar code specifications in a page definition. PSF accesses resources and sends them to the printer with the print data set. For more information, see “PSF Resources” on page 17.

Bar code data objects offer several options:

- One of several bar code types (symbologies) can be selected.
- The bar code elements can be of any height and width, within the limitations of the symbology.
- The bar code can be printed in 1 of several colors (on printers that support printing in more than 1 color). However, you cannot specify color when you are specifying bar codes in a page definition.
- The bar code can be placed at any position on the page.
- The bar code can be rotated.

For more information about coding bar code data objects and about the options you can specify, refer to *Mixed Object Document Content Architecture Reference* or *Bar Code Object Content Architecture Reference*. For information about coding bar code specifications in page definitions, refer to the documentation section of PPFA APAR PN79369.

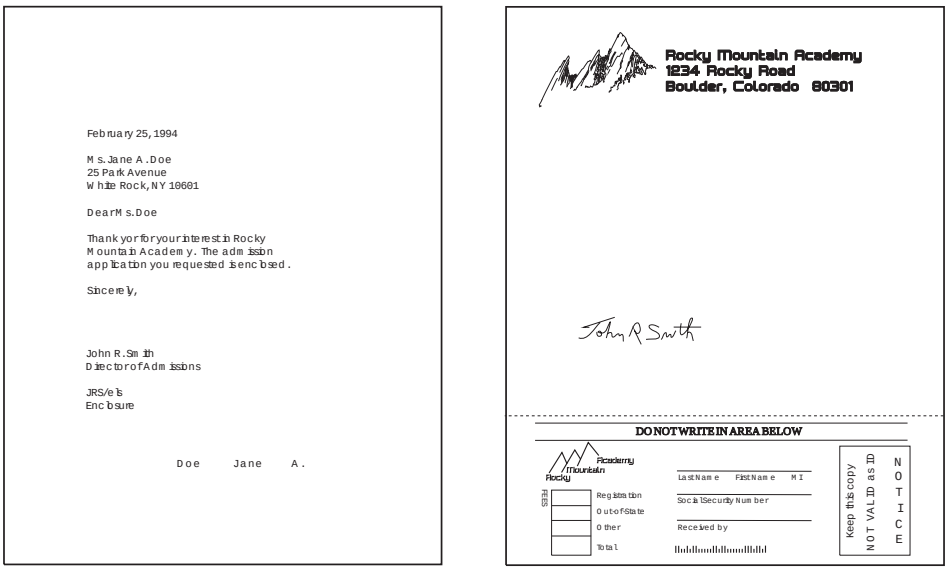
A bar code data object can be printed only on printers that support bar codes. Printers that support bar codes support different bar code symbologies and symbology parameters. Your printer may use default values for unsupported parameters; therefore, you should verify that the bar codes printed by your printer are suitable for your purposes. For information about the bar code support that your printer provides, refer to *Advanced Function Presentation: Printer Information* and the publications for your printer.

Bar codes generated on printers with different resolutions may differ in length because of the resolution correction needed to round the bar code widths to a specific pel size. Each printer may use a different algorithm for this calculation. Therefore, you need to test on each type of printer to be used to print bar codes.

Combining Character, Image, Graphics, and Bar Code Data

PSF can merge variable data from application programs with print resources and can print the data at any location on a page. PSF can merge text, image, graphics, and bar code data to create a composite output of different data types.

Figure 8 on page 14 is a sample page shown in 2 parts: character data and print resources (graphic, image, and bar code data) merged with it.



Character Data

Other Data

Figure 8. Sample Containing Character Data and Other Data

Merging Data with an Electronic Form

The ability to combine data with print resources enables you to use an application to merge data with an electronic form, or *overlay*. An overlay contains constant information and may also contain blanks that can be filled in by the application. The print data set created by the application contains the variable data that is printed on the electronic form. PSF can print both the variable data and the form on a single sheet, eliminating the need to preprint forms to print the variable data on. PSF can also print an overlay on a blank page containing no variable data. Figure 9 shows an electronic form that was used on the sample page. The application program supplied the name *Jane A. Doe* that is printed on the electronic form in Figure 4 on page 6.

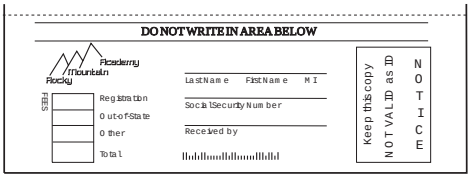


Figure 9. Electronic Form (Overlay) on the Sample Page

Printing Different Data at the Same Place on a Page

All-points addressability enables PSF to print different data *at the same place* on the page. Overlapping data in this manner does not produce bold print as it does with an impact printer. (To produce bold print with a page printer, you must use a bold font.) However, you can use the data-merging capabilities of PSF and page printers to do the following, shown in Figure 10 on page 15:

- Create composite characters by printing more than 1 character at the same location on the page, which is shown in Example 1 in the figure by the slash printed on top of the 0.

- Overlap characters so that parts of each character have a common area, which is shown in Example 2 in the figure by the asterisks (*) printed on top of the characters 200.00.
- Merge a graphics object with character data, which is shown in Example 3 in the figure by the mountain graphic printed over the name of the academy.

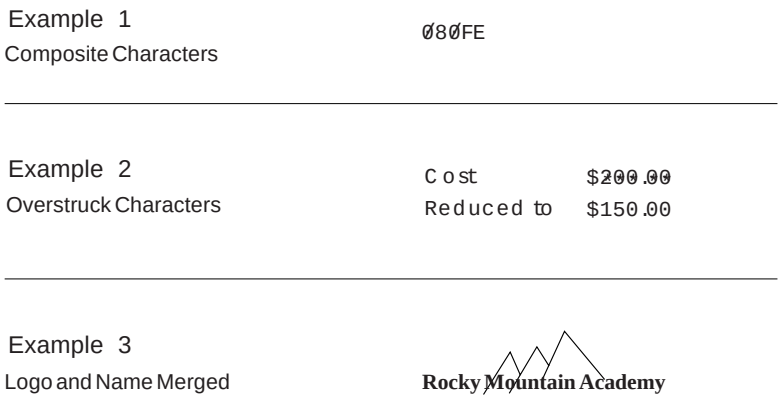


Figure 10. Printing 2 Items in the Same Location on the Page

PSF Data Streams

PSF converts the input data stream contained in a print data set to an output data stream that AFP printers use to place data on pages. PSF accepts 2 types of data as input from print data sets:

- MO:DCA data
- Line data

MO:DCA data is data that is already composed into pages. MO:DCA includes information about the placement and presentation of data (such as what font to use), along with the data to be printed. For more information about MO:DCA-P, refer to *Mixed Object Document Content Architecture Reference*.

Line data is data prepared for printing on line printers such as the IBM 3800 Printing Subsystem Model 1, the IBM 1403 Printer, or the IBM 6262 Printer. Line data does not contain any of the data placement or presentation information needed for printing on page printers.

The output of PSF is a printer data stream consisting of commands that instruct the printer how to print each page. For most page printers, the printer data stream is the Intelligent Printer Data Stream (IPDS). The IBM 3800 printers use a CCW printer data stream, which preceded the IPDS data stream. For more information about IPDS, refer to *Intelligent Printer Data Stream Reference*.

PSF can also accept data sets containing both MO:DCA data and line data. The following sections describe PSF processing for each of these data streams.

MO:DCA Data

The processing that PSF does to MO:DCA data is different from the processing that it does to line data. The reason is that whereas line data must be composed into pages, MO:DCA data is already composed into pages; it consists of structured fields containing commands, plus the data comprising the print job.

To generate MO:DCA output from an application program, you can use any of the following methods:

- You can use a text-formatting program that generates page data, such as Document Composition Facility (DCF).
- You can generate MO:DCA data in your application program.
- You can use the AFP Application Programming Interface (API) for COBOL or PL/I, which is supplied with PSF.

For more information, refer to *AFP Application Programming Interface: Programming Guide and Reference*. For information about generating MO:DCA data streams, refer to *Mixed Object Document Content Architecture Reference*. For information on printing MO:DCA data, see Chapter 6, “Printing MO:DCA Data.”

Line Data

To compose pages for the page printer from line data, PSF separates the incoming print records into pages according to specifications in a resource called a *page definition*. A page definition is always required for printing line data with PSF. You can create your own page definition or use a page definition provided with PSF. For more information about page definitions, see Chapter 3, “Understanding Form Definitions and Page Definitions.”

The line data input to PSF can consist of records that are fully formatted for printing on line printers; or it can consist of records that contain only the fields of data to be printed; or it can consist of records of both types. You can use the page definition resource, described in “PSF Resources” on page 17, to format fields of line data outside of the application program.

Figure 11 shows the difference between formatted and unformatted line text.

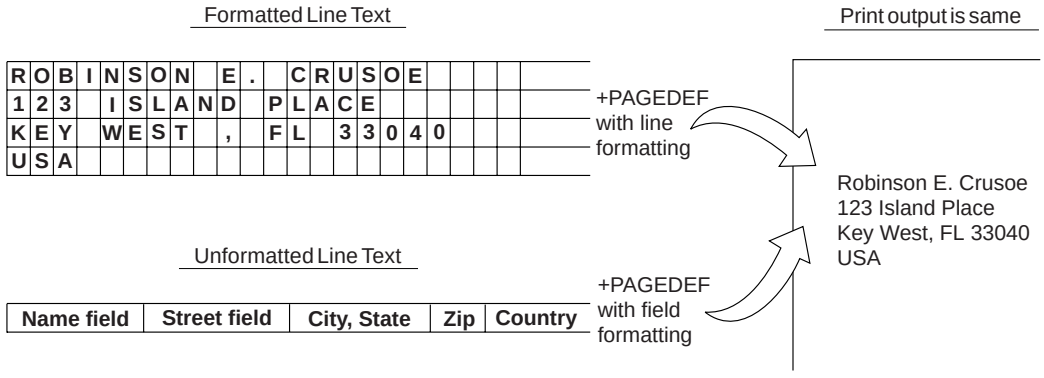


Figure 11. Formatted and Unformatted Text

Line Data with AFP Control Records

For more flexibility in formatting your line-data applications, you can include certain AFP control records in line data to change the formatting of selected pages within a data set or to include images or blocks of composed text on a page. For more information about how you can use AFP functions by including control records in line data, see “Including AFP Structured Fields in Line Data” on page 127. For a description of the structure of the control records that can be intermixed with line data, refer to *Mixed Object Document Content Architecture Reference* or *Advanced Function Presentation: Programming Guide and Line Data Reference*.

Mixed Data

In addition to MO:DCA data, line data, or line data containing AFP control records, PSF can process data sets containing both line data and data that has been composed into pages. If MO:DCA data is already formatted into pages, it cannot be printed on the same logical page as line data. When PSF finds the beginning or the ending of a MO:DCA data page, PSF starts a new page.

For a description of the structured fields used with MO:DCA data, refer to *Mixed Object Document Content Architecture Reference* or to *Advanced Function Presentation: Programming Guide and Line Data Reference*.

PSF Resources

To process a print data set, PSF may require 1 or more of the following resources:

- **Form definitions**, which specify the physical attributes of printed output.
- **Page definitions**, which specify how PSF is to format and compose line data into pages.
- **Fonts**, which are collections of graphic characters of a given size and style that print character data (text).
- **Page segments**, which can contain text and images and can be included anywhere on a page or an overlay.
- **Overlays**, which are collections of data that can be merged with the file being printed. Overlays are complete units rather than parts of pages such as page segments. Overlays can include page segments.
- **Object containers**, which is a general resource type. The only object container resource supported is microfilm setup resource, which specifies the attributes of AFP output on a microfilm device.

PSF manages the resources during printing. These resources can be stored in:

- **System libraries**, including security libraries. Each system library can be a concatenation of partitioned data sets that contain members for 1 or more kinds of resource. Usually, a resource is built by an application or by an AFP licensed program and is stored in a library for other print jobs.
- **Private user libraries** (only on MVS/ESA systems). A private print-resource library is owned by an individual user and is accessed only when an authorized job submitter specifies its name with the JCL USERLIB parameter.
- **Print data sets**, including data-set-level inline resources such as the following:
  - Fonts (coded fonts, code pages, and character sets)
  - Overlays
  - Page segments
  - Object container resources

  - Form definitions
  - Page definitions

  These resources are at the beginning of the print data set enveloped in a MO:DCA resource group structure. PSF uses the inline resources for that data set only; after the data set has been processed, PSF deletes the resources from the printer.

- **Printers**. Some printers store certain fonts as resident resources.
- **Distributed Print Function (DPF) libraries** on a personal computer. For more information about DPF, see publications for “Print Services Facility/2” on page 162.

When a resource is specified in a print data set, PSF searches for it first among the inline resources for the data set (if any), then in any specified user libraries, and finally in the system libraries. Printer-resident resources and resources in DPF libraries are used only after a marked host resource is found. For more information, see “Printer-Resident Fonts” on page 21 and “Using Resources with the Distributed Print Function (DPF) of PSF” on page 34.

Figure 12 shows the resources that PSF uses to process print files.

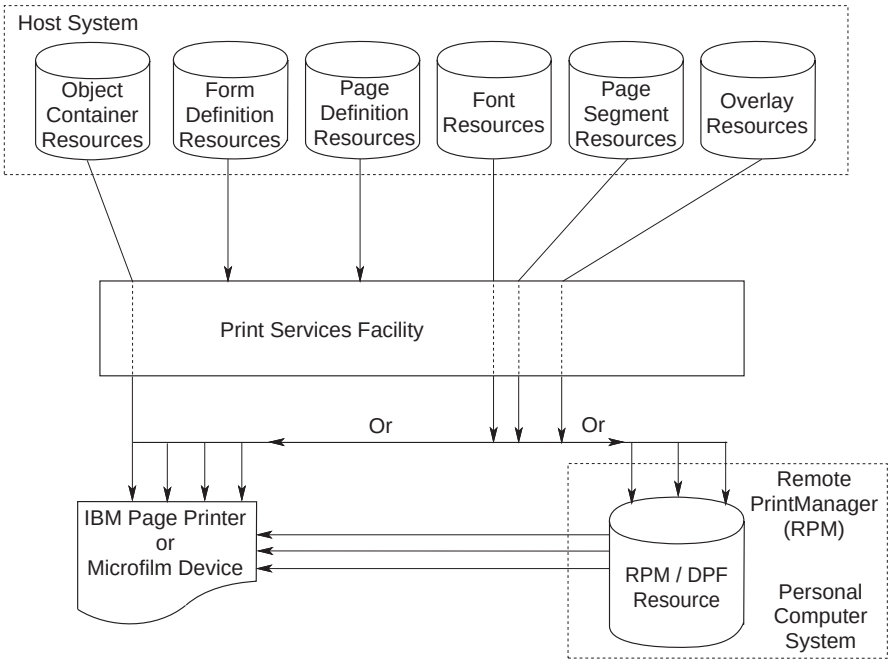


Figure 12. Resources for PSF

Resources found in system libraries, in user libraries, or inline are considered host resources, as distinguished from resident resources, which are in the printer or in an intermediate caching device such as DPF.

The resources that can be used depend on how the printer is set up, what licensed programs are installed, what libraries are assigned to each printer, and what data set is being processed.

When you select your printer by specifying JCL parameters, you are also selecting that printer's assigned system libraries.. Check with your system-support group for

what resources are available, what libraries the resources are in, and what libraries are assigned to each printer on your system.

Resources Supplied with PSF

IBM supplies the following resources with PSF:

- Form definitions, which are listed under Appendix A, “Form Definitions Supplied with PSF.”
- Page definitions, which are listed under Appendix B, “Page Definitions Supplied with PSF.”
- Fonts. IBM's *AFP Fonts* technical references describe the various fonts available for printers with different pel densities. Samples of fonts are shown in *IBM AFP Fonts: Font Summary*.

You can also use other IBM licensed programs to create and tailor PSF resources. For information about these programs, refer to *Guide to Advanced Function Presentation*.

Form Definitions

A *form definition* is the resource that specifies the physical attributes of the printed output. The word *form* refers to a sheet of paper or any other print medium.

You must specify a form definition for each data set you want to print. You can specify a form definition by name, or you can use the default form definition set up by your installation.

A form definition contains printing controls that specify the following, within the limitations of each printer:

- Page origin, which is the top-left boundary for printing. When the printer is duplexing, page origin may be different for the front and back of the page.
- Sheets on which medium overlays are to be printed.
- Sheets on which a forms flash (see the glossary) is to be printed.
- Printing with only overlays or forms flash and no variable data. This is the constant-forms function.
- Number of copies of each page to be printed.
- Paper source (input bin of the printer), for printers with more than 1 paper source.
- Simplex (printing on 1 side of a sheet) or duplex (printing on both sides of a sheet), for printers that support duplex printing.
- Data fields that are to be suppressed, that is, not printed.
- Printed copy groups to be stacked offset from each another.
- Page presentation in either portrait or landscape position.
- Print-quality level (on printers that support different levels of print quality).
- Horizontal adjustment, in pels.
- N_UP Printing: printing multiple logical pages on a sheet. These pages can be either MO:DCA pages (fully composed pages containing data and the structured fields controlling presentation of the data) or line data.

A form definition is required for every print job that you send to an AFP printer. PSF needs the form definition to position the logical page on the physical form. The form definition specifies the origin of the logical page as an offset from the origin of the physical form, or medium. The logical page is an area defined by the page definition for line data or by structured fields for MO:DCA data.

For more information about medium origin and logical page origin, see “Page Position” on page 68.

Note: When creating form definitions for a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Using Form Definitions

For information about form definitions, see the following sections:

- For more information about the function of a form definition in printing, see “Form Definitions” on page 67.
- For information about using a form definition, see “Specifying a Form Definition” on page 91.

Page Definitions

A *page definition* is the resource that specifies how PSF is to format line data into pages. PSF does not use a page definition for MO:DCA data, because that data is composed into pages before PSF receives it. The page definition replaces the forms control buffer (FCB) used by line printers.

You can specify a page definition by name, or you can use the default defined by your installation.

A page definition contains formatting information specifying the following:

- Page size (height and width)
- Print direction for the page of data
- Number of lines per inch
- Fonts to be used for printing the data
- Where data from each input record is to be printed
- Constant data to be printed
- Data fields that can be suppressed
- Print position for carriage control characters or channel codes
- List of page segments
- List of page overlays
- Conditional processing
- Color selection (for printers that support printing in multiple colors)

Using Page Definitions

For information about page definitions, see the following sections:

- For more information about the function of a page definition in printing, see “Page Definition” on page 80.
- For information about using a page definition, see “Specifying a Page Definition” on page 98.
- When sending constant data to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Fonts

A *font* is a collection of graphic characters sharing the same type family, style, and weight. You can use a font for an entire data set, for an entire page, or for selected lines or fields of data on a page. A coded font translates your input (keystrokes) into graphic characters for printing.

Note: When using fonts for a microfilm device, see Appendix D, "Microfilm Device Considerations."

Library-Member Types

An IBM font library is composed of 3 library-member types:

- A *coded font member* associates a code page and a font character set as a pair. A single-byte coded font contains 1 code page and font character-set pair. A double-byte raster coded font, which requires 2 bytes to identify each graphic character, contains 2 or more code page and font character-set pairs; each pair is called a font section. A double-byte outline font contains 1 code page and font-character-set pair.
- A *code page member* associates a code point and a graphic character identifier for each graphic character supported by the code page and specifies how code points that are not valid are to be processed.
- A *font character set member* contains a graphic character identifier and a raster pattern or outline for each graphic character in the font or font section, as well as information about how the characters are to be printed.

Before each page is printed, the fonts required for the page are sent to the printer (downloaded) if the printer does not already have them in its storage. Page printers can print fonts with various point sizes,[5] styles, weights, and widths on a single line or on various lines on a page. Multiple fonts can be printed on a page.

The printer storage required for a font depends on the size and number of characters in the font.

Printer-Resident Fonts

Some AFP printers, such as the IBM 382x printer, IBM 3835 printer, and the IBM 3900-001 printer, use *host fonts* that have been downloaded from PSF to the printer. Some AFP printers, such as the IBM 64xx, use only resident symbol sets, which are stored in the printer. Some AFP printers, such as the IBM 3930, the IBM 3935, and the IBM 3130, can use either host fonts or printer-resident fonts. Some AFP printers, such as the IBM 3130 printer, can capture host fonts as new resident fonts. For information about the fonts your printer supports, refer to *Advanced Function Presentation: Printer Information*.

If you are using a printer that supports only resident fonts, you are limited to printing with the font set stored on that printer. For information about the fonts available, refer to the publications for your printer or to *Advanced Function Presentation: Printer Information*.

If you are using a printer that supports both resident and host fonts, you may want to use resident fonts instead of the equivalent host font. PSF can access the

[5] A point size is a standard typographical measurement of the height of a font. One point is about 1/72 inch.

resident fonts in the printer instead of downloading them from the host font library, which saves transmission time. Your system programmer can identify or *mark* the host fonts by running the APSRMARK utility. For more information about the APSRMARK utility, refer to *Print Services Facility/MVS: System Programming Guide*, or see your system programmer.

A resident font should be the functional equivalent of the *marked* host version of the font. A host font that has been marked as PUBLIC by using the APSRMARK utility is not sent to the printer if a matching printer-resident font is stored in the printer. If a font is marked as PRIVATE, PSF sends that font from the library to the printer even if a matching printer-resident font exists; that is, PSF overrides the printer-resident font. If a font is not marked PUBLIC or PRIVATE, PSF assumes that the font is PRIVATE.

If you have the AFP font collection, and APAR OW08340 has been applied to PSF 2.2.0, most of the fonts are already marked PUBLIC, and you need not run APSRMARK to enable the printer-resident fonts.

If you update a font or create a new version of it by using an IBM AFP utility, the new version will not be marked PUBLIC, and it will be sent to the printer from host or user libraries. If, however, you bypass IBM utilities to directly update the resource object code for a marked font, without changing the APSRMARK stamp that marks it as PUBLIC, PSF treats it as a PUBLIC font and does not download it to the printer; instead, PSF uses the downlevel, printer-resident version. If you want to use the new version, you can mark the font PRIVATE or else mark it PUBLIC with a new RRDATE and RRTIME. Some printers can capture these modified fonts and make them resident, after they have been correctly marked.

Do not mark a metric-only font character set PRIVATE, and do not mark code pages used with double-byte outline fonts PRIVATE. Because a metric-only font contains no font pattern, it must be marked PUBLIC, and print jobs must use the printer-resident version of the font. For more information about the APSRMARK utility, refer to *Print Services Facility/MVS: System Programming Guide*, or see your system programmer.

You cannot use resident fonts for all print jobs. Some printers do not support resident fonts, and some printers do not support all the possible print-direction and character-rotation combinations for resident fonts. In addition, the characters in a printer-resident font may not match the characters in the host version of the font. Your system programmer can tell you which printer-resident fonts are available on each applicable printer in your installation. For more information about support for resident fonts, refer to *Advanced Function Presentation: Printer Information* and the publications for your printer.

Font Capture

Some printers, such as the 3130 printer, can capture downloaded fonts. The concept is similar to “dynamic caching” in DPF. Captured fonts automatically become new temporary “printer-resident fonts,” which improves performance for future jobs that use the same fonts. Both raster fonts and outline fonts can be captured. For more information on font capture, refer to *Advanced Function Presentation: Printer Information*.

Single-Byte Fonts and Double-Byte Fonts

Some page printers can print with both single-byte and double-byte fonts. A single-byte font can access up to 256 graphic characters from a character set. A double-byte font is used for printing languages whose base alphabets contain more than 256 graphic characters. To represent all the characters, more than 256 code points are needed. Therefore, 2 bytes are used: 1 byte to identify the section, and 1 byte to identify the code point in that section. An example of a writing system that requires double-byte fonts is kanji, in which the graphic characters are symbols in Japanese ideographic alphabets.

Double-byte raster fonts are available as a feature in the AFP Font Collection. Double-byte outline (scalable) fonts are available as RPQ 8A8080 product 5799-FCQ, "IBM AFP CJK Metric-only Fonts." This RPQ contains fonts that PSF needs to activate printer-resident versions of double-byte outline fonts such as Japanese Heisei Mincho or Korean Myengjo.

PSF provides some single-byte fonts, samples of which are shown in *IBM AFP Fonts: Font Summary*.

240-pel Raster Font Formats

The AFP fonts are provided as members of the system font libraries in the format required for input to PSF. The AFP fonts are provided in both bounded-box and unbounded-box format. A bounded-box format is designed to use bounded character boxes; that is, boxes whose design does not require that character-positioning information be included (as untoned pels) in a box. An unbounded-box format uses unbounded character boxes, which do require character-positioning information. Fonts in unbounded-box format are used only by the 3800 printer.

300-Pel Fonts

When you print on a printer with a *resolution* (pel density) of 300 pels per inch, you can use either printer-resident fonts or 300-pel host fonts that PSF downloads to the printer.

For printing with host fonts, PSF provides a set of 300-pel *expanded core* fonts, such as Courier, Helvetica, and Times New Roman. Your system programmer can also convert any single-byte 240-pel font in the bounded-box format to a 300-pel font by using the font-conversion program distributed with PSF. This program, APSRCF30, is described in *Print Services Facility/MVS: System Programming Guide*.

For printing with some printer-resident fonts, PSF provides host *metric-only* fonts for each printer-resident font. Metric-only fonts contain measurement information only; they do not contain raster patterns for the characters. Therefore, they cannot be used for printing and cannot be downloaded to a 300-pel printer. The advantage of metric-only fonts is that they require much less disk storage than standard host fonts.

Metric-only fonts are used by host programs such as Document Composition Facility (DCF) for positioning and spacing values when formatting text. They are also used by PSF as the host equivalent for the printer-resident font. These fonts should always be marked as PUBLIC, as is explained in "Printer-Resident Fonts" on page 21.

Outline Fonts

PSF/MVS 2.2.0, with APAR OW08340, supports Single Byte Character Set (SBCS) outline font technology, in which the graphic character shapes are represented in digital form by a series of mathematical expressions that define the outer edges of the strokes. The resultant graphic character shapes can be either solid or hollow. Each outline font character is defined by an algorithmic description of its shape, and, when combined with the font metrics (height, width, and space), is used by the printer to form the rasterized bitmaps at the printer's defined resolution. Outline fonts can be specified wherever raster fonts can be specified; for any reference to an outline font, however, a font size must be specified. The font size and the horizontal scale factor can be specified either in the MO:DCA data stream or in a coded font.

The font technology supported before APAR OW08340 is raster technology, in which each character is formed by pels in the shape of the character. For more details contrasting the 2 technologies, refer to *Font Object Content Architecture Reference*. For information on requesting raster fonts or outline fonts, see "Specifying Fonts" on page 84.

Figure 13. Example of Text Using Outline Fonts

PSF/MVS 2.2.0 with APAR OW14128 supports printer-resident double-byte outline fonts. These scalable fonts are available as a feature on some printers. Host versions of the fonts are available in the IBM AFP CJK Metric-only Fonts product, product number 5799-FCQ, RPQ number 8A8080.

Additional Fonts

Additional fonts are available on the AFP Font Collection CD-ROM. Samples of fonts are shown in *IBM AFP Fonts: Font Summary*, listed in Appendix E, "Related Publications." You can create additional fonts from Adobe Type 1 fonts by using the Type Transformer product shipped with the font CD. Refer to *IBM AFP Fonts: Font Summary*.

Fonts for Different Print Directions and Character Rotations

Print direction is the combination of inline and baseline directions. Text can be printed in 4 print directions. For each of the directions, characters can be printed in 4 rotations.

The *inline direction* is the direction in which successive characters are added to a line of text. The 4 inline directions are:

- **ACROSS**: Text characters are placed in a line from left to right across the page.

- **DOWN**: Text characters are placed in a line from top to bottom down the page.
- **BACK**: Text characters are placed in a line from right to left across the page.
- **UP**: Text characters are placed in a line from bottom to top up the page.

The *baseline direction* is the direction in which successive lines of text are added to a page.

The 4 character rotations for each inline direction are 0°, 90°, 180°, and 270°, measured clockwise around each inline direction. For example, the text in this paragraph is printed *across* the page, and its rotation is 0°. Figure 14 shows the 16 possible combinations of these inline directions and character rotations. For information about the combinations supported by the printer you are using, refer to *Advanced Function Presentation: Printer Information*.

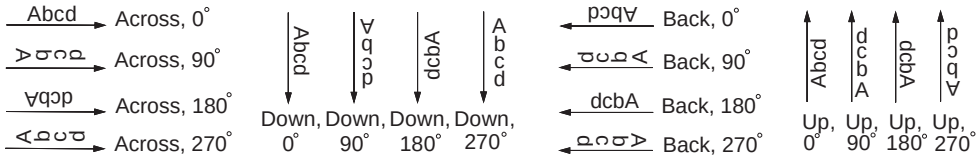


Figure 14. Direction-Rotation Combinations

PSF can process line data so that it can be printed in each inline direction with any character rotation. Figure 15 shows 4 copies of a memo, each printed in a different inline direction with 0° character rotation. The memo was reformatted for each direction.

Pages are said to be printed in *portrait* page presentation when the shorter edges of the paper are the top and bottom of the page, and pages are said to be printed in *landscape* page presentation when the longer edges of the paper are the top and bottom f of the page.

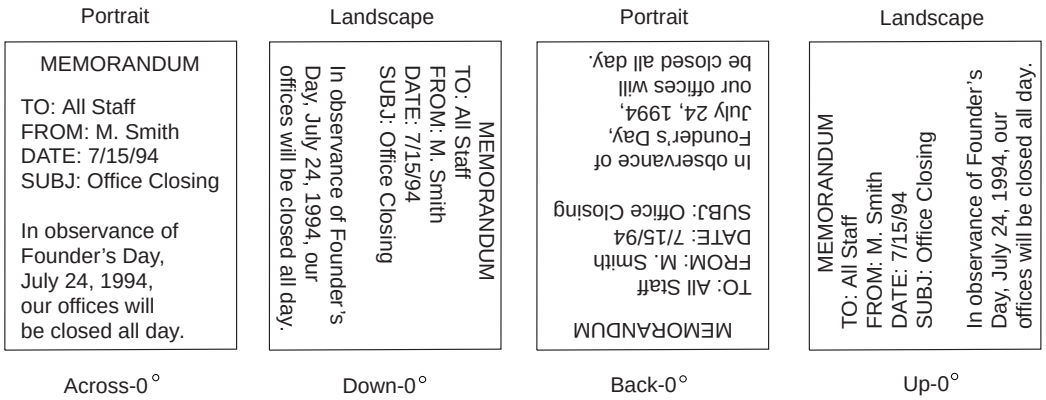


Figure 15. Copies of a Memo Printed in 4 Inline Directions

By having PSF merge data or format line data with a page definition, you can use different inline directions and character rotations for different fields of data. Text-formatting programs can also produce different inline directions and character rotations for text. Figure 16 shows the different combinations of inline directions and character rotations used on the sample page.

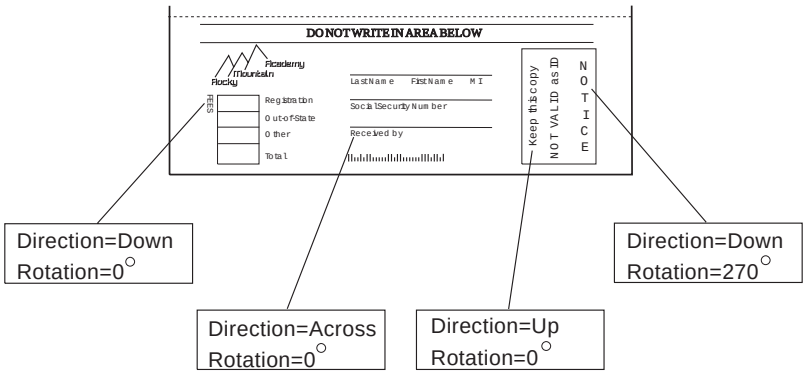


Figure 16. Direction-Rotation Combinations from the Sample Page

Using Fonts

You can use fonts supplied by IBM or other companies, or you can create fonts using an IBM licensed program. You can also convert 240-pel fonts to 300-pel fonts, using the font conversion program provided with PSF.

You can use outline fonts on printers that support outline fonts. The outline font names are unique and begin with either *XZ* or *CZ*. These fonts are available on the AFP Font Collection.

Many outline fonts supplied in the AFP Font Collection have equivalent raster fonts in selected sizes. If you specify outline fonts in your print jobs and are printing on a printer that supports only raster fonts, PSF will try to use equivalent raster fonts. You can create raster fonts from outline fonts in the sizes you need by using Type Transformer and storing them in font libraries for use with PSF. PSF's user font mapping tables should be updated with information about these new fonts. For more information about using PSF's font mapping tables, refer to *Print Services Facility/MVS: System Programming Guide*, or see your system programmer.

For MO:DCA data, you identify the fonts in the data stream. For information about using the CHARS parameter in JCL to specify fonts, see page 42.

PSF host fonts might be fixed-metric fonts with a resolution of 240 pels. To use PSF fonts on a printer that prints with relative metric fonts having a resolution of 300 pels, you must *scale*, or convert, the 240-pel fonts. The APSRCF30 utility uses 240-pel, fixed-metric fonts as input and produces 300-pel, relative metric fonts as output. The APSRCF30 utility, running as a batch program, converts existing fonts and places the output in a partitioned library. For more information about the APSRCF30 font conversion utility, refer to *Print Services Facility/MVS: System Programming Guide*.

Page Segments

A *page segment* is a collection of predefined data that can be merged with the variable data on a page being printed. You can include a page segment in a print data set or in an overlay resource.

A page segment usually contains image data such as signatures, logos, or graphics converted into image format. Page segments containing images can be created by IBM image and graphics products such as GDDM-Image View utility or GDDM-Presentation Graphics Feature.

Page segments can also contain character data formatted as MO:DCA data; however, limitations exist when you are using page segments that contain text, and no IBM products create page segments that contain text. For more information about page segments, refer to *Mixed Object Document Content Architecture Reference*.

The page segments on the sample page are shown in Figure 17.

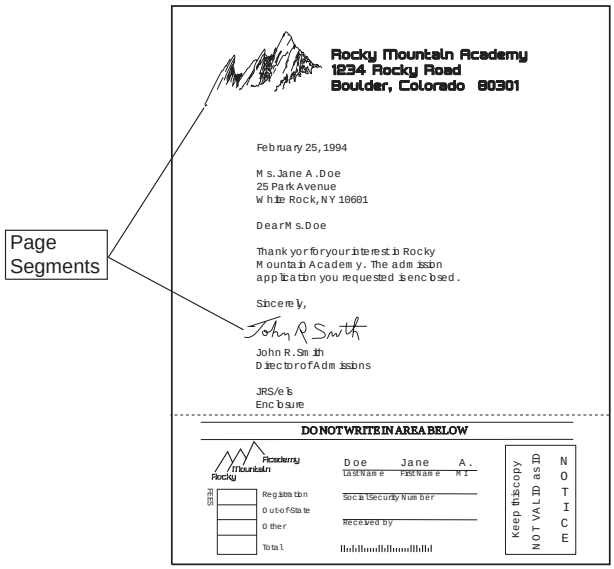


Figure 17. Page Segments on the Sample Page

PSF can position a page segment at various locations on a page. The position of a page segment is defined relative either to the logical page origin or to the origin of the overlay. (The logical page origin is defined in the form definition.) For added flexibility in positioning a page segment, you can specify that PSF print the page segment at the current print position on the page.

Note: When using page segments on a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Using Page Segments

To include a page segment in a print data set or an overlay, you can use commands in an AFP resource utility such as Page Printer Formatting Aid, or in an application such as DCF, to specify the name of the page segment and where to place it on the page. For example, if you are using Document Composition Facility (DCF) to create a print data set, you can include a page segment on a page with a SCRIPT control, described in *Document Composition Facility: SCRIPT/VS Language Reference*.

When printing a page segment on more than 1 page of a print data set, you can improve performance by requesting that PSF retain the page segment in the printer while the file is printing. Page segments that are retained in printer storage are called *hard* page segments. To request that PSF retain the page segment in printer storage:

- For line data, specify the page segment in the page definition.
- For page data, identify the page segment in a structured field in the print data set. If, for example, you are using Document Composition Facility (DCF), you can specify a SCRIPT parameter when you include the page segment.

If a page segment is included in an overlay, PSF always retains the page segment in printer storage as long as the overlay is retained.

You can also include page segments in a print data set by coding structured fields in the print data set. *Mixed Object Document Content Architecture Reference* describes how to code the structured fields used to include page segments on a page.[6] For an example of how to use these structured fields, see “Printing Page Segments” on page 106.

You can specify a maximum of 127 page segments for a single sheet. With IBM printer microcode for extended page segments, the maximum number of page segments is extended to a total of 32 511 page segments per sheet.

[6] The Include Page Segment structured field is used to include a page segment on a particular page of a print data set or in an overlay. The Map Page Segment structured field is used in a page definition or a page print data set to name the page segment for resource management.

Overlays

An *overlay* is a collection of predefined data that can be merged with other data on a page as the page is printed. An overlay can also be printed on a page by itself.

Because an overlay can be printed on a page at the same time as the print data set is printed, overlays can be used as electronic forms to replace preprinted forms. The print data set can contain data that fills out the overlay.

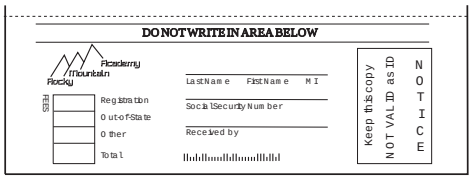


Figure 18. Overlay on the Sample Page

An overlay can contain the following elements sometimes found on a preprinted form:

- Vertical, horizontal, and diagonal rules
- Rules with different weights and thicknesses
- Graphics or company logos
- Bar codes
- Text
- Different inline directions and character rotations for text
- Different fonts, including fonts not used in the print data set
- Boxes with and without shading
- Grids, arcs, and polygons

An overlay can also include page segments; PSF will retrieve the page segment resource when printing the overlay. However, an overlay cannot include another overlay.

PSF supports *medium overlays* and *page overlays*. The 2 kinds differ in where PSF places the overlay on a page during printing, and in how you specify that the overlay is to be printed on a page.

You can include any overlay on a page as either a medium overlay or a page overlay. However, PSF positions medium overlays and page overlays differently on a page; therefore, when you create an overlay, the size and position you specify for it will need to be specified differently according to which type you are creating.

You can create both medium and page overlays by using an IBM product such as Overlay Generation Language/370 (OGL/370). For more information about coding overlays, refer to *Overlay Generation Language/370: User's Guide and Reference*.

Note: When using overlays on a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Medium Overlays

PSF always positions a medium overlay at the media origin. Thus, a medium overlay is positioned at the same place on each page on which it is printed. The medium origin is printer-dependent. For information about the medium origin for your printer, refer to *Advanced Function Presentation: Printer Information*.

Figure 19 shows the positioning of a medium overlay.

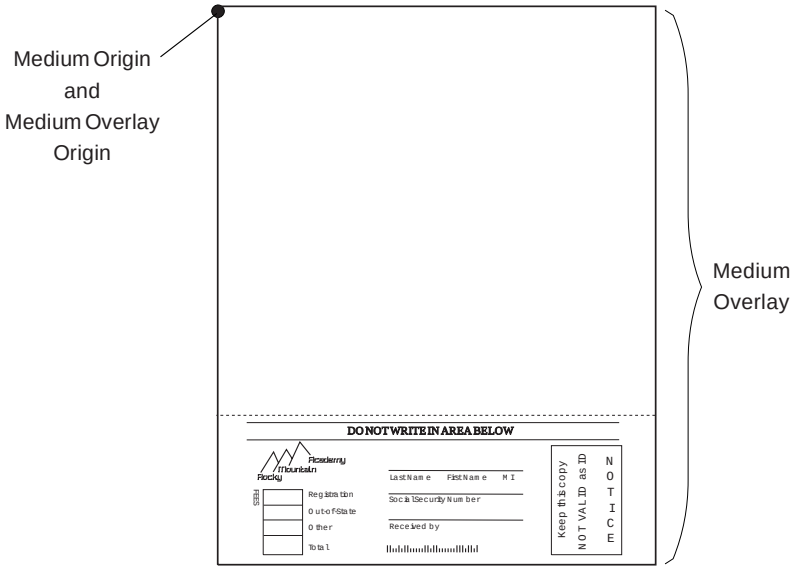


Figure 19. Positioning of a Medium Overlay. The overlay is positioned at the medium origin.

As Figure 19 shows, the size of a medium overlay is generally the same as the page size. In this example, the positioning information in the overlay causes PSF to print the data in the overlay at the bottom of the page.[7]

PSF positions a medium overlay at the same location on every page. To print overlay data in different locations on different pages, you will need to create different versions of a medium overlay.

Page Overlays

PSF can position a page overlay, like a page segment, at various locations on different pages. The position of a page overlay can be defined in the print data set itself, relative to the logical page origin of each page on which the page overlay is to be printed. The logical page origin is defined in the form definition.

Another way to include a page overlay is to specify its name and position in a page definition, using an AFP resource utility such as Page Printer Formatting Aid (PPFA). For more information, refer to *Page Printer Formatting Aid: User's Guide and Reference*.

[7] For example, the OGL/370 OFFSET parameter increases the total size of the compiled overlay resource and offsets the overlay data within the expanded overlay boundaries.

For added flexibility in positioning a page overlay on a page, you can specify in the print data set that the page overlay is to be printed at the current print position on the page. Figure 20 shows the positioning of a page overlay at specified horizontal (***x***-direction) and vertical (***y***-direction) distances from the logical page origin.

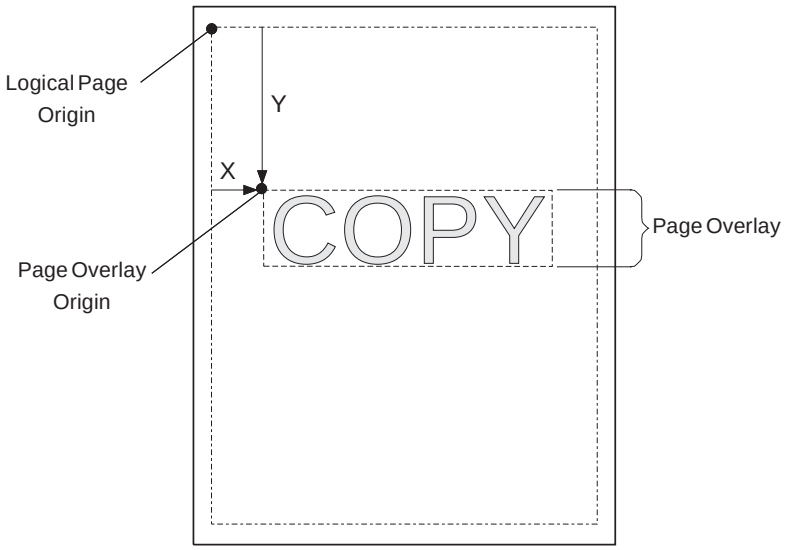


Figure 20. Positioning of a Page Overlay. The overlay is positioned relative to the logical page origin.

The size of a page overlay may be the same as the page size; or, as Figure 20 shows, the size of a page overlay may be limited to the size of the overlay data. If you limit the size of the overlay to the size of the overlay data, you will be able to position the overlay more easily in different locations on different pages.[8]

Because the same page overlay can be positioned at different locations, you need to create only 1 version of a page overlay, even if that overlay is to be printed in several different positions. To change the positioning of a page overlay, you need only to change the placement of the overlay in the print data set, or change the logical page origin in the form definition.

Note: Not all AFP printers support page overlays. If the printer does not support page overlays, PSF issues an error message and prints the page without the overlay.

Medium Overlays and Page Overlays on the Same Page: PSF can print medium overlays and page overlays on the same page. Figure 21 on page 32 shows the sample page with the medium overlay from Figure 19 on page 30 and the page overlay from Figure 20 printed along with the variable data from the print data set.

[8] If you are creating the overlay with OGL/370, use overlay OFFSET value of zero.

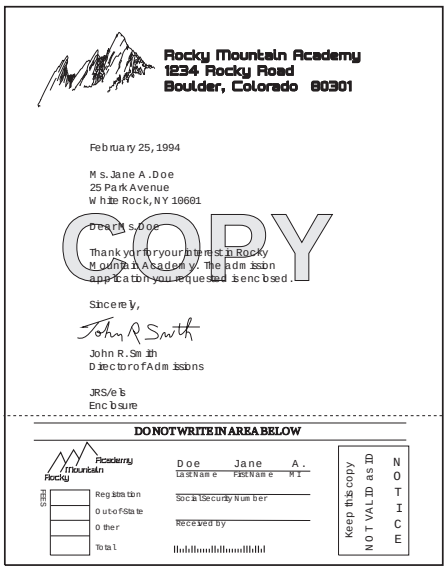


Figure 21. Medium Overlays and Page Overlays on the Same Page

Using Overlays

You can use either medium overlays, page overlays, or both. The choice depends in part on the application. To print the same overlay in different positions on a page, you must use a page overlay. To print the same overlay in the same position on each page of a print data set, you can use either a medium overlay or a page overlay; for this kind of application, however, you will find specifying a medium overlay easier.

The 2 kinds of overlay are specified by different procedures.

Medium Overlays: To include a medium overlay on some or all of the pages of a print data set, specify the overlay name in the form definition. Because a medium overlay is included by simply specifying the overlay in a form definition, you need not modify the print data set in order to print a medium overlay on every page.

If you want to print a medium overlay without merging it with data from the print data set, you can include a blank page in the print data set, or you can specify the *constant-forms* function in the form definition. When the constant-forms function is specified, PSF prints the medium overlay on a page by itself. Thus you can, for example, print an overlay on the back side of each page in a print data set without adding any blank pages to the print data set.

When you use a medium overlay, you can also specify different overlays for different copies of a page. For example, you can print 1 copy of a page with an overlay, the second copy of the page with a different overlay, and the third copy with no overlay.

The number of medium overlays that you can use on 1 page depends on the complexity of the overlays and on the amount of storage required for fonts and page segments in the overlays and for the page. You can specify a maximum of 8 medium overlays in a subgroup.

Page Overlays: To include a page overlay in line data, you specify its name in the page definition. You do this by use of commands either in an AFP resource utility

such as Page Printer Formatting Aid (PPFA). You can also code a structured field to include the overlay on a particular page and to identify the print position of the overlay on that page.

To include a page overlay in a MO:DCA document, specify the Include Page Overlay structured field. For example, if you are using Document Composition Facility (DCF), you can include a page overlay by using a SCRIPT control. *Document Composition Facility: SCRIPT/VS Language Reference* describes the use of SCRIPT controls.

Mixed Object Document Content Architecture Reference describes how to code the structured fields required to include page overlays.[9]

Because a page overlay is included by simply specifying the overlay in the print data set (and also in the page definition for line data), you do not need to create a new form definition in order to print a file with a page overlay.

The number of page overlays that you can use on a single page depends on the complexity of the overlays and on the amount of storage required for fonts and page segments in the overlay and for the page.

Note: You can specify a maximum of 127 page overlays for a single sheet when printing on a 3800 printer, and up to 254 page overlays when printing on any other printer. This maximum number includes medium overlays, page overlays, and overlays used for security labeling. (For information about security labeling, refer to *Print Services Facility: Security Guide*). With IBM printer microcode for extended overlays, the maximum number of overlays is extended to 32 511 overlays per sheet.

Object Container Resource

An *object container resource* is a set of structured fields used to carry object data for a variety of objects. The only object container resource supported by PSF is the microfilm setup resource.

Microfilm Setup Resource (COMSETUP)

To print to a microfilm device, you need to specify a microfilm setup resource for your document. The microfilm setup resource is similar in concept to a form definition. The structure of the object container data field in the microfilm setup resource is defined by the manufacturer of the microfilm device. You can use microfilm setup resources to specify the processing commands for various microfilm device functions, such as titles, format, and extraction masks.

For information on how to use the utility that generates and packages the data in the microfilm setup resource, refer to *Print Services Facility/MVS: System Programming Guide*. The utility, provided with the microfilm device, places the setup resource information in an AFP Object Container, which is associated with the AFP print job by using JCL keywords. The resulting microfilm resource object container can be placed in a system library, in a user library, or inline in the data stream. PSF sends the object container setup resource with your data set to the microfilm device. If you do not specify a microfilm setup resource for your

[9] To include a page overlay on a particular page of a print data set, use the Include Page Overlay structured field. To name the page overlay for resource management, use the Map Page Overlay structured field in a page definition or a print data set.

document, PSF uses the host default microfilm setup resource that your system programmer created, and your output may not appear as you expected.

For more information on the microfilm setup resource, refer to the publications provided with the microfilm device.

Because a microfilm setup resource is 1 type of object container resource, and PSF does not enforce a prefix for the 8-character name of the microfilm setup resource, your site should define a naming convention to avoid future conflicts with other object container resources. IBM recommends a prefix of **H1** for microfilm setup resources.

For more information on sending print jobs to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Using Resources with the Distributed Print Function (DPF) of PSF

The Distributed Print Function (DPF) of PSF stores data and resources sent from a host system (System/390, AIX/6000, or OS/400) on a Local Area Network (LAN) or a Wide Area Network (WAN). DPF drives printers attached to PSF to print data sent from the host.

DPF can store PSF/MVS and PSF/VSE resources in its libraries. Thus you save the time and expense of having to transmit the resources from the host libraries. However, DPF cannot store PSF/VM or OS/400 resources; resources from these 2 systems must be sent with the print data each time they are needed.

For more information about using the DPF function of PSF, refer to the publications listed under “Print Services Facility/2” on page 162. For more information about using APSRMARK to mark resources to use with DPF, refer to *Print Services Facility/MVS: System Programming Guide*.

Using APSRMARK to Mark Resources

Resources are marked as PUBLIC or PRIVATE by using a PSF utility called APSRMARK. A mark of PUBLIC instructs PSF to look for a resident version of this resource first, and, if n1 is found, to download the host resource. A mark of PRIVATE (or no mark at all) instructs PSF to download the host resource.

If you create a new version of a marked resource by using an AFP utility program such as Overlay Generation Language/370, the new version will not be marked and will be downloaded and used instead of the current version. If you then use APSRMARK to mark the new version of the resource as PUBLIC, that resource will be time- and date-stamped by APSRMARK. The next time a job uses that resource, the new host version will have a more recent stamp than the current resident version, and therefore the version will be downloaded and saved again.

Because the time- and date-stamp for PUBLIC resources are always compared, you are assured of getting the most recent host version of a resource if you use standard IBM utilities for creating your resources. If you do not use an IBM utility to directly update the resource object code for a marked resource, however, you can circumvent the process. If you update the resource without deleting or changing the APSRMARK time- and date-stamp, PSF cannot differentiate between your updated resource and the resident version and will use the resident version instead of the updated version.

For additional information about using APSRMARK, refer to *Print Services Facility/MVS: System Programming Guide*. System programmers generally use APSRMARK to mark resources.

Chapter 2. Using JCL for Advanced Function Presentation

This chapter describes the commands and syntax rules you use to print a job with PSF, and gives abbreviated examples. For more complete task examples, see Chapter 4, “Printing Tasks and Examples.”

Each page printer is started with an assigned set of initialization parameters that identify resource libraries and establish the initial printing environment.[10] You can specify JCL parameters in OUTPUT and data definition (DD) statements to do the following:

- Assign OUTPUT statements to a DD statement.
- Select the printer.
- Indicate when checkpoints should be taken, so that printing can be continued from a checkpoint if the processing of a data set is interrupted.
- Specify how error messages are to be handled during printing.
- Specify how data checks are to be handled during printing.
- Select page definitions and form definitions for PSF.
- Specify font information.
- Specify the number of copies to be printed.
- Specify a form name for channel attached printers (except the 3820 printer).
- Specify a user resource library.
- Request that PSF notify you or another user when your print job is finished.
- Select a forms flash for the IBM 3800 printer.
- Indicate whether you want output on continuous-forms paper to be burst.
- Specify parameters to aid in the distribution of your output.
- Override print-labeling defaults (for authorized users only). For more information about print labeling, refer to *Print Services Facility: Security Guide*.
- Specify the microfilm object container resource to be used when you send AFP output to a microfilm device.
- Specify the output bin in which the printed output is to be placed.

This chapter shows you which JCL statements and parameters are used for each of these tasks and provides simple coding examples. For details about parameters and syntax, refer to the JCL reference publication for your operating system.

Determining Printer Defaults

When JES starts a printer, JCL in the PSF startup procedure establishes libraries and sets up initial parameters for the printer. These parameters become the default values for the system. For an explanation of the JCL in the startup procedure and for a general description of how PSF selects resources, refer to *Print Services Facility/MVS: System Programming Guide*.

[10] PSF-supported printers are commonly, but not always, controlled by OUTPUT and DD statements. For more information, refer to the JCL publications.

Page Printer Defaults Form

Before submitting a file for printing, you need to know the defaults for each printer in your installation. At the minimum, you must know the printer selection parameters for the printer you want to use. See your system support group for this information.

For a handy reference, you can copy the form on page 156 and fill it in with the defaults for your installation. Complete a form for each printer you use. You can then select a printer with the defaults you need by referring to the information you have on the forms. You can also determine the parameters you must specify for your file. As an example, a completed form is shown on the next page.

PAGE-PRINTER DEFAULTS FORM

PAGE-PRINTER Type is: **3820**

Printer selection parameters: CLASS = **B**

DEST = **REMOTE**

1. Media or form usually loaded in printer:
   For continuous-forms printer, size of form usually loaded: _____

   For continuous-forms printer with a burster-trimmer-stacker (BTS), output usually burst? YES | NO

   For cut-sheet printer, type of media usually in each source:

   Bin 1 **8 1/2 x 11 (white paper)** Bin 2 **8 1/2 x 11 (blue paper)**

   Bin ___ _____ Bin ___ _____
2. DATACK = BLOCK | **UNBLOCK** | BLKCHAR | BLKPOS
3. CKPTPAGE = **0** or CKPTSEC = _____
4. PIMSG = **NO** | YES Message count = **16**
5. FORMDEF = **F1A10110**

   Copy Group = **F2A10110**

   Page Position = **0.165", 0.165"**

   Paper source for cut-sheet printers: **BIN1** | BIN2 | BIN ___

   Duplex: **NO** | NORMAL | TUMBLE | RNORMAL | RTUMBLE

   For 3800 printer: Flash activated? YES | NO
6. PAGEDEF = **P1A06462**

   Form size = **8 1/2 x 11**

   No. of print lines = **64**

   Printable area = **8.17" x 10.67"**

   Lines per inch = **6**

   Print direction: **ACROSS** | DOWN | BACK | UP
7. Default fonts are (listed in TRC sequence):

   0 = **GT10** 1 = _____ 2 = _____ 3 = _____

   If PAGEDEF is not specified in JCL:

   Fonts **CAN** | CANNOT be changed using CHARS[11]
8. Fonts supported by printer: Resident | **Host** | Both

   Double-byte fonts supported? **YES** | NO

   If yes, default PRMODE: **SOSI1** | SOSI2

   PSF exit APSUX07 has raster fonts automatically mapped to outline fonts for this printer: YES | **NO**
9. Printer attached to DPF or PSF Direct? YES | **NO**
10. Other characteristics of this printer:

    2 megs of raster storage

    _____
11. COMSETUP = **H1SETUPD** (microfilm device only)
12. Where are the messages sent: **Printed with output** | Printed on printer _____ | On output class_____

Creating JCL for Direct Printing Mode

When a job requires direct-printing mode, the PRINTDEV JCL statement is included in the JCL for the job. No PSF startup procedure is used. Some parameters can be specified in the PRINTDEV JCL statement rather than in an OUTPUT or DD statement. The PRINTDEV statement is described in *Print Services Facility/MVS: System Programming Guide*.

Some of the parameters described in this chapter do not apply if the printer is in direct-printing mode. If a parameter does not apply, its description includes a statement to that effect.

Note: Some printers cannot be used in direct-printing mode. For more information about these printers, refer to *Advanced Function Presentation: Printer Information*.

Assigning OUTPUT Statements to a DD Statement

To assign 1 or more OUTPUT statements to the DD statement, use the OUTPUT parameter in your DD statement. This causes the parameters on the OUTPUT statement to apply to that DD statement. The names of the OUTPUT statements are referred to by the OUTPUT parameter in the DD statement.

All OUTPUT statements assigned to a DD statement must precede the DD statement. The system processes a separate output data set for each OUTPUT statement listed in the OUTPUT parameter of a DD statement. The following example shows the OUTPUT statements that are assigned to the DD1 DD statement:

EXAMPLE

```
//OUTPUT1 OUTPUT DEST=name
//OUTPUT2 OUTPUT DEST=name,FORMDEF=fdefname,PAGEDEF=pdefname
//DD1 DD SYSOUT=A,OUTPUT=(*.OUTPUT1,*.OUTPUT2)
```

In this example, the system produces separate output data sets for each of the OUTPUT statements: a data set for OUTPUT1, and a data set for OUTPUT2.

Some parameters can be specified only in a DD statement, and some can be specified only in an OUTPUT statement. Others can be specified in either a DD statement or an OUTPUT statement. If you specify the same parameter in both statements, PSF uses the DD statement parameter rather than the OUTPUT statement parameter. For example, if you use the following JCL, the printer destination selected is PRT0, not PRT1.

EXAMPLE

```
//OUTPUT1 OUTPUT DEST=PRT1
//DD1 DD SYSOUT=A,DEST=PRT0,OUTPUT=(*.OUTPUT1)
```

In this and other examples in this publication, DD statements use specifically assigned OUTPUT statements. However, an OUTPUT statement can be assigned to a DD statement by default by specifying DEFAULT=Y on the OUTPUT statement. For more information about this assignment or about the DD and

[11] Fonts can be changed unless a JES default forms control buffer (FCB) (page definition) defined for this printer specifies fonts.

OUTPUT statements, refer to the JCL reference publication for your operating system.

For a summary of which parameters apply to DD statements and which apply to OUTPUT statements, see Table 1 on page 66.

Specifying AFP Parameters in the JCL

The AFP parameters in JCL are as follows:

```
BURST=YES | NO
CHARS=(fontname1,fontname2[,fontname3][,fontname4])
CKPTPAGE=pages | CKPTSEC=seconds
CLASS=name
COMSETUP=membername
COPIES=(nnn,(groupvalue,groupvalue...))
DATACK=BLOCK | UNBLOCK | BLKCHAR | BLKPOS
DEST=destination name
FCB=pdefname
FLASH=(flashname,count)
FORMDEF=fdefname
FORMS=formname
NOTIFY=(node.userid1,node.userid2[,node.userid3],node.userid4])
OUTBIN=(1-65 535)
PAGEDEF=pdefname
PIMSG=YES | NO | (YES,nnn) | (NO,nnn) |(,nnn)
PRMODE=SOSI1 | SOSI2 | aaaaaaaa
SEGMENT=page-count
SYSOUT=(class,, formname)
TRC=YES | NO
UCS=fontname
USERLIB.=('libname1','libname2',...'libname8')
```

Figure 22. AFP Parameters in JCL

The following user JCL parameters add information to the separator pages printed with an output data set. These additional parameters can help in distributing your printed output.

```
ADDRESS=address
BUILDING=building
DEPT=dept
NAME=name
ROOM=room
TITLE=title
```

Figure 23. Additional JCL Parameters for Distributing Output

BURST=YES | NO

Specifies whether you want continuous-forms output separated into single sheets. Bursting and stacking continuous-forms paper on a 3800 or 3900 printer means separating continuous-forms paper into separate sheets. Use

the BURST parameter to specify whether you want the output paper to go to the post-processing device that bursts, trims, or stacks, or to the continuous-forms stacker (CFS). If your data set is printed on a cut-sheet printer, PSF ignores the BURST parameter.

Specify the BURST parameter either in a DD statement or in an OUTPUT statement, as follows:

Y or YES
Burst the printed output into separate sheets in the burster-trimmer-stacker (BTS).

N or NO
Specifies that the output is to be sent to the continuous-forms stacker in continuous fanfold form.

EXAMPLES

- Following is an example of JCL for printing on a 3900 printer with a BTS installed. The BURST parameter is specified in the DD statement to indicate that you want the output burst (separated into single sheets).

```
//OUTPUT1 OUTPUT FORMDEF=offset
//DD1 DD SYSOUT=class,DEST=pr3900,BURST=YES,OUTPUT=(*.OUTPUT)
```

- Following is an example of JCL in which BURST is specified in the OUTPUT statement:

```
//OUTPUT1 OUTPUT BURST=YES,FORMDEF=offset
//DD1 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
```

Note: If you are using the “Page-Printer Defaults” form, it indicates whether a specific continuous-forms printer has a BTS. See page 38.

CHARS=(*fontname1***,***fontname2***[,***fontname3***][,***fontname4***])**
Specifies the 4-character member name of the coded font that you want to use to print a data set.

fontname
Specifies the name of a coded font (in a font library) containing 6 or fewer characters, including the prefix.

When you uses CHARS to specify the member name, do not include the 2-character prefix of the coded-font name (X0 through XG).

Coded fonts that can be used with the CHARS parameter are supplied with the IBM AFP Font Collection. For details on the available fonts and the naming conventions, refer to *IBM AFP Fonts: Font Summary*.

To use fonts with PSF, you make 2 choices: you *specify* fonts and *select* fonts. You specify fonts to print the entire print data set. You select fonts to print individual lines or fields of data.

For MO:DCA data, you specify and select the fonts within the data stream. If PSF receives MO:DCA data in which no fonts are specified, PSF uses the fonts specified in the JCL.

For data that PSF has to compose into pages, the fonts are specified either in a page definition or in JCL, but not in both. For a single data set, you cannot mix fonts specified in a page definition with fonts specified in JCL. Select fonts with table reference characters (TRCs), with AFP control records, or in a page definition.

Remember the following when you specify fonts with CHARS:

- If you want the entire data set printed in a single direction, you can specify fonts only in JCL. PSF uses the fonts that have 0° character rotation for the specified direction. When a data set requires fonts with more than 1 print direction or character rotation, you must specify the fonts in the page definition. To verify whether fonts can be specified in JCL for the default page definition, see the "Page-Printer Defaults" form.

  Not all printers can print in all 4 directions. For information about the print directions your printer supports, refer to *Advanced Function Presentation: Printer Information*.

- The fonts you specify must reside in a library assigned to the printer you are using or in a user library specified with the JCL USERLIB parameter, or else they must be inline with the print data set.

- You can specify from 1 to 4 fonts in the JCL statement. If you specify more than 1 font with the JCL CHARS parameter, you must use the TRC parameter to tell PSF which font to use for each line of data.

- Raster versions of the outline fonts will be used unless the system programmer specifies XTP7MTOF in PSF exit 7 (APSUX07) and the font name conforms to the CHARS naming convention. Generally, you should specify raster fonts in the CHARS parameter.

EXAMPLES

- Following is an example of JCL in which the CHARS parameter is specified in the DD statement:

  ```
  //DD1 DD SYSOUT=class,CHARS=(fontname1,fontname2[,fontname3,)
  ```

- Following is an example of JCL in which the CHARS parameter is specified in the OUTPUT statement:

  ```
  //OUT1 OUTPUT CHARS=(fontname1,fontname2[,fontname3,],])
  //DD1 DD SYSOUT=class,OUTPUT=(*.OUT1)
  ```

Remember the following when you specify fonts:

- If you specify a page definition using the PAGEDEF or FCB parameter, and that page definition or FCB specifies fonts for your data set, PSF ignores the CHARS parameter. For more information, see "Specifying Fonts" on page 84.

- If you specify both a CHARS parameter and a USC[12] parameter, PSF ignores the USC parameter. If you specify neither a CHARS nor a USC parameter, and if no fonts are contained in the page definition you specified, PSF uses the default fonts defined for your printer. See your system programmer for more information. If you are using the "Page-Printer Defaults" form, the default fonts are identified.

- If PSF uses a default page definition that names a font, and that page definition is specified in the PSF startup procedure, you can use a CHARS parameter to override font specifications in the default page definition. However, if the default page definition was specified as the JES default

[12] The USC parameter is never used in deferred-printing mode under JES3.

FCB for your printer, you cannot use the CHARS parameter to override fonts specified in that page definition.

- You can use a CHARS parameter in jobs printed on 3800 line printers, without changing your JCL.

CKPTPAGE=*pages*

Specifies how many pages you want between data-set checkpoints. The range for *pages* is from 1 to 32 767, inclusive.

Note: When sending output to a microfilm device, see Appendix D, "Microfilm Device Considerations."

CKPTSEC=*seconds*

Specifies how many seconds you want between data-set checkpoints. The range for *seconds* is from 1 to 32 767, inclusive.

Note: When sending output to a microfilm device, see Appendix D, "Microfilm Device Considerations."

Use the CKPTPAGE or CKPTSEC parameter to specify when you want the checkpoint data recorded.

EXAMPLES

These examples specify the CKPTPAGE or CKPTSEC parameters in an OUTPUT statement:

- The following specifies the printer checkpoint to be 20 pages. For printing a large data set, see the considerations described below.

```
//OUT2 OUTPUT CKPTPAGE=20
//DD2 DD SYSOUT=class,OUTPUT=(*.OUT2)
```

The following specifies the printer checkpoint to be 120 seconds:

```
//OUT1 OUTPUT CKPTSEC=120
//DD1 DD SYSOUT=class,OUTPUT=(*.OUT1)
```

Remember the following when you specify checkpoints:

- CKPTPAGE and CKPTSEC do not apply to direct-printing mode.
- Do not specify both CKPTPAGE and CKPTSEC. If you do, the parameter used will depend on the defaults for your installation.
- If you specify neither the CKPTPAGE nor the CKPTSEC parameter, JES uses the system default. If you are using the "Page-Printer Defaults" form shown on page 156, the default checkpoint values for the system are shown. If no system default is given, PSF does not record checkpoints.
- PSF takes internal checkpoints at the specified intervals and transmits a request to JES to record the checkpoint data it gathers. The operator can use these internal checkpoints when issuing commands to the printer.
- A smaller checkpoint interval causes more internal checkpoints to be taken during the processing of a data set. This enables PSF to find the target of the operator command with less processing overhead. However, too small a checkpoint interval can cause excessive use of virtual storage. Be aware of the need to balance between the overhead for processing operator commands and the use of virtual storage.
- If a system failure occurs, PSF does not guarantee that checkpoint information will be used to restart the data set.

- When the PSF attachment is SNA or TCP/IP, the checkpoint interval specifies the maximum number of pages or amount of time after which PSF will request an acknowledgment from the printer. If acknowledgments are requested too often, printing performance can be adversely affected.
- When the PSF attachment is SNA or TCP/IP and after a VTAM session is lost, PSF attempts to recover from the failure, starting from a checkpoint, but only if FAILURE=WCONNECT is coded in the PRINTDEV. After the recovery, the printer may not be able to report what had been printed, and therefore PSF may not be able to account for some pages in the SMF type 6 recording. For other values of FAILURE=, the data set that was printing at the time that the session was lost will be restarted from the beginning.

CLASS=*name*

Specifies the 1-alphanumeric character print class for printing the data set.

To specify an output class, either use a SYSOUT parameter on the DD statement, or use a CLASS parameter on an OUTPUT statement assigned to that DD statement. You must specify a class in 1 statement or the other.

If a single printer is defined for an output class, CLASS is all you have to specify. If, however, an output class contains a group of printers, select a printer from the group by specifying a destination (DEST) parameter. If you do not, the system selects the first available printer in the class.

EXAMPLES

To select a printer when only 1 printer is defined for an output class, type the following:

```
//DD1 DD SYSOUT=A
```

The CLASS subparameter of the DD statement SYSOUT parameter overrides the OUTPUT JCL CLASS parameter. In order to use the OUTPUT JCL CLASS parameter, you must code a null class—SYSOUT=(,)—on the DD statement.

```
//OUTPUT3 OUTPUT CLASS=C,FORMDEF=fdefname
//DD3 DD SYSOUT=(,),OUTPUT=(*.OUTPUT3)
```

Understanding Disabled Mechanisms:

PSF has the following disabled mechanism support for some PSF-supported printers (your installation may have defined print classes or destinations that do not have this support for disabled mechanisms):

- If duplex printing is disabled or is not supported for a specific printer, the printer prints in simplex.
- If the primary paper source is disabled, the printer selects paper from 1 of the alternate paper sources.
- If offset stacking is disabled, no offset stacking occurs.

For information about which mechanisms can be disabled for each type of printer, refer to *Advanced Function Presentation: Printer Information*.

If a mechanism is disabled, your output may not be as expected, and you may want to reassign output to another printer.

COMSETUP=*membername*

Specifies the member name (from 1 to 8 alphanumeric or national characters) of the object container for the microfilm setup resource. The full member name must be specified; PSF will not add a prefix. The first character cannot be numeric. IBM recommends a prefix of **H1** for microfilm setup resources. PSF

uses the COMSETUP parameter only when sending output to a microfilm device.

EXAMPLES

Following is an example of JCL in which the COMSETUP parameter is specified in the OUTPUT statement for data sent to a microfilm device.

```
//OUTPUT1 OUTPUT DEST=name,FORMDEF=fdefname,PAGEDEF=pdefname,
//         COMSETUP=membername
//DD1    DD  SYSOUT=q,OUTPUT=(*.OUTPUT1)
```

Following is an example of JCL in which 2 OUTPUT statements are specified: 1 for data sent to a microfilm device, and 1 for data sent to a printer.

```
//OUTPUT1 OUTPUT CLASS=comclass,FORMDEF=fdefname,PAGEDEF=pdefname,
//         COMSETUP=membername
//OUTPUT2 OUTPUT CLASS=prtclass,FORMDEF=fdefname,PAGEDEF=pdefname
//DD1    DD  SYSOUT=q,OUTPUT=(*.OUTPUT1,*.OUTPUT2)
```

Check with your system programmer about which CLASS or DEST is set up for sending AFP output to the microfilm device.

Using Microfilm Setup Resources from a User Library:

You can instruct PSF to select a microfilm setup resource from your user library rather than from a system library assigned to PSF. To use a microfilm setup resource from a user library, do the following:

- Reference the user library containing the microfilm setup resource in your JCL. For details, see the description of the USERLIB parameter on page 60 .
- Specify the name of the microfilm setup resource in the JCL COMSETUP parameter.

Using Inline Microfilm Setup Resources: To use a microfilm setup resource, do the following:

- Include an inline microfilm setup resource in the print data set.
- If you specify the COMSETUP parameter in your JCL, the name of the inline microfilm setup resource must match the form definition name specified in your JCL, or else you must specify COMSETUP=DUMMY in the JCL.
- If you do not specify the COMSETUP parameter in your JCL, PSF selects the first inline COMSETUP in the print data set.
- If a microfilm setup resource is included inline with the data, the data set must be identified as containing carriage control characters. If the length of the records in the microfilm setup resource is less than or equal to the logical-record length defined for the data set, you can specify fixed-length records for the record format. If the length of the records in the microfilm setup resource is greater than the logical-record length defined for the data set, you must specify variable-length records variable-blocked with ANSI carriage control characters (VBA) or variable-blocked with machine carriage control characters (VBM) for the record format.

You can include more than 1 inline microfilm setup resource in a print data set, and you can change the microfilm setup resource name in the JCL on different printing jobs to test different microfilm setup resources. If, however, the name of a inline microfilm setup resource does *not* match the COMSETUP name specified in the JCL, PSF uses the microfilm setup resource from the resource library that matches the name in the JCL.

COPIES=(*nnn***,(***groupvalue***,***groupvalue***...))**
Specifies the number of copies you want to print.

Note: When specifying copies to a microfilm device, see Appendix D, "Microfilm Device Considerations."

nnn
Specifies the total number of collated copies you want to print. Up to 255 copies can be printed. The data set is sent to the printer *nnn* times, and each copy is printed in page-number sequence. This parameter applies only to the user's data set. Only 1 transmission is processed for separator pages and for the PSF message data set.

group value
Specifies the number of copies of each page of the data set to be printed consecutively before the next page is printed. You can specify up to 8 copy groups (group values), and the data set is sent to the printer 1 time for each group. The total number of copies equals the sum of the group values. No single group value can be more than 255, nor can the sum of those specified be more than 255.

If you specify 1 or more group values, PSF ignores the *nnn* value.

If a FORMDEF parameter is specified for the print data set, the number of group values specified determines the number of times the user's print data set is transmitted. Only 1 transmission is processed for separator pages and for the PSF message data set. In this case, group values following the first group value are ignored. However, the number of copies of each page is determined by information in the form definition. The effects of specifying FORMDEF and COPIES are described in "Specifying FORMDEF and COPIES or FLASH Parameters in JCL" on page 78.

For more information about printing multiple copies of the job header or the job trailer pages, refer to *Print Services Facility/MVS: System Programming Guide*.

EXAMPLE

In this example, the COPIES parameter is specified in the DD statement. Fourteen copies of the data set are to be printed in page number sequence.

```
//DD1 DD SYSOUT=class,COPIES=14
```

The data set is sent to the printer 14 times. If the data set contains 3 pages, the output is like the example in Figure 24 on page 48.

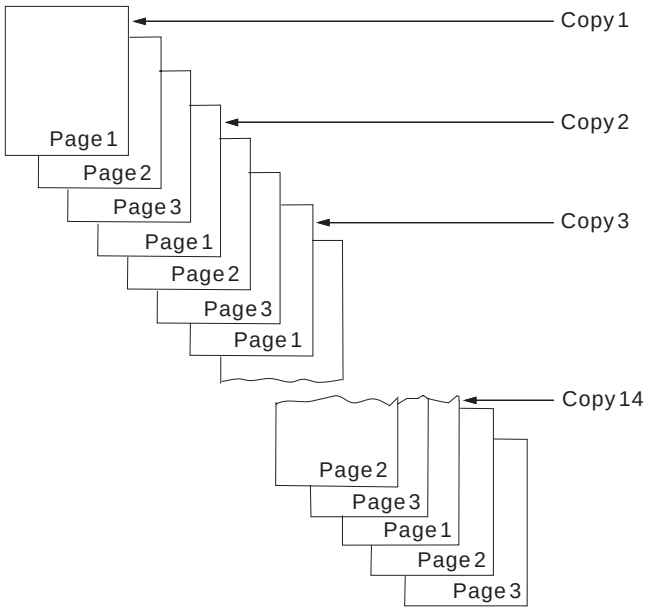


Figure 24. Sample Output for COPIES=14

EXAMPLE

In this example, the COPIES parameter is specified in the OUTPUT statement. Three copy groups are to be printed, producing a total of 6 copies of the data set. The group values are set so that the first group contains 1 copy of each page, the second contains 3 copies, and the third contains 2 copies.

```
//OUT2 OUTPUT COPIES=(,(1,3,2))
//DD2 DD SYSOUT=(,),OUTPUT=(*.OUT2)
```

The data set is sent to the printer 3 times, once for each group value. If the data set contains 3 pages, the output is like the example in Figure 25 on page 49. Notice that in the second and third copy groups, all the copies of a given page are printed before the next page is printed.

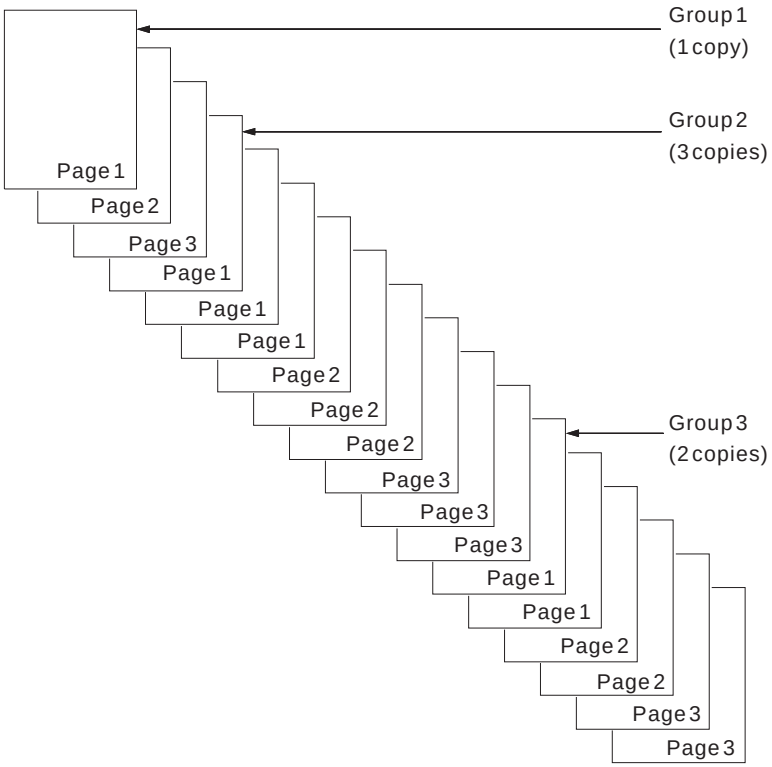


Figure 25. Sample Output for COPIES=(,(1,3,2))

DATACK=BLOCK | UNBLOCK | BLKCHAR | BLKPOS

Specifies whether you want the printer to block print-positioning and invalid-character errors. A print-positioning error is an attempt to print outside the valid printable area. An invalid-character error is an attempt to use a code point that is not assigned to a character.

BLOCK

Specifies that the printer is not to report data-check errors. The printer does not return error messages to PSF, even if data is lost. No data check error messages are created.

UNBLOCK

Specifies that the printer is to report all data-check errors. If a print-positioning error occurs, the exception is highlighted on the printed page to help you find the place at which the printer attempted to print outside the valid printable area.

BLKCHAR

Specifies that the printer is not to report invalid-character errors. The printer reports print-positioning errors as usual, with exception highlighting to show the locations of the errors.

BLKPOS

Specifies that the printer is not to report print-positioning errors. The printer reports invalid-character errors as usual.

EXAMPLE

The following JCL specifies that data-check errors are not to be blocked:

```
//OUTPUT1 OUTPUT DATACK=UNBLOCK
//DD1 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
```

When you specify a DATACK parameter, remember the following:

- Not all printers perform exception highlighting, and the types of highlighting differ among printers. To learn which printers support exception highlighting, refer to *Advanced Function Presentation: Printer Information*.
- If you receive a message stating that a print-positioning error occurred, you can submit the print job again and specify BLOCK or BLKPOS for the DATACK parameter to see how much of the job will print without the error. Output produced in this manner is sometimes acceptable.
- If you notice that data is missing from your output, and you specified DATACK BLOCK or BLKPOS, you can submit the print job again, specifying UNBLOCK or BLKCHAR to receive PSF messages about any print-positioning errors that PSF finds. This method does not apply to invalid-character errors.
- If you do not specify the DATACK parameter, PSF uses the default from the PSF startup procedure. If you are using the Page-Printer Defaults form, the default is shown on the form. If no default is specified, PSF uses the default value, DATACK=BLOCK.

Note: When specifying the DATACK parameter to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

DEST=*name*

Specifies a destination name for the print data set. The destination can be a 1- to 8-character name assigned to a printer on your system, or a node and printer name for a printer attached to a different system.

Not every printer has an assigned destination name. To determine what destination name, if any, you should use for your printer, see page 156 or consult your system programmer.

EXAMPLES

- Following is an example of JCL in which DEST is specified in the DD statement:

  ```
  //DD1 DD SYSOUT=class,DEST=destination
  ```

- This example selects the printer in an OUTPUT statement for the print data set in DD2:

  ```
  //OUT2 OUTPUT DEST=destination
  //DD2 DD SYSOUT=A,OUTPUT=(*.OUT2)
  ```

Remember the following about the DEST parameter:

- If you specify a DEST parameter both in the DD statement and in the OUTPUT statement, PSF ignores the parameter in the OUTPUT statement.
- When your job uses direct-printing mode, the DEST parameter does not apply.

FCB=*pdefname*

Specifies the 1- to 4-character name of the page definition to be used in formatting a print data set. PSF adds the system prefix P1 to the FCB name you specify, and uses the page definition named P1*name*, where *name* is the member name specified in the JCL. Therefore, do not add the P1 prefix to the name of the page definition.

EXAMPLES

- In this example, an FCB is specified in the DD statement:

  ```
  //DD1 DD SYSOUT=class,FCB=pdefname
  ```

- In this example, an FCB is specified in the OUTPUT statement:

  ```
  //OUTPUT1 OUTPUT FCB=pdefname
  //DD1 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
  ```

If you use the FCB parameter, remember the following:

- If both an FCB parameter and a PAGEDEF parameter are coded in your JCL, PSF ignores the FCB parameter. For information about the PAGEDEF parameter, see page 55 .
- Because the maximum length of a PAGEDEF parameter is 6 characters, and the maximum length of an FCB parameter is 4 characters, you cannot use the FCB parameter to specify all page definitions. IBM recommends that you use the PAGEDEF parameter, except when you are converting jobs from line printers, and you do not want any changes in the JCL.

FLASH=(*flashname,count***)**

Specifies the name of the forms flash to be printed, and the number of copies on which the flash is to be printed. A forms flash is a 3800 printer hardware frame that prints a photographic negative on selected forms.

flashname

Specifies the 1- to 4-character name of the forms flash.

count

Specifies the number of copies (from 1 to 255) on which the forms flash is to be printed. The brackets around *count* signify that it is an optional value.

EXAMPLES

- In this example, the number of copies is smaller than the number of flash copies; therefore, the forms flash will print on all 10 copies. In this example, the FLASH parameter is specified in the DD statement.

  ```
  //DD1 DD SYSOUT=class,FLASH=(OVL1,15),COPIES=10
  ```

- If the number in *count* is smaller than the total number of copies requested, the forms flash is printed only on the number of copies specified by the FLASH parameter, beginning with the first copy. Therefore, if you are printing 15 copies and specify a count of 10 on FLASH, the forms flash is printed on the first 10 copies.

  In this example, the FLASH parameter is specified in the OUTPUT statement:

  ```
  //OUTPUT1 OUTPUT FLASH=(OVL1,10)
  //DD1 DD SYSOUT=class,COPIES=15,OUTPUT=(*.OUTPUT1)
  ```

When you specify a forms flash, remember the following:

- If you specify a forms flash within the active copy-group definition, but do not specify a name with the FLASH parameter, PSF uses the forms-flash frame that is currently loaded.
- If you specify both the FORMDEF and FLASH parameters in your JCL, but the form definition does *not* specify FLASH, PSF ignores the FLASH parameter, because the FORMDEF parameter overrides it. The forms flash will not be printed on any of the copies of your data set.
- If you specify the count subparameter in the JCL FLASH parameter, and PSF uses the default form definition because you did not specify the FORMDEF parameter, the modifications specified in the default form definition subgroup (number of copies, flash, overlay identifiers, and suppression identifiers) are not used. A form definition subgroup is described in "Subgroups" on page 77.
- The FLASH parameter is valid only for the IBM 3800 printer; PSF ignores FLASH for other printers. However, you can use overlays on any printer. Overlays are described in "Overlays" on page 29.
- If you are using the "Page-Printer Defaults" form, it indicates whether the default form definition activates forms flash.

Note: A FLASH parameter specified in your JCL statement overrides any FLASH specified in the FORMDEF parameter in the PSF PRINTDEV statement. The PRINTDEV statement, which is set up by your system programmer, contains defaults for PSF to use for any parameters not specified in the JCL.)

FORMDEF=*fdefname*

Specifies the member name (from 1 to 6 alphanumeric or national characters) of the form definition you want to use. Omit the system prefix, F1, from the name; PSF adds F1 to the member name specified in the JCL.

A *form definition* specifies how a page of data is placed on a form, the number of copies of a page, any modifications to that group of copies, the paper source, and whether duplex printing is needed. For more information about the contents of a form definition, see "Form Definitions" on page 67 and Appendix A, "Form Definitions Supplied with PSF."

EXAMPLE: To specify *F1USER10* as the form definition, use the following JCL:

```
//OUT2 OUTPUT FORMDEF=USER10
//DD2 DD SYSOUT=class,OUTPUT=(*.OUT2)
```

When you specify a form definition, remember the following:

- If the FORMDEF parameter is not specified, PSF uses the default form definition specified in the startup procedure for PSF. If you are using the "Page-Printer Defaults" form, this default is contained in the form. For a copy of this form, see page 156.
- When the job uses direct-printing mode, you can specify the FORMDEF parameters in an OUTPUT statement or in a PRINTDEV JCL statement. The PRINTDEV statement is described in *Print Services Facility/MVS: System Programming Guide*.

The form definition you use may be stored in any of 3 places:

- In a system library assigned to PSF for your printer
- In a user library referenced in your JCL
- Inline in the print data set

Using Form Definitions from a User Library:

You can instruct PSF to select a form definition from your user library rather than from a system library assigned to PSF. To use a form definition from a user library, do the following:

- Reference the user library containing the form definition in your JCL. For details, see the USERLIB parameter on page 60 .
- Specify the name of the form definition in the JCL FORMDEF parameter.

Using Inline Form Definitions:

To use an inline form definition, do the following:

- Include the inline form definition in the print data set.
- If you specify the FORMDEF parameter in your JCL, ensure that the name of the inline form definition matches the form definition name specified in your JCL, or else specify FORMDEF=DUMMY in the JCL. If you do not specify the FORMDEF parameter in your JCL, PSF selects the first inline form definition in the print data set.
- If a form definition resource is included inline with the data, ensure that the data set is identified as containing carriage control characters. If the length of the records in the form definition is less than or equal to the logical-record length defined for the data set, you can specify fixed-length records for the record format. If the length of the records in the form definition is greater than the logical-record length defined for the data set, you must specify variable-length records as variable-blocked with ANSI carriage control characters (VBA) or as variable-blocked with machine carriage control characters (VBM) for the record format.

You can include more than 1 inline form definition in a print data set, and you can change the form definition name in the JCL for different printing jobs to test different form definitions. If the name of an inline form definition does not match the FORMDEF name specified in the JCL, PSF uses the form definition from the resource library that matches the name in the JCL. For more information about using inline form definitions, refer to *Advanced Function Presentation: Programming Guide and Line Data Reference*.

Note: If you specify FORMDEF=DUMMY in your JCL, and you do not include an inline form definition, PSF uses the default form definition for your printer.

FORMS=*formname*

Specifies the 1- to 8-character name of the form. The printer operator receives a message saying to load this form only for channel attached printers.

This parameter can also be used to notify the operator that specific Selectable Medium Modifications should be enabled on a post processor. Examples of modifications are applying colored plates, cutting, perforating select pages of output, or using MICR fonts.

Consult your system-support group for the paper sizes or Selectable Medium Modifications available on your printer and for the form identification that you use in JCL to specify the paper or Selectable Medium Modifications you want.

EXAMPLE

In this example, the form is specified in the OUTPUT statement:

```
//OUTPUT1 OUTPUT FORMS=formname
//DD1 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
```

If you are using the “Page-Printer Defaults” form, the form usually loaded in the printer is shown.

Note: For information about special forms and forms for cut-sheet printers, see “Specifying BIN (Paper Source)” on page 93.

NOTIFY=(*node.userid1***,** *node.userid2***,***node.userid3***],***node.userid4***])**

Specifies that you want PSF to notify up to 4 user IDs when the printer finishes printing your job. PSF sends the notice when an output group is stacked. An output group may contain multiple data sets.

The NOTIFY message advises you that the job has been completed, successfully or unsuccessfully, and tells you which output is finished, incomplete, or unprintable.

If the printer is attached through DPF, the message that PSF issues does not indicate that printing is complete; it means that PSF has finished the processing.

EXAMPLE

This example shows how to specify the NOTIFY parameter in an OUTPUT statement:

```
//OUTPUT1 OUTPUT NOTIFY=node.userid
//DD1 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
```

Even if you have specified the NOTIFY parameter in your print-job JCL, in the following circumstances, you will not receive a message saying that your print job has been cancelled:

- If a printer operator presses the printer CANCEL key while the first data set of a job containing multiple data sets is printing, the first data set will be cancelled, rather than the entire group or job. You will not receive a message saying that your print job has been cancelled, but you will receive a message saying that your job is complete, even though the first data set is *not* complete. If the operator presses the CANCEL key while the last data set of a group is printing, you will receive a message saying that your print job has been cancelled.
- If a printer operator types a JES CANCEL command that specifies your job number, and the job contains multiple data sets, your job will be cancelled, but you will not receive a message saying that your output has been cancelled.

OUTBIN=*1-65 535*

Specifies the 1- to 5-decimal-digit identifier of the output bin into which PSF will place a print job. If the output bin is not specified, or if the printer does not support the selection of an output bin, the job is stacked in the default output bin for the printer.

EXAMPLE

In this example, OUTBIN is specified in the OUTPUT statement:

```
//OUTPUT OUTPUT OUTBIN=4
//DD2 DD SYSOUT=class,OUTPUT=(*.OUTPUT1)
```

PAGEDEF=*pdefname*

Specifies the member name (from 1 to 6 alphanumeric or national characters) of the page definition you want to use. When you specify the name in the JCL, omit the system prefix, P1; PSF adds it automatically.

A page definition defines the page format that PSF uses to compose line data into pages. For more information about the contents of a page definition, see "Page Definition" on page 80 and Appendix B, "Page Definitions Supplied with PSF."

EXAMPLE

In the following example, *P1USER10* is specified as the page definition in the OUTPUT statement:

```
//OUT2 OUTPUT PAGEDEF=USER10
//DD2 DD SYSOUT=class,OUTPUT=(*.OUT2)
```

When you select a page definition, remember the following:

- If the PAGEDEF parameter is not specified, but an FCB parameter is coded in the JCL, PSF uses the page definition named in the FCB parameter. See page 51 for information about coding the FCB parameter.
- If neither a PAGEDEF parameter nor an FCB parameter is coded in your JCL, PSF uses the system default. If you are using the "Page-Printer Defaults" form, this default is shown on the form. For a blank copy of this form, see page 156.
- PSF does not support the LINECT parameter on the /*JOBPARM, /*OUTPUT, and OUTPUT JCL statements. The maximum number of lines to be printed on a page can be defined in a page definition.
- When your job uses direct-printing mode, you can specify the PAGEDEF parameter in an OUTPUT statement or in a PRINTDEV JCL statement. The PRINTDEV statement is described in *Print Services Facility/MVS: System Programming Guide*.

The page definition you use may be stored in any of 3 places:

- In a system library assigned to PSF for your printer
- In a user library referenced in your JCL
- Inline in the print data set

Using Page Definitions from a User Library:

You can instruct PSF to select a page definition from your user library rather than from a system library assigned to PSF. To use a page definition from a user library, do the following:

- Include in your JCL a reference to the user library that contains the page definition. For details, see the USERLIB parameter on page 60 .
- Specify the name of the page definition in the JCL PAGEDEF parameter or the FCB parameter of your JCL.

Using Inline Page Definitions:

To use an inline page definition, do the following:

- Include the inline page definition in the print data set.
- If you specify the PAGEDEF parameter or the FCB parameter in your JCL, ensure that the name of the inline page definition matches the name of the page definition name specified in your JCL, or else specify PAGEDEF=DUMMY in the JCL.
- If you do not specify the PAGEDEF or FCB parameter in your JCL, PSF selects the first inline page definition in the print data set, unless a JES default page definition exists.
- If a page definition resource is included inline with the data, ensure to identify the data set as containing carriage control characters. If the length of the records in the page definition is less than or equal to the logical-record length defined for the data set, you can specify fixed-length records for the record format. If the length of the records in the page definition is greater than the logical-record length defined for the data set, you must specify variable-length records variable-blocked with ANSI carriage control characters (VBA) or variable-blocked with machine carriage control characters (VBM) for the record format.

You can include more than 1 inline page definition in a print data set, and you can change the page definition name in the JCL on different printing jobs to test different page definitions. If, however, the name of an inline page definition does not match the PAGEDEF name specified in the JCL, PSF uses the page definition from the resource library that matches the name in the JCL. For more information about using inline page definitions, refer to *Advanced Function Presentation: Programming Guide and Line Data Reference*.

Note: If you specify PAGEDEF=DUMMY in your JCL, and you do not include an inline page definition, PSF uses the default page definition for your printer.

Using Page Definitions Converted from FCBs: PSF supports line-data applications designed for line printers, so that those jobs can be processed on a page printer with no need to change the application program or its JCL. If you do not specify a PAGEDEF parameter, but you do specify a forms control buffer (FCB) parameter, PSF uses the FCB name as the page definition name.

If you have been using line printers, you have probably used some of the FCB modules provided by IBM. Those FCBs have been converted to page definitions and are provided in the system page definition library; they are listed in Appendix B, “Page Definitions Supplied with PSF.” If you used other FCB modules for your line printer, check with your system support group to see whether the FCBs have been converted to page definitions before you attempt to use them.

PIMSG=YES | NO | (YES,*nnn*) | (NO,*nnn*) |(,*nnn*)

Specifies whether you want PSF messages printed.

- One error produces a primary message and possibly other, associated environmental messages.
- A group of messages is generated for a terminating error. For example, the messages for an I/O error contain the following:
  - A message identifier and the I/O error description

– Additional message identifiers with their descriptions of the environment in which the error occurred

- Message groups are separated with a blank line in the printed output.

Y or YES
Specifies that all message groups generated in the processing of a data set are to be printed at the end of the data set, preceding any trailer pages. If a data set is left incomplete because of an error, message groups generated up to this error are printed, including the message group describing the terminating error. YES is the default.

N or NO
Specifies that no message groups are to be printed unless an error occurs that forces printing to stop prematurely. If that happens, only the message group describing the terminating error is printed.

nnn
Specifies that PSF is to stop processing the data set after *nnn* message groups are generated. The default value is 16. The final count of printed messages may be larger than *nnn* if errors are reported for pages sent to the printer before the message count is reached. A value of 0 for *nnn* allows printing of the data set to continue regardless of the number of message groups generated, unless a terminating error occurs.

Note: Your system programmer may have redirected the messages associated with your print job. Consult your programmer to determine where the messages for your job will be sent.

EXAMPLE

This example shows specifying the PIMSG parameter in an OUTPUT statement.

```
//OUT2 OUTPUT PIMSG=NO
//DD1 DD SYSOUT=class OUTPUT=(*.OUT2)
```

If PIMSG is not specified, the default is either of the following:

- If a default is defined in the PSF startup procedure, PSF uses it. For the "Page-Printer Defaults" form used to record the system defaults, see Appendix C, "Page-Printer Defaults Form."
- If a default is not defined in the PSF startup procedure, PSF uses PIMSG=(YES,16).

PRMODE=SOSI1 | SOSI2 | | *aaaaaaaa*
Specifies the type of data in the print data set and whether PSF must perform optional processing of the data.

SOSI1
Specifies that each shift-out, shift-in code is to be converted to a blank and a Set Coded Font Local text control.

The data conversion that PSF makes for SOSI1 is described in "Using Shift-Out, Shift-In (SOSI) Codes" on page 126.

SOSI2
Specifies that each shift-out, shift-in code is to be converted to a Set Coded Font Local text control.

The data conversion made by PSF for SOSI2 is described in “Using Shift-Out, Shift-In (SOSI) Codes” on page 126.

aaaaaaaa
Specifies any alphanumeric string defined in the JES initialization parameters as a selection criterion for a printer to process the data set. **LINE** and **PAGE** are examples of PRMODE values that JES uses for job routing information.

JES uses values in the PRMODE parameter for job routing. PSF ignores all values except SOSI1 and SOSI2, which it uses in printing data sets containing both single-byte and double-byte fonts.

EXAMPLE

The following example specifies that the SOSI1 process mode is to be set up for a data set. PRMODE is specified in the OUTPUT statement.

```
//OUT1 OUTPUT CHARS=(fontname1,fontname2),PRMODE=SOSI1
//DD1 DD SYSOUT=class,OUTPUT=(*.OUT1)
```

Remember the following when you are using the SOSI process:

- The SOSI process must be started when the printer is started. If you are using the Page-Printer Defaults form, it shows whether the process is active or not.
- For the process to work correctly, the first font specified in the CHARS parameter (or in a page definition font list) must be the single-byte font, and the second font must be the double-byte font.

SEGMENT=*page-count*
Specifies that part of the output for a job is to be spooled to print while the job is still running, or specifies that different segments of a job are to be printed simultaneously on different printers. This parameter is optional and can be used only with line-mode data, not with page-mode data.

page-count
Specifies the number of pages to be printed in this segment of the output data set. When the page-count is reached, JES prints that number of pages and begins counting the next segment of 100 pages; it continues counting until the data set is printed.

Notes:

1. SEGMENT works only on JES2 Version 4 Release 1.0 and above. You might use SEGMENT when you need to print a large job but do not want to monopolize a single printer for a long time.
2. Using the SEGMENT parameter with conditional processing will produce unexpected results, because conditional processing is not supported across data-set boundaries. The SEGMENT parameter causes a data set to be broken into what JES considers separate data sets, which causes conditional processing to stop at the end of the first segment (data set).

EXAMPLE

This example shows how to print a large data set in segments of 100 pages each. When JES writes 100 pages to an output data set, the

segment is scheduled for printing, and JES begins the second segment of 100 pages. The process continues until the entire data set is printed. Specify the SEGMENT parameter in the DD statement.

```
//DD1 DD SYSOUT=A,SEGMENT=100
```

SYSOUT=(*class*,, *formname*)

Specifies the output class and, optionally, the form name for the print data set. The output class and the form name can also be specified in the CLASS and FORMS parameters of the OUTPUT statement. For information on using these parameters, see the CLASS parameter on page 45 and the FORMS parameter on page 53 .

EXAMPLE

The following example specifies an output class of **A**. The form name is specified in the OUTPUT statement:

```
//OUT1 OUTPUT FORM=formname
//DD1 DD SYSOUT=A,OUTPUT=(*.OUT1)
```

In the next example, both class and form are specified with the SYSOUT parameter:

```
//DD2  DD  SYSOUT=(A,,formname)
```

To allow class to be specified in an OUTPUT statement, code the SYSOUT parameter as in the following example:

```
//OUT3 OUTPUT CLASS=class
//DD3  DD  SYSOUT=(,),OUTPUT=(*.OUT3)
```

If the class or the form name is coded both in the OUTPUT statement and in the SYSOUT parameter of the DD statement, PSF uses the value coded in the DD statement.

TRC=YES | NO

Specifies whether the print data set contains table reference characters (TRCs).

In line data, you can use different fonts on different lines of a file by specifying TRCs at the beginning of each line after the carriage control characters, if any are present.

EXAMPLES

In the following examples, an OUTPUT statement and a DD statement are used to specify that TRCs are present:

- This example shows the TRC parameter in an OUTPUT statement:

  ```
  //OUT2 OUTPUT CLASS=name,
  //       CHARS=(fontname1,fontname2),TRC=YES
  //DD2 DD SYSOUT=(,),OUTPUT=(*.OUT2)
  ```

- This example shows the DCB subparameter in a DD statement:

  ```
  //OUT2 OUTPUT CLASS=name
  //DD2 DD SYSOUT=(,),OUTPUT=(*.OUT2),
  //       CHARS=(fontname1,fontname2),DCB=OPTCD=J
  ```

When you use table reference characters, remember the following:

- The order in which the fonts are specified in the CHARS parameter establishes which number is assigned to each associated TRC. For

example, the reference characters for the fonts in the preceding example are 0 for fontname1 and 1 for fontname2.

- If both TRC=NO and DCB=OPTCD=J are specified, PSF ignores the TRC=NO parameter and expects the line data to contain multiple fonts.
- If TRC=YES or DCB=OPTCD=J is specified, but the data set contains no TRCs, the first character of each line (or the second character if carriage control characters are used) is interpreted as the font identifier. Consequently, the font used to print each line of the data set may not be the 1 you expect.
- If you do not specify TRC=YES or DCB=OPTCD=J in the JCL, but your line data contains a TRC as the first character of each line (or the second character if carriage control characters are used), the TRC is not used as a font identifier, but is printed as a text character.

UCS=*fontname*

Serves as another way to select a font.

EXAMPLE

- This example shows the UCS parameter in a DD statement:

```
//DD1 DD SYSOUT=class,UCS=fontname
```

Consider the following:

- When a CHARS parameter is not specified in the output JCL, you can specify the universal character set (UCS) parameter to identify 1 font.
- If you specify a page definition on the OUTPUT statement that specifies fonts for your data set, the UCS parameter is ignored. For more information, see “Specifying Fonts” on page 84.

USERLIB=(*'libname1','libname2',...'libname8'***)**

Specifies the name of 1 to 8 cataloged MVS data sets (user libraries) containing AFP resources for processing this print data set. PSF dynamically allocates these data sets and searches for resources in them in the order specified on the USERLIB statement. If PSF finds no resources, PSF searches the system libraries defined in the startup procedure. The libraries you specify can contain any AFP resources: fonts, page segments, overlays, page definitions, form definitions, or object container resources. If RACF is installed on your system, RACF checks the authority of the user ID requesting access to a user library.

Note: Because AFP resources (except page segments and object containers) have reserved prefixes, naming conflicts should not occur; for example, O1 for overlays, F1 for form definitions, P1 for page definitions, S1 for page segments, X0 for coded fonts, C0 for character sets, and so on. IBM recommends that all microfilm setup resources begin with H1.

EXAMPLE

In this example, the USERLIB parameter in the OUTPUT statement tells PSF to search the library specified on the USERLIB statement for the specified data sets. If the data sets are not found, PSF searches the system libraries defined in the startup procedure.

```
//OUT1 OUTPUT USERLIB=('USER.IMAGES','USER.AFP.RESOURCES')
//DD1 DD SYSOUT=class,OUTPUT=(*.OUT1)
```

When you specify the USERLIB parameter, remember the following:

- PSF does not use resources from a user's libraries for security page labeling or for processing separator pages or message data sets.
- For user data sets, PSF ensures that a USERLIB resource is used if USERLIB is specified, even if a resource (not from a USERLIB) with the same name is already loaded in the printer or in virtual storage.

  Note: To use a page definition or a form definition from a user library, you must specify the name of the page definition or form definition in the user JCL.
- An inline resource overrides a resource of the same name contained in a USERLIB parameter.
- If you are printing on a printer driven by the Distributed Print Function of PSF/2 or are using a printer that supports resident fonts, you can use a DPF-resident resource or a printer-resident font instead of your USERLIB resource. For a description of situations in which this can occur, see "Printer-Resident Fonts" on page 21.
- As soon as PSF finishes processing a data set, PSF deletes all the USERLIB resources from the printer or from memory. PSF deallocates USERLIB data sets at data set end to make them unavailable to another user.

Remember the following constraints to dynamic allocation:

- Any libraries specified with the USERLIB parameter must be accessible on all systems on which output may be printed.
- The USERLIB parameter is not supported for direct-printing mode.

If you are using fonts from user libraries, and the fonts have a different pel resolution from that of the printer you are using, you must scale the fonts to fit your printer's pel resolution. For information about how to convert fonts, refer to *Print Services Facility/MVS: System Programming Guide*. Converting fonts is a task usually done by a system programmer.

Using Inline Resources

Resources can be placed in a print data set by an application such as IBM DisplayWrite/370. A resource that is in a print data set is called an *inline resource*. For information about how to code inline resources, refer to *Advanced Function Presentation: Programming Guide and Line Data Reference*.

PSF uses an inline resource to print a data set if the name of the inline resource matches the name of a resource used by the print data set. For example, if the print data set references a page segment named S1LOGO, and an inline page segment is named S1LOGO, PSF uses the inline page segment. For an inline form definition, a page definition, or a microfilm setup resource, you can specify the name in the JCL FORMDEF, PAGEDEF, or COMSETUP parameter, or specify the name DUMMY, or specify no parameters. If you have specified the inline resource name in the FORMDEF, PAGEDEF, or COMSETUP parameter, and your data set contains that inline resource, PSF uses the inline resource. If the name in the FORMDEF, PAGEDEF, or COMSETUP parameter does not match the name of an inline resource, PSF uses the resource from the resource library that matches the name in the JCL. If you have not specified the FORMDEF, PAGEDEF, or

COMSETUP parameter, and your data set contains inline resources, PSF uses the first inline resource (FORMDEF, PAGEDEF, or COMSETUP) encountered in the data set. For information on how PSF selects resources, refer to *Print Services Facility/MVS: System Programming Guide*.

If you use an outline font as an inline resource in your data stream, consider including the equivalent raster version of the font, to ensure that your job can be printed, in case the target printer supports only one type of font technology. For additional information, see “Printing with Inline Resources” on page 113. For more information about font mapping, refer to *Print Services Facility/MVS: System Programming Guide*.

Additional Parameters to Help in Distributing Output

The following user JCL parameters add information to the separator pages printed with an output data set. A sample separator page using these parameters is shown in Figure 26 on page 65. These additional parameters are available only on systems running MVS/ESA SP 4.1.0 and above with MVS/SP-JES2 4.1.0 and above.

If this is a multiple data set job, and each data set has different distribution information on its OUTPUT statement, the distribution information from the OUTPUT statement for the first data set in the job will be in the job header, and the distribution information from the OUTPUT statement for the last data set in the job will be in the job trailer.

If a data set has data-set-header pages (APSUX03), and the distribution information is being printed on them, they will contain the distribution information from the OUTPUT statement for that data set.

ADDRESS=*address*

Specifies up to four 1-character to 60-character delivery addresses for an output data set. You can type a name in the address field when the name associated with an address is not the name associated with the output data set, which is specified in the NAME parameter. This parameter is optional.

EXAMPLES

- This example prints “J. Plant, 1234 Main Street, Boulder, CO 80301” on the separator pages of each data set that references OUTDS2:

```
 //OUTDS2   OUTPUT  ADDRESS=('J. Plant','1234 Main Street',
//           'BOULDER, CO','80301')
```

  The output is printed as follows:

```
J. Plant
1234 Main Street
Boulder, CO
80301
```

- This example prints “57 Fair Lane, Omaha, NE 12121” on the separator pages of each data set that references OUTDS3. The first line reserved for the addressee's name will be blank on the separator page. The zip code does not require apostrophes, because it contains only characters that are valid without apostrophes:

```
//OUTDS3 OUTPUT ADDRESS=(,'57 Fair Lane','Omaha, NE',12121)
```

  The output is printed as follows:

```
57 Fair Lane
Omaha, NE
12121
```

BUILDING=*building*

Specifies the name of a building, containing 1 to 60 characters, on the separator pages of the output data set. This can help in distributing output. This parameter is optional.

EXAMPLE

This example prints “920” on the line reserved for BUILDING on the separator pages of any output data set that references OUTDS4:

```
//OUTDS4 OUTPUT BUILDING='920'
```

DEPT=*department*

Specifies a department name, containing 1 to 60 characters, associated with the output data set. This can help in distributing output. This parameter is optional.

EXAMPLE

This example prints “PAYROLL” on the line reserved for DEPT on the separator pages of any output data set that references OUTDS5:

```
//OUTDS5 OUTPUT DEPT='PAYROLL'
```

NAME=*name*

Specifies a name, containing 1 to 60 characters, that is associated with the output data set. This can help in distributing output. This parameter is optional.

EXAMPLE

This example prints “R. ROPER” on the line reserved for NAME on the separator pages of any output data set that references OUTDS6:

```
//OUTDS6 OUTPUT NAME='R. ROPER'
```

Note: When sending output to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

ROOM=*room*

Specifies the name, containing 1 to 60 characters, of the room to be associated with the output data set, which can help in distributing output. This parameter is optional.

EXAMPLE

This example prints “CONFERENCE ROOM” on the line reserved for ROOM on the separator pages of any output data set that references OUTDS7:

```
//OUTDS7 OUTPUT ROOM='CONFERENCE ROOM'
```

When sending output to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

TITLE=*title*

Specifies a description, containing 1 to 60 characters, of the output data set. This can help in distributing output. This parameter is optional.

EXAMPLE

This example prints “ANNUAL REPORT” on the line reserved for TITLE on the separator pages of any output data set that references OUTDS8:

```
//OUTDS8 OUTPUT TITLE='ANNUAL REPORT'
```

Sample Header Page Created by Using the Distribution Parameters

Figure 26. Sample Header Page with Additional Distribution Information

Specifying JCL Statements for the Parameters

Table 1 on page 66 lists the JCL parameters described in this chapter and shows in which JCL statement you can specify each parameter. For some tasks (such as selecting a printer class), the parameter you specify in the DD statement is not the same as the parameter you would specify in the OUTPUT statement. For other tasks (such as selecting a printer destination name), the same parameter can be specified in either statement. If a parameter for the same task is specified in both a DD statement and an OUTPUT statement, PSF uses the parameter in the DD statement.

Note: If you do not specify a JCL parameter, you can use defaults specified in the PSF startup procedure or the JES initialization statements. For details, refer to *Print Services Facility/MVS: System Programming Guide*.

[13] FCB is for line-printer-applications compatibility.

[14] The UCS parameter is never used in deferred-printing mode under JES3.

Table 1. JCL Parameters for Printing on a Page Printer

Output Parameter	Equivalent Parameter in DD Statement
BURST	BURST
CHARS	CHARS
CKPTPAGE	(none)
CKPTSEC	(none)
CLASS	SYSOUT
COMSETUP	(none)
COPIES	COPIES
DATACK	(none)
DEST	DEST
FCB	FCB[13]
FLASH	FLASH
FORMDEF	(none)
FORMS	SYSOUT
NOTIFY	(none)
OUTBIN	(none)
PAGEDEF	FCB
PIMSG	(none)
PRMODE	(none)
(none)	SEGMENT
TRC	DCB=OPTCD=J
UCS	UCS[14]
USERLIB	(none)

Table 2 shows the additional parameters for distributing output.

Table 2. Additional Distribution Parameters

Distribution Parameter	Equivalent Parameter in DD Statement
ADDRESS	(none)
BUILDING	(none)
DEPT	(none)
NAME	(none)
ROOM	(none)
TITLE	(none)

Chapter 3. Understanding Form Definitions and Page Definitions

PSF uses a *form definition* to control modifying and printing forms. PSF uses a *page definition* to compose pages from line data.

This chapter describes form definitions and page definitions that you can create by using the IBM Page Printer Formatting Aid licensed program (PPFA). You can also create form definitions and page definitions using some other products, but those products may differ from the products described in this chapter.

Form Definitions

Every data set printed on a page printer requires a form definition. A form definition contains 1 or more *copy groups*, each of which consists of a complete set of printing controls.

A copy group controls the printing of a form, or physical sheet of paper. If you are printing on both sides of a sheet, a copy group controls the printing of both sides. If you change copy groups, PSF automatically ejects to a new sheet before using the controls in the next copy group.[15]

Many of your form definitions contain a single copy group. All the pages of the print job are printed according to the same form specifications. However, if your job requires different specifications for different pages of output, you can use a form definition that contains multiple copy groups.

To begin printing a data set, PSF selects the first copy group in a form definition.[16] A copy group (also called a medium map) can be selected by a structured field in a print data set or by conditional processing in a page definition. For more information about selecting copy groups, see “Using Multiple Copy Groups or Page Formats” on page 104. For more information about conditional processing, see “Conditional Processing” on page 85.

[15] When you are using Enhanced N_UP, changing copy groups may *not* cause PSF to eject to a new form.

[16] That is, unless the data set begins with a control record that includes instructions to select a different copy group.

Copy Group

The printing controls that can be specified in a copy group are:

- Page position
- Paper source (bin) on cut-sheet printers
- Printers that support duplex printing
- Duplex page offset
- Multiple pages on a sheet (N_UP printing)
- Constant forms control
- Page presentation
- Offset stacking
- Print quality level
- Horizontal adjustment for the 3800 printer
- Subgroup modifications
  - Page copies
  - Overlays
  - Suppression
  - Forms flash for 3800 printers

Note: When sending output to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

These specifications are described in the following sections.

Page Position

The copy group assigns horizontal and vertical offsets to position the top left corner of the logical page on the physical form (sheet). The page position is relative to the medium origin.

The medium origin is the top left corner of the physical piece of paper (or other medium) being printed on, as seen by the printer microcode. For cut-sheet printers, the medium origin is always the top left corner of a sheet viewed with the short side as the top. For continuous-forms printers that support page presentation compatibility, such as the 3835 and 3900 printers, the medium origin is always the top left corner of the narrow edge of the form. For more information about medium origin, refer to *Advanced Function Presentation: Printer Information*.

The page origin is the starting position of the logical page, which contains the user's print data. The logical page is defined in the page definition for line data, or in structured fields for MO:DCA data. All the data in the print data set must fit within the boundaries of this logical page.

The form definition positions this logical page on the physical form. The page origin is specified as a horizontal offset and a vertical offset from the medium origin. The page could be positioned at the medium origin, but it is usually offset to avoid unprintable areas or areas too near the edges of forms. For information about printable areas for your printer, refer to *Advanced Function Presentation: Printer Information*.

Figure 27 on page 69 shows the relationship between the medium origin and the page origin. The size of the logical page and its print direction are defined in the page definition for line data or in structured fields for MO:DCA data. The offset of the page origin is defined in the form definition. In the figure, the horizontal offset is labeled *X*, and the vertical offset is labeled *Y*.

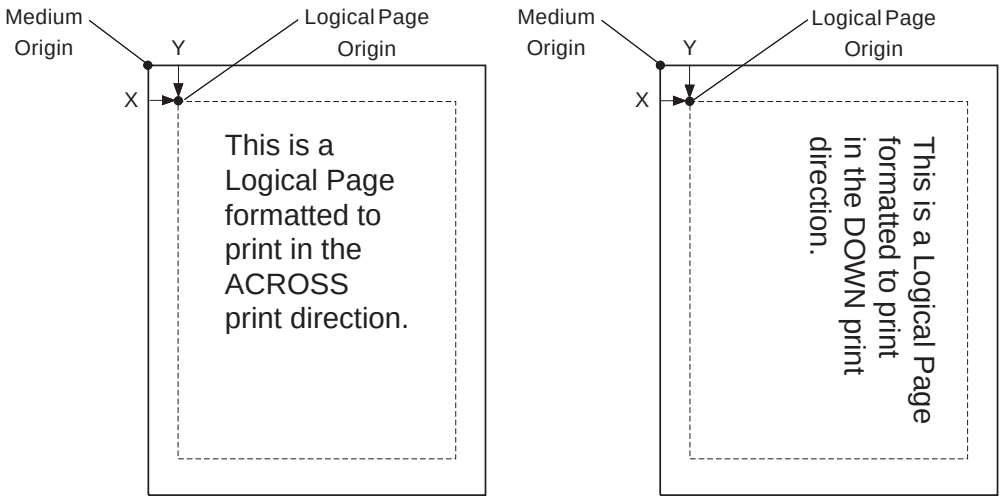


Figure 27. Relationship between the Medium Origin and the Page Origin

Page origin is not affected by the print direction specified in the page. As Figure 27 shows, changing the page print direction from across to down does not change the position of the page origin.

You can specify *different* logical page offsets for the front and the back of a duplex printed page. For additional explanation, see "Duplex-Page Offsets" on page 72.

Selecting a Paper Source on a Printer with More than 1 Source

To specify the paper source, insert a control in the copy group or a subgroup of it[17] (Also see "Subgroups" on page 77.) This control indicates whether paper is to be fed from the primary source or from an alternative source. Printers can have up to 255 alternative paper sources. For more information, refer to *Advanced Function Presentation: Printer Information*.

Note: Some printers support disabled mechanisms. For example, if a form definition specifies the primary source, but that source is disabled, the operator can print the job from an alternate source. The printers that support disabled mechanisms are identified in *Advanced Function Presentation: Printer Information*.

For continuous-forms printers, PSF ignores copy controls for paper selection, because these printers have only 1 paper source.

[17] To specify the paper source in the form definition, use "BIN." For an example that shows how to specify the paper source, see "Specifying BIN (Paper Source)" on page 93.

Specifying Duplex (Double-Sided) Printing

Some printers can print on 1 side of the sheet (simplex printing) or on both sides (duplex printing). For information about printers that can print in duplex mode, refer to *Advanced Function Presentation: Printer Information*.

You specify duplex or simplex printing in a control in the copy group. The following choices are available:

- To print on only 1 side, specify *no duplex*.
- If the sheets are to be bound on the long edge of the paper, as Document A and Document B are in Figure 28, specify *normal duplex*.
- If the sheets are to be bound on the short edge of the paper, as Document C and Document D are in Figure 29 on page 71, specify *tumble duplex*.

Figure 28 shows Document A and Document B printed with normal duplex. Notice that the pages are printed in the *portrait position* for Document A and in the *landscape position* for Document B.

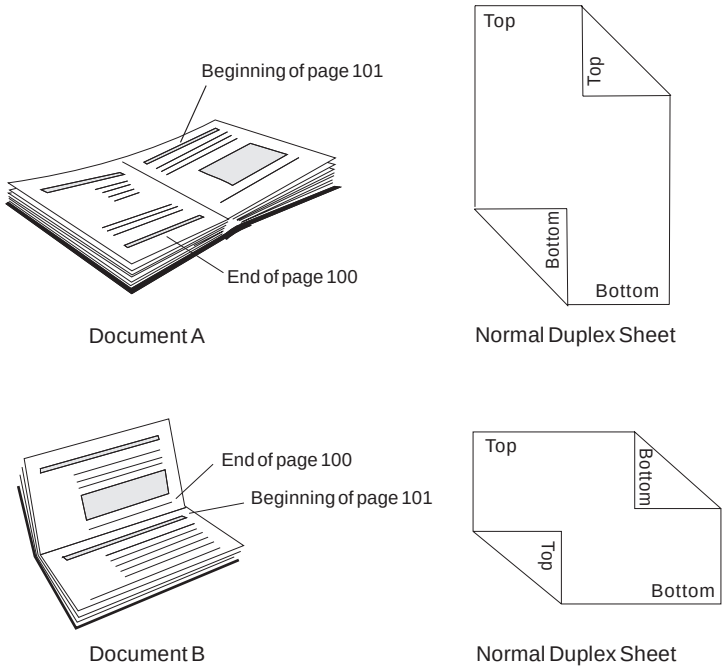


Figure 28. Duplex Documents A and B, Specified as Normal Duplex

Figure 29 on page 71 shows Document C and Document D printed with tumble duplex. Notice that the pages are printed in the *portrait position* for Document C and in the *landscape position* for Document D.

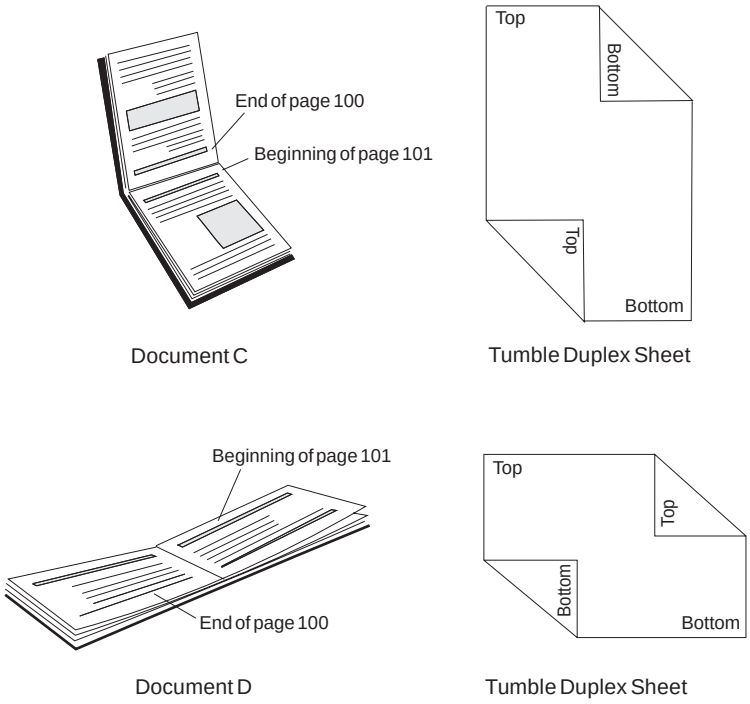


Figure 29. Duplex Documents C and D Specified as Tumble Duplex

Some printers can print despite a disabled duplex paper path; other printers are unable to print in duplex printing mode. For a printer that does not support printing in duplex or that has a disabled duplex path, PSF processes the duplex control in the following ways:

- For no N_UP printing, PSF processes 2 consecutive pages of data as a pair. The first page is processed as the front of the sheet, and any controls specified in the form definition for the front are applied. The second page is processed as the back of the sheet, and any controls specified in the form definition for the back are applied. If you request multiple copies in the form definition, all the copies of front pages are printed first, and then all the copies of back pages are printed.
- For *basic N_UP printing*, PSF processes the first *n* consecutive pages of data as the front of the sheet, applying any controls specified in the form definition for the front. PSF processes the next *n* consecutive pages of data as the back of the sheet, applying any controls specified in the form definition for the back. If you request multiple copies in the form definition, all the copies of the front *n* pages are printed first, and then all the copies of the back *n* pages.
- For *enhanced N_UP printing*, PSF is unable to process the duplex control, because page buffering and page reordering may be required to simulate duplex printing. In this case, the data cannot be printed.

For information about printers that support disabled mechanisms and can print in duplex printing mode, refer to *Advanced Function Presentation: Printer Information*.

Duplex-Page Offsets

You can specify *different* logical page offsets for the front and back of a duplex sheet. You may want to specify different page positions when, for example, the printed pages will be bound or when you are printing on 3-hole punched paper. In these cases, you may want to specify a smaller left margin for the back of the sheet to leave room on the right side of the sheet for the binding or the holes.

N_UP Printing

PSF 2.2.0 and later supports *N_UP printing*, a printer media-handling support that is specified in the form definition. With N_UP printing, you can place multiple pages in partitions on a sheet, enabling you to print much more data on a sheet, saving printer-use costs, paper, and storage space.

N_UP printing differs from the previously provided function called multiple-up printing, in that N_UP printing is specified in the form definition and works with MO:DCA data (page data) and with line data. N_UP printing enables you to place multiple MO:DCA pages or line data on a sheet, and to format each of the N_UP pages differently. In contrast, multiple-up printing is activated in a page definition and works only with line data. The entire multiple-up impression is formatted with a single page format, and only appears to have multiple logical pages.

| To use N_UP printing, you need a form definition that specifies N_UP. (IBM has
| enhanced Page Printer Formatting Aid [PPFA] to enable you to create form
| definitions that specify the N_UP subcommand.) These new form definitions specify the number of pages on a sheet in addition to the other form definition options, such as duplex and page offsets.

The form definition does not control the size of the pages on the sheet. The size of the pages on the sheet is controlled in the page definition for inline data, or in the structured fields in a MO:DCA page.

PSF supports 2 levels of N_UP: *basic* N_UP, and *enhanced* N_UP. Enhanced N_UP is supported only on the newer AFP printers that use the AFCCU control unit; it requires APAR OW03243 and PPFA/370 APAR PN54401. Basic N_UP is supported on most current channel-attached AFP printers. In addition, if you want to enable PSF support for changeable medium origin on a cut-sheet printer that does not support N_UP, you can send that printer a form definition that specifies an N_UP value of 1. Changeable medium origin is controlled by the page presentation options in the form definition. For information on printer support of N_UP and changeable medium origin, refer to *Advanced Function Presentation: Printer Information*.

With *basic N_UP printing*, using a form definition, you can print up to 4 pages on 1 side of a sheet of paper in simplex mode, and up to 8 pages in duplex mode, by creating a new form definition that contains the PPFA N_UP subcommand. With basic N_UP, you accept the default placement of the pages in the partitions. The partitions are all the same size and are placed 1 to 4 per side, depending on the number specified in your N_UP subcommand. The page must be the correct size to fit within the partition area. For basic N_UP, the valid printable area (VPA) is the intersection of the partition and the current logical page. Figure 30 on page 73 shows the equal partitions created on a side of a sheet by including the basic N_UP subcommand in a form definition. The figure shows continuous forms, and cut forms with both wide and narrow leading edges, feeding into the printer.

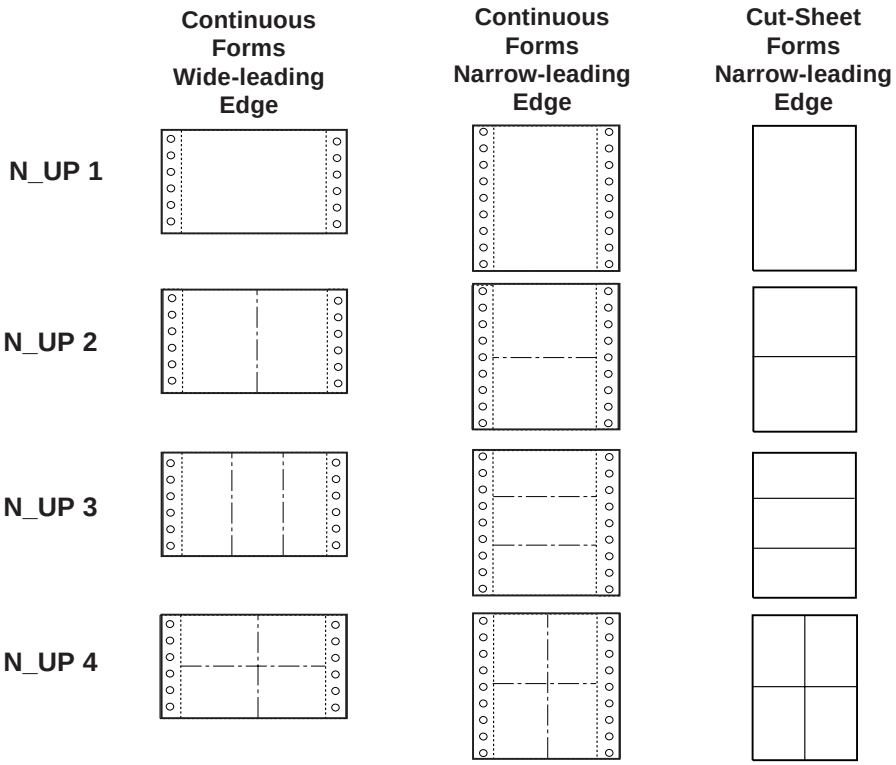


Figure 30. N_UP Printing Partitions for Various Media

With *enhanced N_UP printing*, you can place pages at any location on either side of the sheet. You can also:

- Place a page relative to any partition origin on either side of the sheet, in any orientation, and in any size that will fit on the sheet. You can place multiple pages relative to the same origin, when the total number of pages does not exceed the N_UP limit for that sheet.
- Place overlays relative to any partition origin, with or without variable page data from the application program.
- Specify a different rotation for each page.
- Specify 1 or more different overlays for each page.
- Specify a different offset for each page.

Page Printer Formatting Aid: User's Guide and Reference describes how to use N_UP printing.

Note: When sending N_UP data to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Printing Constant Forms

The *constant-forms* function enables you to print medium overlays or a forms flash (3800 printer only) on blank pages with no need to add blank pages to your print data set; PSF generates the blank pages on which to print the overlays or the forms flash. These pages generated by PSF are called *constant forms* because no variable data from the print data set is printed on the page.

For example, you can print an overlay containing constant text on the back of each page of a print data set without modifying the data set; you simply specify the

constant-forms function in the form definition. You specify the constant-forms function for an entire copy group; you identify the overlays and form flashes in the subgroups of a copy group. See “Subgroups” on page 77.

Note: When printing constant forms to a microfilm device, see Appendix D, “Microfilm Device Considerations.”

Using the constant-forms function, you can request that PSF generate and print the constant form as the front and/or back side of each sheet in the copy group, as follows:

- For the front side of each sheet:

  PSF prints the constant form as the front side of each sheet.

  If duplex printing is specified for the copy group, PSF prints the pages from the print data set on the back side of each sheet. The print data set must contain at least 1 page that is printed by use of this copy group; otherwise, PSF will not generate any constant forms for this copy group.

  If simplex printing is specified for the copy group, the print data set must not contain any pages to be printed by use of this copy group; for subsequent pages in the print data set, a different copy group must be used.

- For the back side of each sheet (duplex printing only):

  PSF prints the constant form as the back of each sheet, and the pages from the print data set on the front of each sheet.

  The print data set must contain at least one page printed by use of this copy group; otherwise, PSF will not generate any constant forms for this copy group.

- For both the front and back side of each sheet (duplex printing):

  PSF prints the constant form as both the front and back sides of each sheet.

  The print data set must not contain any pages to be printed by use of this copy group; a different copy group must be used for subsequent pages in the print data set.

Figure 31 on page 75 shows 2 pages printed by use of a copy group that specifies the constant-forms function for the back side of each sheet. A subgroup in that copy group specified overlay O1CODES for the back side of the sheet. Note that the print data set contains only 2 pages; PSF generates the pages that are printed as the back sides of the sheets.

If the constant-forms function is specified in a copy group, but no overlays or forms flashes are specified in the subgroups of that copy group, PSF generates a blank page. In the example shown in Figure 31 on page 75, the back side of each sheet would be blank.

Maintaining Page-Presentation Compatibility

PSF-supported printers can have different hardware (default media) origins. As a result, for compatibility across PSF-supported printers, you may need to use form definitions that contain page-presentation controls. For a description of the default medium origins for each printer, refer to *Advanced Function Presentation: Printer Information*.

That same publication contains examples of incompatibilities between cut-sheet printers and continuous-forms printers. It also contains figures showing how to use

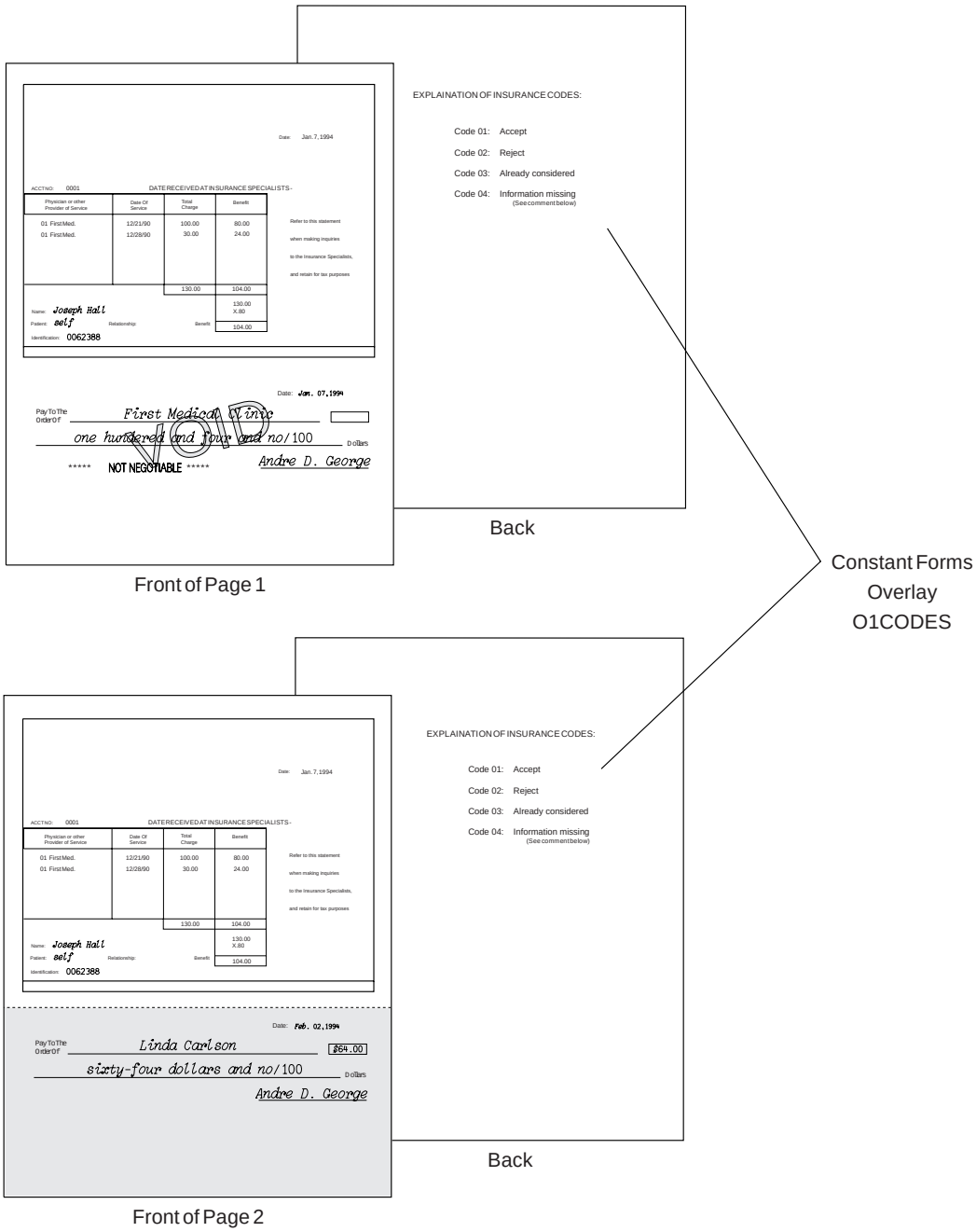


Figure 31. Copy Group Printed by Use of the Constant-Form Function

the PSF-supplied compatibility form definitions listed in Appendix A, “Form Definitions Supplied with PSF.”

You can also build other form definitions for page-presentation compatibility by using PPFA. For more information, refer to *Page Printer Formatting Aid: User's Guide and Reference*.

Offset Stacking of the Printed Output

In offset stacking, the sheets printed according to 1 copy group are stacked perpendicular to the sheets printed according to the preceding copy group. If you want your printed output to be offset-stacked, specify offset stacking in the copy group.

Notes:

1. If the form definition specifies offset stacking, but the stacker is disabled, some printers enable the operator to print the job without offset stacking. *Advanced Function Presentation: Printer Information* identifies the printers that support disabled mechanisms.
2. You can specify that data sets are to be offset-stacked from each nother by using codes in the JESDS parameter in the OUTPUT statement for the job.

If you want the printed output from a continuous-forms printer to be offset-stacked, the printer must be equipped with a burster-trimmer-stacker (BTS) feature or an equivalent post-processing device. If your continuous-forms printer does not have a BTS feature but supports edge-marking, the printer will change the edge-markings on the sheets that you specify to be offset-stacked.

The following example shows how to specify offset stacking. Assume that you have a 5-sheet data set to be printed with the same set of printing controls, but that you want sheet 3 offset from sheets 1 and 2, and you want sheets 4 and 5 offset from sheet 3.

1. Create a form definition containing 2 copy groups that are identical except that copy group A does not specify offset, and copy group B does.

   Notes:

   a. The printer offsets a copy group relative to the previous copy group. If you specify copy group A (which does not specify offset) for sheets 4 and 5, those sheets will not be offset from sheet 3.

   b. Because the same offset is used for all the sheets in a group, they are in 1 stack.
2. Copy group A is the default copy group. PSF selects it for sheets 1 and 2, which will not be offset from any sheets preceding them.
3. For page 3, offset stacking is specified by either of the following:

   - An Invoke Medium Map (IMM) structured field specifying copy group B, inserted at this point in the data set
   - Conditional processing, specified in the page definition

   Whichever method is used, sheet 3 will be offset from sheets 1 and 2.
4. At the start of sheet 4, copy group B (the copy group that specifies offset stacking) is selected again, by either of the methods just described. Sheets 4 and 5 will be offset from sheet 3, and will be stacked together.

Print Quality Level

With some printers, you can select different levels of print quality, such as draft or near-letter quality. For higher print quality, printing speed is slower.

To specify the print-quality level, insert a control in the copy group of the form definition.

If you specify a quality level for a printer that supports only 1 quality level, and if the quality level you specify is not that level, PSF sends an error message and ignores that specification. For information about the levels of print quality that your printer supports, refer to *Advanced Function Presentation: Printer Information*.

Specifying the Horizontal-Adjustment Value for the 3800 Printer

Specify the horizontal-adjustment value as a control in the copy group. This value indicates both the starting print position and the amount of space by which the 3800 printer operator can adjust the position of the printed data to the left or right. Specify the adjustment value as a number of picture elements (pels). For information about the adjustment values for the 3800 printer, refer to *Advanced Function Presentation: Printer Information*.

Subgroups

A copy group contains 1 or more subgroups, each of which can contain specifications for different versions of a page. In a subgroup, you can specify what modifications are to be made to a page and how many copies of each version are to be printed. The sum of the number of copies specified in the subgroups is the total number of copies of each page to be printed.

In a subgroup, you can specify the modifications listed below. For simplex printing, you can code a subgroup to specify the modifications for a single page in that subgroup. For duplex printing, you can code a subgroup to specify 1 set of modifications for both sides of the sheet, or you can code 2 subgroups to specify a different set of modifications for each of the 2 sides. The modifications are as follows:

Identifying Overlays: You can identify the names of up to 8 medium overlays for each side of a sheet. In Figure 32 on page 78, the INVOICE overlay was printed for the first subgroup, and the PACKLIST overlay was printed for the second subgroup.

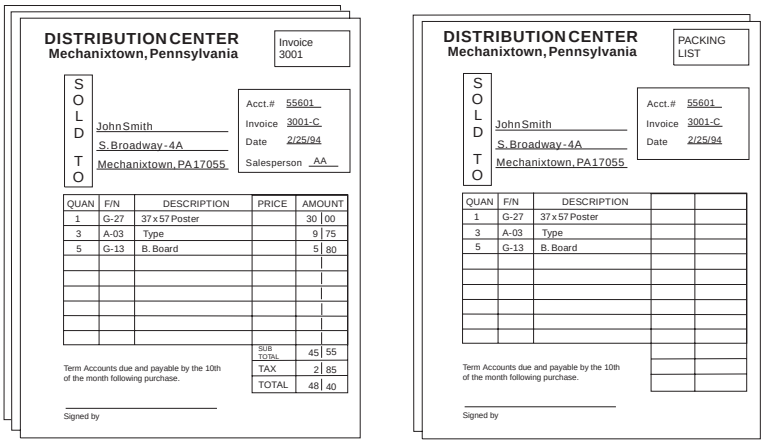
Suppressing Data: You can identify up to 8 suppression names (names identifying fields that are not to be printed), provided they have been defined in the page definition. In Figure 32 on page 78, no suppression names were specified for the first subgroup, and 3 suppression names were identified for the second subgroup: Salesperson, PRICE, and AMOUNT.

Specifying Paper Source: In a form definition created using the PPFA program, you can include the BIN subcommand in the subgroup to specify the paper source. Use this subgroup subcommand only for printers with more than 1 media source and that support bin selection in a subgroup.

Specifying Forms Flash for the 3800 Printer: You can specify whether the 3800 printer forms-flash unit prints its negative on each sheet in a subgroup. For the other page printers, PSF ignores this control.

You can also specify that PSF is to print overlays or forms flashes without printing any variable data on certain pages of your output. For example, you can specify that a constant overlay is to be printed on the back of each page of a duplex print job. See “Printing Constant Forms” on page 73.

As an example, Figure 32 shows a page in a data set printed using a copy group containing 2 subgroups. In the first subgroup, 3 duplicate sheets were printed with a set of modifications; in the second subgroup, 2 duplicate sheets were printed with a different set of modifications. The 2 sets of sheets were generated from the same page of information in the data set but were modified differently.



First Subgroup Second Subgroup

Figure 32. Subgroups Printed from 1 Page of the Data Set

Specifying FORMDEF and COPIES or FLASH Parameters in JCL

If you specify a form definition by coding the FORMDEF parameter in the JCL OUTPUT statement, the number of copies is determined by the specifications in the subgroups of a copy group. The JCL COPIES parameter is used only to determine how many times the data set is to be transmitted.

The following 2 examples show the effects on the number of copies printed when the JCL COPIES parameter is coded and the number of copies is specified in the form definition selected by FORMDEF. “COPIES” on page 47 shows how to code the JCL COPIES parameter.

For the first example, assume that you have a 2-page print job using a form definition that includes a copy group containing 2 subgroups. The first subgroup calls for 2 copies with a particular set of modifications, and the second subgroup calls for 1 copy with another set of modifications. In the JCL, you would specify COPIES=3. The data set will be transmitted to the printer 3 times, and you will get 3 collated copies of the entire job. The printed output would be as shown in Figure 33 on page 79. Nine copies of page 1 are printed, and 9 copies of page 2 are printed.

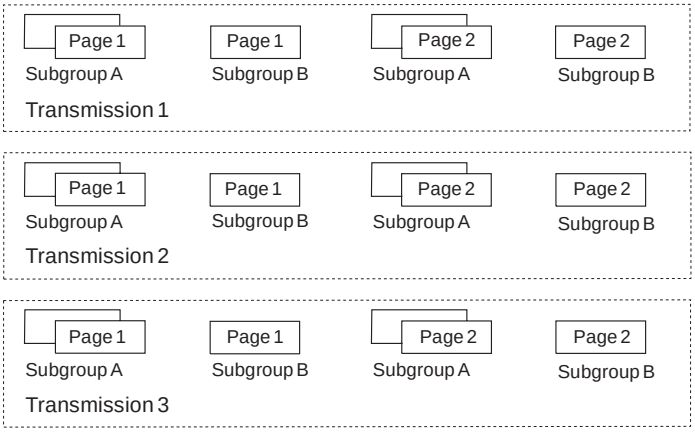


Figure 33. Output from 3 Transmissions of a 2-Page Data Set

For the second example, assume that you use the same form definition as in the first example: the first subgroup specifies 2 copies, and the second subgroup specifies 1 copy. In the JCL, specify COPIES=(,(3,1,5,2)).

The 2-page data set is sent to the printer 4 times, because 4 *groupvalues* (3,1,5,2) are specified; see page 47. However, those groupvalues have no effect on the number of copies printed, because that number is determined by the form definition specified in the JCL FORMDEF parameter; the number of copies printed would be exactly the same if you used an entirely different set of groupvalues. When the data set has printed, 12 copies of page 1 are printed, and 12 copies of page 2 are printed, as Figure 34 shows.

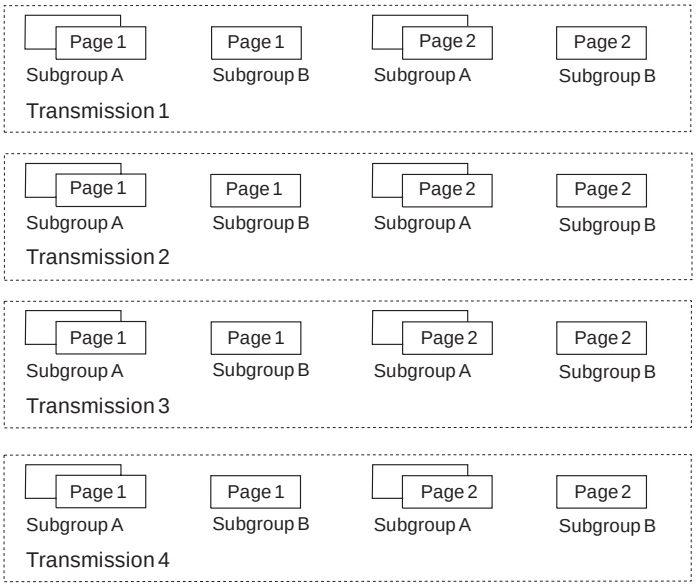


Figure 34. Output from 4 Transmissions of a 2-Page Data Set

Specifying COPIES without a FORMDEF Parameter in JCL

If PSF uses the default form definition because you did not code the FORMDEF parameter, you can use the JCL COPIES parameter to specify the number of copies to be printed. The system interprets the COPIES parameter variables, *nnn* and *groupvalue*, as described under “COPIES” on page 47.

If you specify group values in the COPIES parameter, PSF uses only the following modifications in the default form definition's subgroup:

- Maximum horizontal adjustment
- Offset stacking
- Edge marking

PSF ignores other modifications. If you specify group values in your JCL, the values override the copy group in the default form definition. A default form definition modified by the groupvalue variable in the COPIES parameter is classified as a modified-default form definition. If you omit the COPIES parameter, and if no copy group values are specified in the active copy group in the form definition, PSF prints a single copy.

Specifying FLASH without a FORMDEF Parameter in your JCL

If PSF uses the default form definition because you did not code the FORMDEF parameter, you can use the JCL FLASH parameter to specify whether the forms flash unit on your 3800 printer is to be used for your print job. The default form definition modified by the FLASH parameter is classified as a modified-default form definition. For more information about the FLASH parameter, see “Using a Forms Flash on a 3800 Printer” on page 94.

PSF uses only the following modifications in the default form definition subgroup:

- Maximum horizontal adjustment
- Offset stacking
- Edge marking

PSF ignores other modifications. If you omit the FLASH parameter, PSF uses the default form definition without modification.

Form Definitions Supplied with PSF

PSF provides general-purpose form definitions, described in Appendix A, “Form Definitions Supplied with PSF.” These form definitions specify a page position at the top of the page; 1 copy of each page is printed, on 1 or both sides, with no modifications.

Page Definition

A *page definition* is the resource that contains formatting specifications for line data. This resource is required for any data set or any part of a data set that is not already composed into pages when PSF receives it. No page definition is used for data that is already composed into pages. If you specify a page definition for MO:DCA data, PSF ignores it.

A page definition contains one or more *page formats* (also known as data maps), each of which contains a complete set of page formatting specifications.

A page format controls the printing of an entire page (a physical sheet of paper) if you are printing on only 1 side, or 1 *side* of a sheet of paper if you are printing in duplex (on both sides of the sheet). If you change page formats, PSF automatically ejects to a new page before using the controls in the next page format.

Many of your page definitions will have only 1 page format. All the pages of the print job will be printed to the same specifications. If your job requires different specifications for different pages of output, you can use a page definition that contains multiple page formats.

To begin printing a data set, PSF selects the first page format in a page definition. You can specify a page format by inserting a structured field in a data set or by using conditional processing in the page definition. For more information about conditional processing, see “Conditional Processing” on page 85.

To change from 1 page format to another, you can use the Invoke Data Map (IDM) structured field, as described under “Using Multiple Copy Groups or Page Formats” on page 104.

Page Format

A page format contains formatting controls for your data set that indicate where and how text, and optionally page overlays and page segments, are to be placed on the page. The page format is defined relative to the origin of the sheet specified in the form definition. A page format must contain the following information:

- The size of the page area to be formatted, specified as the width and height of the page
- The print direction
- The starting position for the first print line or record
- The line spacing for the page, specified in lines per inch
- The color in which to print, if the printer supports color

Page Size

The page size, width, and height are defined in the page format. This area is called the *logical page*. All the text and images contained in your print data set must fit within the boundaries of this logical page.

The logical page should cover the entire area of the physical form on which your data can print. If you are printing *multiple-up* applications (in which 2 or more pages of application data are formatted on the same side of a physical form), you must make your logical page large enough to contain all the pages of application data. This applies only to multiple-up formatting for which a page definition is used; it does not apply to N_UP printing.

Characters or images that extend outside the boundaries of the logical page cannot be printed. Whether you receive error messages indicating the characters or images were not printed depends on the value in the DATACK parameter. For more information, see the description of the DATACK parameter on page 49.

Be careful when positioning text near the top or bottom of the logical page area. The position you specify for character data in your page definition is the position at

which the baseline of the characters will be printed.[18] When you position character data in a page definition, be sure to leave room for the characters and their ascenders and descenders. For example, never place a character 0 inches down from the top of the page.

You should also be careful not to extend data off the right side of the page. This can happen if your print lines are too long, or if you have used a font that is too large to fit within your page area.

In most cases you do not have to be concerned with placing characters too near the left margin of the page. Characters are positioned from the front or leading edge of their baseline, so that a character can be placed at the exact left edge of the logical page. Italic fonts, however, can extend off the left edge of the page. This is because italic characters are *kerned* to tuck under the adjacent characters. Some italic characters extend a few pels to the left of their character space origin. Therefore, when you print with an italic font, be sure to position the text line a few pels to the right of the left edge of the logical page.

For more information about character baselines and positioning, refer to *IBM AFP Fonts: Font Summary*.

Print Direction

The page format assigns a print direction to the lines of text in the logical page. The print direction can be 1 of 4 inline directions supported by AFP printers:

- ACROSS—oriented 0° from the page origin
- DOWN—oriented 90° from the page origin
- BACK—oriented 180° from the page origin
- UP—oriented 270° from the page origin

In addition to print direction, AFP text can be printed in four character rotations: 0°, 90°, 180°, and 270°. Each of these character rotations is specified relative to the print direction. Unless you specify otherwise, a 0° rotation will be used with any print direction you select for your page definition. For more information about print direction and character rotation, see “Fonts for Different Print Directions and Character Rotations” on page 24.

Not all AFP printers can print in every combination of print direction and character rotation. For information about the print orientations your printer supports, refer to *Advanced Function Presentation: Printer Information*.

Figure 14 on page 25 shows the different print directions possible on AFP printers. For information about the relationship between print direction and page origin, see “Page Position” on page 68.

[18] The baseline is the imaginary line on which characters sit. The character extends above the baseline. Characters such as the lower-case “g” or “y” have pieces called *descenders*, which extend below the baseline.

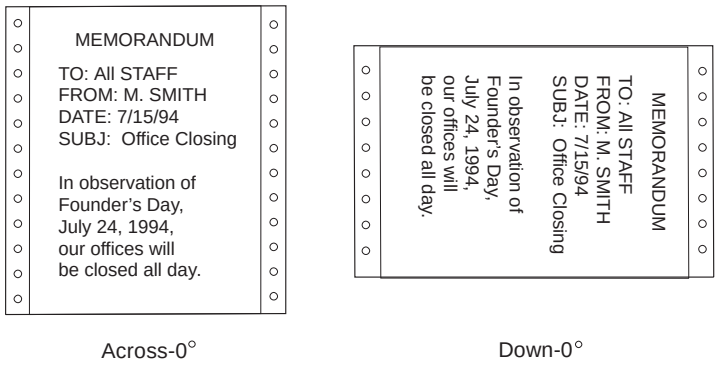


Figure 35. Pages Printed in 2 Directions on Continuous-Forms Paper

Page Format Options for Formatted Print Records

When the print data set contains records that are already formatted into output print lines, you can specify additional formatting options in the page format. For this type of data stream, you can use the page format to:

- Specify the number of print lines per page
- Specify the number and starting position for a group of lines on a page
- Specify how carriage controls in the data are to be processed
- Specify the position on the page at which channel codes in the data are to start
- Change line spacings for groups of lines within the page
- Select fonts for the entire page or for groups of lines on a page
- Provide a list of fonts for selection by table reference characters (TRCs) in the data
- Change print directions of 1 or more lines
- Change the character rotation of a font for 1 or more lines
- Define and place fields of constant text on each page with the variable data
- Define conditional processing tests on the input record to effect formatting changes
- Provide a list of page segments to be mapped in the printer
- Provide a list of page overlays to be included by the print data set
- Select multiple colors, if the printer supports multiple colors

  Note: Not all printers support multiple colors. For information about color support for your printer, refer to *Advanced Function Presentation: Printer Information*

Page Format Options for Formatting Fields

You can use the page definition not only to place records that have been formatted into print lines, but also to format individual fields within print records. The page format specifies the position, the print direction, and the font for each individual field that you want printed. You can also prevent fields in the print record from printing, by simply not specifying them in the page format. You can use field formatting to change the output for formatted print records that you are currently printing, or you can use field formatting to print unformatted print records that are not formatted into print lines but that contain only the data to be printed.

This capability of AFP enables an application program to generate *only* the variable data to be printed. The data fields in the print record can be generated in any sequence by the program. Specifications in the page format enable you to use each field more than 1 time, if you so choose, and to place a field at any position

on the page. This makes maintaining applications easier, because the format of the printed output can be changed without affecting the application program. In addition, as long as all of the necessary data fields are contained in the print records, new report formats can be created from an existing application by simply creating a new page definition or page format.

When you format fields from application print records, the page format can be used to:

- Define conditional processing tests based on individual fields in the data
- Specify the starting position and length in the print record for each field to be formatted
- Place each field on the page
- Specify different print directions for fields within the page, and, optionally, character rotation
- Select fonts for each field
- Identify fields that can be suppressed (that is, not printed) on some copies of output or in some transmissions of the output data set
- Define and place fields of constant text on each page with the variable data
- Select a color for each field, if the printer supports multiple colors

  Note: Not all printers support multiple colors. For information about color support for your printer, refer to *Advanced Function Presentation: Printer Information*.

Formatted and unformatted print records can be used together in an application program to add formatting flexibility.

Specifying Fonts

You can specify fonts in a page format in several ways:

- You can name a single font for printing all the text on the page.
- You can name different fonts for use with groups of lines, a single line, or data fields in a record.
- You can add a character rotation parameter to a specified font to rotate characters in the inline print direction, as described in “Fonts for Different Print Directions and Character Rotations” on page 24.
- You can specify fonts in a font list for selection by including table reference characters (TRCs) in the data.

If your JCL specifies a page definition that specifies 1 or more fonts, those fonts will be used instead of fonts named in the JCL CHARS or UCS parameter. If you want to select fonts with a JCL parameter, do not use a page definition that names fonts.

If you are using the default page definition for a printer, you can select fonts with the JCL CHARS or UCS parameter, *even if* the default page definition specifies a font.[19] A default page definition modified by the CHARS or UCS parameter is classified as a modified-default page definition.

[19] This statement has 1 exception: If an FCB is named in the JES initialization statements for the printer, and your system programmer has created a page definition by the same name, which names a font, that page definition will be the printer default, and the font named in it cannot be overridden by CHARS or UCS in JCL.

If you are using table reference characters (TRCs) to select fonts, you can specify the fonts either in a font list in the page format, or in the JCL CHARS parameter. For rules that may affect your decision about where to specify fonts, see “Using Table Reference Characters to Select Fonts” on page 121.

You can use the PPFA FONT command to select either resident fonts or outline fonts. You can also specify scaling information and size for outline fonts. The line spacing should be coordinated with the size of the font you are using.

Conditional Processing

The *conditional-processing* function enables you to:

- Define tests to be performed on fields in selected input records. These tests are called *conditions*.
- Specify *actions* that you want PSF to perform when certain conditions are met.

The actions PSF can conditionally perform are:

- Activating a different copy group. You can do this to change options such as offset stacking and bin selection.
- Activating a different page format. You can do this to change options such as print direction and line spacing.
- Starting a new page.
- Starting a new sheet.

You can also specify when PSF is to perform the conditional actions:

- Before the current line is formatted
- Before the current subpage is formatted
- After the current line is formatted
- After the current subpage is formatted

Note: A subpage is a logical page; or, in multiple-up applications, a subpage can be part of a logical page. For a description of subpages in multiple-up applications, see “Conditional Processing for Multiple-Up Applications.”

Because you can specify that PSF is to change processing options *before* the current line or subpage, you can test an input record and change the processing options before that record is printed. For example, you can specify that when the contents of a certain field in a record are different from the contents of the corresponding field in the preceding record, PSF should start printing on a new page.

Conditional Processing for Multiple-Up Applications: The printable portion of a sheet can be divided into sections called *subpages*, each having the appearance of a smaller printed page, as in Figure 36 on page 86. Subdividing the printed page in this way is called *multiple-up* printing. In the figure, the MO:DCA data is printed with subpages to give the *appearance* of 4 separate pages. Do not confuse multiple-up printing with N_UP printing, which places up to 4 *actual* logical pages of data on 1 side of a sheet.

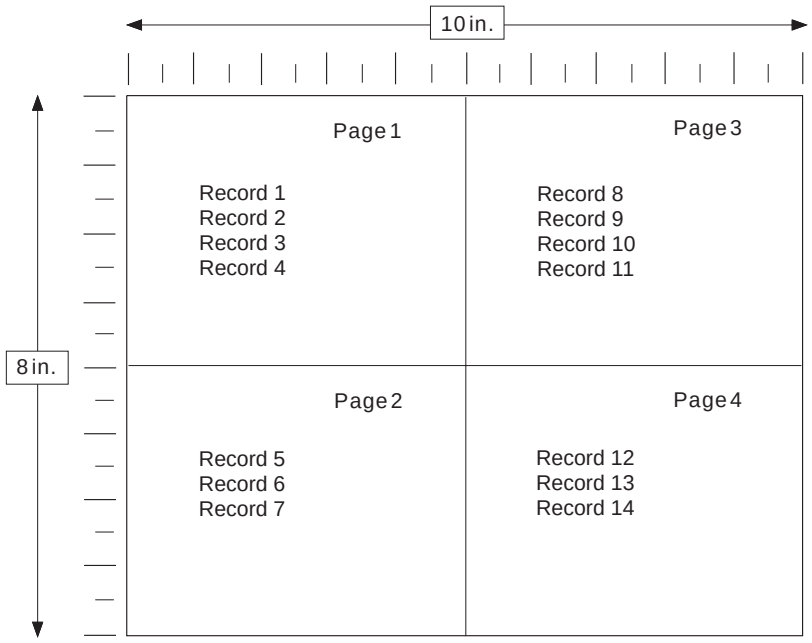


Figure 36. One Logical Page Divided into 4 Subpages

You can use multiple-up processing with or without defining sections of the page as subpages. However, you must define subpages to perform conditional processing either before or after 1 of the subdivisions (subpages) of a multiple-up page.

With the IBM Page Printer Formatting Aid program, you can identify the input records that form a subpage. For more information, refer to *Page Printer Formatting Aid: User's Guide and Reference*.

Multiple Conditions: With conditional processing you can specify more than 1 condition and corresponding action for the same input record. You can also specify conditions and actions for more than 1 line in a subpage. If your page definition contains such multiple conditions, more than 1 of the conditions may be met.

PSF handles multiple conditions as follows:

- If you define multiple conditions for the same input record, PSF will perform the specified action for the *first* condition that is met. PSF will not perform any other conditional actions for that input record.
- If you define conditions for more than 1 input record in the same subpage, and if a condition is met that specifies that PSF is to take an action *after* the current subpage is formatted, PSF will not perform any other conditional actions for subsequent records in that subpage.

Reprocessing: *Reprocessing* occurs when PSF is requested to perform an action either before the current line is formatted or before the current subpage is formatted. In these cases, PSF must reprocess 1 or more input records, perhaps using a new copy group or page format.

While reprocessing records, PSF does not perform some conditional actions that would require additional reprocessing. Because of these restrictions, you may get unexpected results.

When PSF reprocesses records because a condition specified that PSF was to perform an action before the current line, the following restriction applies:

PSF does not perform actions specified to occur either before the line or before the subpage. This restriction is in effect only for the 1 input record being reprocessed.

When PSF reprocesses records because a condition specified that PSF was to perform an action before the current subpage, the following restrictions apply:

- PSF does not perform actions specified to occur before the current subpage. This restriction is in effect for all of the input records in the current subpage.
- PSF does not perform actions specified to occur before the current line. This restriction is in effect only for the first input record in the current subpage.

Notes:

1. The reprocessing restrictions apply even if you specify that PSF is to change to a new page format before reprocessing the input records.
2. Do not use the JCL SEGMENT parameter and conditional processing at the same time. The JCL SEGMENT parameter causes a data set to be broken up into multiple small data sets, and conditional processing is not supported across data set boundaries. If you use the functions at the same time, the results may be unpredictable.

Page Definitions Supplied with PSF

General-purpose page definitions for printing line data on some of the more common paper sizes are provided with PSF. These page definitions are described in Appendix B, “Page Definitions Supplied with PSF.”

Also supplied are page definitions converted from the FCB modules that were provided with the 3800 Printer Model 1 (FCB3STD1, FCB3STD2, FCB3STD3, FCB36, and FCB38). FCB modules are used with line printers to define the vertical format of printed output: lines per inch, skipping and spacing, and length of form. If you have created customized FCB modules for your line printers, you must convert the FCBs to page definitions for use by a page printer. For information about the page definitions in your system libraries, consult your system-support group.

Chapter 4. Printing Tasks and Examples

This chapter shows detailed examples of tasks requiring PSF to print data sets on AFP printers. If you need detailed reference information about using the JCL commands, see Chapter 2, “Using JCL for Advanced Function Presentation.”

The following is a list of the examples of printing tasks in this chapter:

Printing on an AFP Printer

To print on an AFP printer, you send your MVS print job to the AFP printer that your system programmer has defined to your MVS system and to PSF. For most jobs, you can use the same JCL that you would use for non-AFP printers, changing only the MVS routing information to direct your job to the correct printer. If you do not require special AFP functions, you do not have to specify any AFP options for your print job; you can use the PSF defaults defined for your printer.

The following examples show how to select a printer by specifying the JCL class and destination parameters.

You specify an output class either with a SYSOUT parameter in the DD statement for the data set, or with a CLASS parameter on an OUTPUT statement assigned to that DD statement. You must specify a class in 1 statement or the other.

In the first example, a single printer is defined for an output class; therefore, the CLASS parameter is all you need to specify. However, if an output class contains a group of printers, you also need to include a destination (DEST) parameter to select a specific printer from the group, as in the second example. You can specify DEST either in a DD statement or in an OUTPUT statement. If you do not specify the DEST parameter, and the output class contains more than 1 printer, the system selects the first available printer in the group.

EXAMPLES

- The first example shows how you select a printer with print class defined as A, using only a SYSOUT parameter in the DD statement.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//DD1 DD SYSOUT=A
```

- The second example shows how you select a specific 3820 printer from a group of printers all of which are defined as CLASS B. We want the 3820 printer with destination name REMOTE1. For this example, the DEST option is coded on the OUTPUT statement.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//OUT2 OUTPUT DEST=REMOTE1
//DD2 DD SYSOUT=B,OUT=(*.OUT2)
```

In both of these examples, the job prints according to the AFP defaults that PSF has specified for the printer. For jobs requiring special AFP options, see the following examples.

Specifying a Form Definition

A form definition is a resource that defines numerous parameters, such as:

- The placement of a page of data on a form
- The number of copies of a page
- Any modifications to that copy group
- Whether you want a copy group stacked offset from the preceding group
- Whether you want any data fields suppressed (not printed)

Using the forms completed with your installation's defaults, read the descriptions of the form definitions available to you. Also see Appendix A, “Form Definitions Supplied with PSF,” for descriptions of form definitions provided with PSF. On the basis of this information, you can decide which form definition best meets your present printing requirements.

If your job requires special form-definition options, use the JCL for the print job to specify a form definition that contains those options. The name of the form definition, without its 2-character prefix of “F1,” is coded in the FORMDEF parameter of the OUTPUT statement.

The form definition must be in a PSF system library, in a PSF user library (if 1 is supported for your system), or inline as part of the print data set. You can create your own form definition or use a form definition already available on your system.

PSF provides standard form definitions, which are listed in Appendix A, “Form Definitions Supplied with PSF.” The form definition provided for the 3800 printer offsets the page down 0.5 inch to avoid the unprintable area of that printer. Form definitions for other AFP printers offset the page 1/6 inch across and down from the medium origin and provide different combinations of duplex and paper source options. These form definitions are referenced in the following examples.

EXAMPLES

- The first example specifies a PSF-supplied form definition named F1A10111 to specify duplex printing on a cut-sheet printer. Notice that the prefix “F1” is not coded in the JCL.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT FORMDEF=A10111
  //DD1   SYSOUT=A,OUT=(*.OUT1)
  ```

  Form definition F1A10111 is described in Appendix A, “Form Definitions Supplied with PSF.”

- The next example specifies a user-created form definition named F1MYFDEF, as well as a page definition named P1MYPDEF. For more information about specifying page definitions, see “Specifying a Page Definition” on page 98.

  ```
  //AFPUSERB JOB ...
  //STEP1 EXEC PGM=USERB
  //OUT1  OUTPUT FORMDEF=MYFDEF,PAGEDEF=MYPDEF
  //DD1   SYSOUT=A,OUT=(*.OUT1)
  ```

  To create a form definition, use an AFP utility product such as IBM Page Printer Formatting Aid (PPFA). For instructions on how to create a form definition specifying the options you require, refer to the publications for your utility product.

The FORMDEF parameter can specify the resource name of the inline form definition or can specify the keyword DUMMY. If the name in the FORMDEF parameter does not match the name of an inline form definition, PSF uses the resource from the resource library that matches the name in the JCL. If the job does not specify the FORMDEF parameter, PSF uses the first inline form definition in the print data set.

- In the next example, form definition F1MYFDEF is coded inline in the print data set generated by program USERC. For information about coding inline resources, refer to *Advanced Function Presentation: Programming Guide and Line Data Reference*.

```
//AFPUSERC JOB ...
//STEP1 EXEC PGM=USERC
//OUT1  OUTPUT FORMDEF=DUMMY
//DD1   SYSOUT=A,OUT=(*.OUT1)
```

- To use a form definition from a PSF user library, see “Printing with Resources from a User Library” on page 112.

Specifying Duplex Printing

If your AFP printer is capable of printing in duplex mode (printing on both sides of the paper), use the form definition to control duplexing for your print job. For additional information, see “Specifying a Form Definition” on page 91.

Printing in duplex saves paper as well as the space required for storing blank forms and printed documents. You can compound the benefits of multiple-up printing by printing more than 1 page of application data on 1 side of a sheet of paper. For more information, see “Multiple-Up Printing” on page 101.

EXAMPLE

- This job uses PSF-supplied form definition F1A10112 to print tumble duplex on paper selected from the primary paper source.

```
//DUPLEX JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT FORMDEF=A10112
//DD1 DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

  Form definition F1A10112 is described in Appendix A, “Form Definitions Supplied with PSF.”

If your output requires different page margins on the front and back of the form for binding, you can create a form definition that specifies different page offsets for the front and back of the page. See “Duplex-Page Offsets” on page 72.

Specifying BIN (Paper Source)

Some AFP printers have multiple bins (paper sources) from which you can select the paper for printing your job. To control selecting the paper source, use the form definition. For additional information, see “Specifying a Form Definition” on page 91.

EXAMPLE

- This job uses form definition F1A10120, supplied with PSF, to select paper from the alternate paper source.

```
//ALTBIN JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT FORMDEF=A10120
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

  Form definition F1A10120 is described in “Form Definitions for Printers Other than the 3800, PCL4, and PPDS Printers” on page 135.

Changing the Paper Source in a Document

In the example above, all the pages of the document will be printed on paper from the same source. You might prefer to print your document on white paper from the main paper source and use sheets of colored paper from the alternate source as internal dividers.

To do this, use a form definition that contains multiple copy group, and tell PSF when to use which copy group. For an example using a form definition that contains multiple copy groups, see “Using Multiple Copy Groups or Page Formats” on page 104.

Using a Forms Flash on a 3800 Printer

A forms flash is a 3800 hardware frame that prints a photographic negative on selected forms. The following examples show how to use a forms flash on a 3800 printer.

EXAMPLES

- This example uses a forms flash named LOGO. The JCL requests that the forms flash be printed on 15 copies of the output. The COPIES parameter requests only 10 copies, with LOGO printed on each.

  ```
  //DD1 DD SYSOUT=A,FLASH=(LOGO,15),COPIES=10
  ```

- If the number of copies to be flashed is smaller than the total number of copies requested, the overlay is printed only on the number of copies specified by the FLASH parameter, beginning with the first copy. Therefore, if you are printing 15 copies and specify a count of 10 overlays, the overlay is printed on the first 10 copies. An example of this is shown in the following JCL statement:

  ```
  //OUT1 OUTPUT FLASH=(LOGO,10)
  //DD1 DD SYSOUT=A,COPIES=15,OUTPUT=*.OUT1
  ```

- In the next example, the JCL specifies form definition F1UFLASH, a user-created form definition that specifies that a forms flash be used. The name of the flash to be loaded and the number of copies on which to print the flash are provided by the JCL FLASH parameter. The operator is notified to load the flash named FL001, which will be used on the first 2 transmissions of the data set.

  ```
  //OUT2  OUTPUT FLASH=(FL001,2),FORMDEF=UFLASH
  //DD2 DD SYSOUT=A,COPIES=3,OUTPUT=*.OUT2
  ```

  If the JCL specifies a form definition that does not specify a forms flash, such as the IBM-supplied form definition F1A10110, PSF ignores the FLASH parameter.

Printing with Overlays

Electronic overlays can be invoked by the form definition. In addition, with PSF/MVS V2, you can invoke an overlay by using the OVERLAY subcommand in PPFA. You can name and position overlays on the PRINTLINE command, so that at print time, PSF can access them and merge them with the variable data. Doing this eliminates the need to code IPO structured field records in the print application.

An overlay invoked by a form definition is called a *medium overlay*, and an overlay invoked by an Include Page Overlay (IPO) structured field is called a *page overlay*. For more information, see “Overlays” on page 29. The following is a list of some typical jobs that can be performed by using medium overlays and page overlays.

You can use a medium overlay to do such things as:

- Print the same overlay on every page of a data set.
- Print different overlays on different copies of the same page.
- Print different overlays on front and back of a duplexed data set.
- Print an overlay on either the back or the front of a duplexed form.
- Print an overlay on the back of a duplexed form, without printing application data on the back. For additional information, see “Printing Constant Forms” on page 73.
- Print different overlays on different pages of a document by using medium overlays defined in multiple copy groups. For an example of using multiple copy groups, see “Changing Formatting within a Document” on page 104.

For an example of using a medium overlay, see “Printing a Medium Overlay.”

You can use a page overlay to do such things as:

- Print different overlays on different pages of the same job
- Print the same overlay at different positions on different pages of the data set
- In multiple-up applications, use different arrangements of overlays on the “subpages” of a physical form

For an example of using a page overlay, see “Printing a Page Overlay” on page 96.

Printing a Medium Overlay

A medium overlay is invoked in the form definition. Printing a medium overlay requires the following:

1. An overlay resource, built by an AFP utility such as IBM Overlay Generation Language/370 (OGL/370). For more information, refer to *Overlay Generation Language/370: User's Guide and Reference*.
2. A form definition that names that overlay for printing. You can build the form definition by using an AFP utility such as IBM Page Printer Formatting Aid (PPFA). For more information about building form definitions, refer to *Page Printer Formatting Aid: User's Guide and Reference*.
3. The print job JCL that names the form definition.

EXAMPLE

This example uses a user-created form definition, called F1USEROV, that specifies an overlay to be merged with the print data.

```
//AFPUSER JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT FORMDEF=USEROV
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

Printing a Page Overlay

A page overlay is invoked either by adding the Include Page Overlay (IPO) structured field in the print data or by referencing the overlay in the page definition used for line data. Printing page overlays with a line-data application requires the following:

- First create an overlay resource. You can use, for example, an AFP utility such as IBM Overlay Generation Language/370 (OGL/370). For more information, refer to *Overlay Generation Language/370: User's Guide and Reference*.
- Next use the overlay, by either of the following methods:
  - Use a page definition that names the overlay for printing. Then specify the name of the page definition in your JCL. You can create the page definition by use of an AFP utility such as IBM Page Printer Formatting Aid (PPFA). For more information, refer to *Page Printer Formatting Aid: User's Guide and Reference*.
  - Code an IPO structured field in the print job.

EXAMPLE

This example, which is applicable for line data only, shows you how to print an overlay named O1PAGE1 on page 1 of the output and an overlay named O1PAGE3 on page 3 of the output. This example includes the print records as part of the print job and uses the MVS utility IEBGENER to send them to the printer.

```
//PAGOVLY JOB ...
//STEP1     EXEC  PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=*
//OUT1   OUTPUT PAGEDEF=P1USEROV
//SYSIN     DD  DUMMY
//SYSUT2    DD  SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=80),OUTPUT=*.OUT1
//SYSUT1    DD  *
    1This is print record 1 of page 1.
    !........O1PAGE1........
     This is print record 2 of page 1.
     This is print record 3 of page 1.
    1This is print record 1 of page 2.
     This is print record 2 of page 2.
     This is print record 3 of page 2.
    1This is print record 1 of page 3.
    !........O1PAGE3........
     This is print record 2 of page 3.
     This is print record 3 of page 3.
    1This is print record 1 of page 4.
     This is print record 2 of page 4.
     This is print record 3 of page 4.
```

The IPO structured field contains unprintable hexadecimal coding, represented by periods in the example. For information about coding the IPO structured field, refer to *Mixed Object Document Content Architecture Reference*; see also “Including AFP Structured Fields in Line Data” on page 127 in this publication.

MO:DCA data also requires an IPO structured field to invoke a page overlay. The coding of the IPO structured field is identical with the coding for line data. The name of the page overlay to be used on a page has to be specified in the Map Page Overlay (MPO) structured field of the Active Environment Group for that page. If the MO:DCA application is created by a text-formatting product, that product may automatically create both the MPO and IPO records when the page overlay is requested. If you are writing the Active Environment Group records, refer to *Mixed Object Document Content Architecture Reference* for the content and structure of the MPO structured field.

Positioning a Page Overlay

If you use an IPO structured field to include an overlay, you specify in the structured field the position at which the overlay is to print. If you use the page definition OVERLAY subcommand to include an overlay, you specify the position as a parameter on that command. This position includes the offset coded within the overlay resource. For ease of positioning, page overlays should be created with an internal overlay offset of 0,0.

Take care when positioning page overlays in a page that has a rotated print direction, such as DOWN (90°), BACK (180°), or UP (270°). PSF positions the overlay relative to the logical page origin, which does not change when the print direction (that is, the text orientation) is changed. In addition, PSF positions the physical, top-left corner of the overlay. This means that in a page with a DOWN print direction, the IPO position is measured from what appears to be the bottom left corner of the page to the bottom left corner of the overlay, as shown in Figure 37 on page 98.

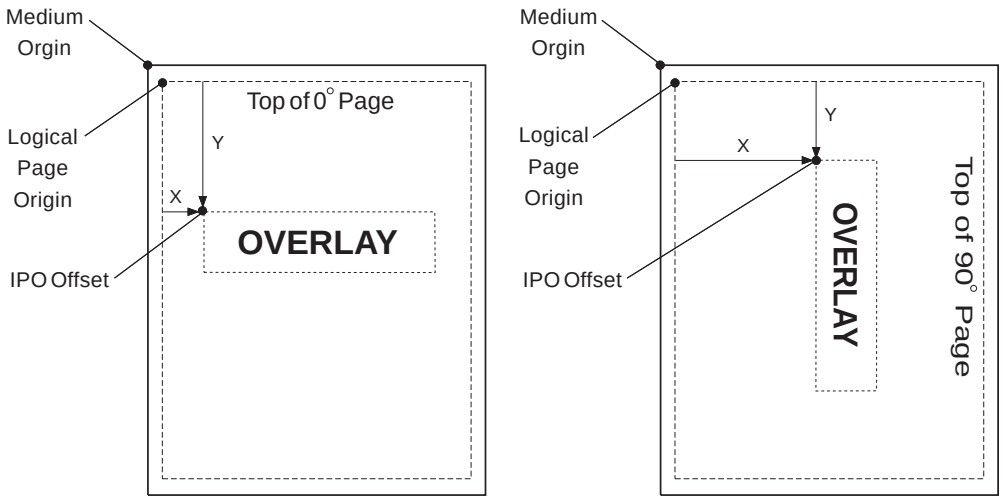


Figure 37. Positioning a Page Overlay

For more information about coding page overlays, refer to *Overlay Generation Language/370: User's Guide and Reference*.

Printing Line Data

Line data is printed according to instructions in the page definition used for printing the data set. A page definition coded in the PAGEDEF or the FCB parameter must be contained in 1 of the libraries available to the target printer. For additional information, see Chapter 5, “Printing Line Data.”

To print line data, you can use the default page definition defined for your printer, or you can specify your own page definition.

Specifying a Page Definition

If your job requires special page definition options, specify a page definition containing those options in the JCL for the print job. To do this, code the page definition name, without its 2-character prefix of “P1,” in the PAGEDEF parameter of the OUTPUT statement. For compatibility with non-AFP printers, you can also code the page definition name in the FCB parameter of the SYSOUT or the OUTPUT statement. The name coded in the FCB option cannot be more than 4 characters long.

The page definition must be stored in a private user library, a PSF user library (if supported for your system), or inline as part of the print data set. You can create your own page definition or use a page definition already available on your system.

PSF provides standard page definitions, which are listed in Appendix B, “Page Definitions Supplied with PSF.” Page definitions are provided to fit standard paper sizes for AFP printers. These page definitions provide different combinations of line spacing and print direction, as well as some multiple-up definitions. The page definitions are referred to in the following examples.

EXAMPLES

- The first example specifies a user-created page definition named P1MYPDEF, as well as a form definition named F1MYFDEF. Do not code the prefixes P1 and F1 in your JCL.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//OUT2  OUTPUT PAGEDEF=MYPDEF,FORMDEF=MYFDEF
//DD2 DD SYSOUT=A,OUT=(*.OUT2)
```

  To create a page definition, use an AFP utility product such as IBM Page Printer Formatting Aid (PPFA). For information about how to use PPFA to create page definitions, refer to *Page Printer Formatting Aid: User's Guide and Reference*. For information about form definitions, see “Specifying a Form Definition” on page 91.

- In the next example, a page definition named P1STD1, supplied with PSF for compatibility with non-AFP printers, is selected by use of the FCB option. Because no form definition is selected for this job, the PSF default form definition for the printer is used.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//DD2 DD SYSOUT=A,FCB=STD1
```

  Page definition P1STD1 is described in Appendix B, “Page Definitions Supplied with PSF.”

- In the next example, page definition P1MYPDEF is coded inline in the print data set generated by the program USERC. For information about coding inline resources, refer to *Mixed Object Document Content Architecture Reference*.

```
//AFPUSERC JOB ...
//STEP1 EXEC PGM=USERC
//OUT1  OUTPUT PAGEDEF=DUMMY
//DD1   SYSOUT=A,OUT=(*.OUT1)
```

  The PAGEDEF parameter must specify either the resource name of the inline page definition or the keyword DUMMY. If the name in the PAGEDEF parameter does not match the name of an inline page definition, PSF uses the resource from the resource library that matches the name in the JCL. If the PAGEDEF parameter is not specified in the JCL, PSF selects the first inline page definition in the print data set.

- To use a page definition from a PSF user library, see “Printing with Resources from a User Library” on page 112.

Specifying Print Direction

For line data, the page definition can specify a print direction of ACROSS, DOWN, BACK, or UP. For a description of AFP print directions, see “Print Direction” in Chapter 2, “Using JCL for Advanced Function Presentation.” In the following examples, PSF-supplied page definitions are used to control the print direction.

EXAMPLES

- This example specifies printing in the DOWN direction on a 3800 printer loaded with forms measuring 12 inches wide by 8.5 inches high. The resulting output

prints in the portrait format; that is, the page is turned so that its top is the short (8.5-inch) edge of the form.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT PAGEDEF=06061
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

Page definition P106061 is described in Appendix B, “Page Definitions Supplied with PSF.”

- This example specifies printing in the ACROSS direction on a cut-sheet printer. The resulting output is in portrait format because the top of the page will be the shorter, 8.5-inch side of the form. For IBM cut-sheet printers such as the 3825 printer, the top of an ACROSS page is always the short side of the form.

  ```
  //AFPUSERB JOB ...
  //STEP1 EXEC PGM=USERB
  //OUT1  OUTPUT PAGEDEF=A06462
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
     /*
  ```

  Page definition P1A06462 is described in Appendix B, “Page Definitions Supplied with PSF.”

- This example specifies printing in the ACROSS direction on a 3835 printer. Because the 3835 printer has the “Page Presentation Compatibility” feature, the output will be in portrait format unless the form definition used affects page compatibility. For more information, see “Maintaining Page-Presentation Compatibility” on page 74.

  ```
  //AFPUSERC JOB ...
  //STEP1 EXEC PGM=USERC
  //OUT1  OUTPUT PAGEDEF=A06462
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
     /*
  ```

  Page definition P1A06462 is described in Appendix B, “Page Definitions Supplied with PSF.”

Specifying Lines-per-Inch Spacing

For line data print jobs, the page definition controls the spacing of the print lines on the page. The following examples use page definitions supplied with PSF to specify different lines-per-inch (lpi) spacing.

EXAMPLES

- This example specifies printing at 8 lpi on a 3800 printer. Page definition P106080 prints 60 lines in the ACROSS direction on forms 12 inches wide and 8.5 inches high. The job also specifies the GT12 font, recommended for this page definition.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT PAGEDEF=06080
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),CHARS=GT12
     /*
  ```

  Page definition P106080 is described in Appendix B, “Page Definitions Supplied with PSF.”

- The next example prints at 8.5 lpi on a non-3800 printer. Page definition P1V06683 prints 66 lines in the DOWN print direction on letter-sized paper. The job also specifies the GT15 font, recommended for this page definition as suitable for printing at 8.5-lpi spacing.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//OUT1  OUTPUT PAGEDEF=V06683
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),CHARS=GT15
   /*
```

  Page definition P1V06683 is described in Appendix B, “Page Definitions Supplied with PSF.”

Multiple-Up Printing

For line data print jobs, the page definition can specify the arrangement of print lines to enable multiple pages of application data to fit on a single printed page. This is called *multiple-up* printing, which is **not** the same as N_UP printing. For additional information, see “Basic N_UP Printing: Printing Multiple Pages on a Sheet” on page 117.

Multiple-up printing reduces the number of pages required to print a file and can also increase the throughput (that is, the number of pages of application data per minute) of your printer, freeing your printer for other work. In addition to saving paper, multiple-up printing saves space required to store blank forms and printed output. To compound these benefits, see “Specifying Duplex Printing” on page 92.

The following examples use page definitions supplied with PSF to specify multiple-up printing. When printing multiple-up, be careful to select a font small enough to allow the data to fit on the page.

EXAMPLES

- This example specifies printing 2 side-by-side application pages of 66 lines each on a 3800 printer. Page definition P1M13280 is designed for forms 12 inches wide and 8.5 inches high. No font is specified in the JCL for the print job, because the page definition itself specifies that font GT24 is to be used for printing.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT PAGEDEF=M13280
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

  Page definition P1M13280 is described in Appendix B, “Page Definitions Supplied with PSF.”

- This example specifies printing 2 over/under application pages of 60 lines each on a non-3800 printer, using page definition P1W120C2. That page definition is designed for letter-size cut-sheet paper or continuous-forms paper 12 inches wide by 8.5 inches long.

  Program listings, dumps, and similar application output that are usually printed at 229 impressions per minute (ipm) on a 3900 printer (1 up) will still be printed at or near the same ipm, but will be printed at 2 pages, rather than 1 page, per impression. By printing multiple-up (with 2 pages on each side of a sheet), you can effectively print as many as 458 pages of application data per minute.

```
//AFPUSER1 JOB ...
//STEP1 EXEC PGM=USER1
//OUT1 OUTPUT PAGEDEF=W120C2,FORMDEF=C10110
//PRINT DD  SYSOUT=A,OUTPUT=(*.OUT1), CHARS=GT20
```

Suppressing Print Data

For line data, you can tell PSF not to print certain fields in the print records you send to the printer. You do this by using either selective field formatting or print suppression.

You can format selective fields by using a page definition to format only the fields in the print record that you wish to print. Fields that you omit from the format descriptions in the page definition will not be sent to the printer by PSF.

To suppress print data, use a page definition to format the fields *and* label the fields as eligible for suppression. Then use a form definition to specify which of the labeled fields are not to be printed. The suppressed fields are sent to the printer. Use this method of suppression when you are printing multiple copies of a page and you want the fields printed on some copies but suppressed on other copies.

To create page definitions and form definitions that format fields and suppress data, use an AFP program such as IBM Page Printer Formatting Aid (PPFA). For more information, refer to *Page Printer Formatting Aid: User's Guide and Reference*.

EXAMPLES

- This example uses a user-created page definition to format only the first 80 bytes of a 120-byte record.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT PAGEDEF=FORMAT
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
     /*
  ```

- In this example, a user-created page definition formats all the fields to be printed and labels a *Salary* field for suppression. The user-created form definition prints 2 copies of each page, suppressing the *Salary* field on the first copy.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT PAGEDEF=SUPSAL,FORMDEF=SUPSAL
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
     /*
  ```

Specifying Fonts

You can specify fonts for printing line data in your JCL or in the page definition. If you do not specify fonts in either place, PSF uses the default font for the printer.

Select a font that is of the correct size for the amount of data and the lines per inch spacing of your print job. Recommended fonts are listed with the page definition descriptions in Appendix B, “Page Definitions Supplied with PSF.” This appendix also contains a cross reference to *Page Definition Line-Spacing Values and Fonts* on page 150.

For additional information about fonts, see “Fonts” on page 21; for information about referencing page definitions, see “Specifying a Page Definition” on page 98.

EXAMPLES

- In this example, the JCL CHARS parameter is used to select font GT15. Because no font is specified in page definition P1V06683, PSF uses the font specified in the CHARS parameter.

  ```
  //AFPUSERB JOB ...
  //STEP1 EXEC PGM=USERB
  //OUT1  OUTPUT PAGEDEF=V06683
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),CHARS=GT15
     /*
  ```

  Page definition P1V06683 is described in Appendix B, “Page Definitions Supplied with PSF.”

- This example uses a page definition that contains a font specification. You need not code a CHARS parameter in your JCL. If you do code the CHARS parameter in your JCL, PSF will ignore it and will use the font specified in the page definition.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT PAGEDEF=M13280
  //PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
     /*
  ```

  Page definition P1M13280 is described in Appendix B, “Page Definitions Supplied with PSF.”

- The next example uses the default page definition for the printer and specifies font GT10 in the JCL.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //PRINT DD SYSOUT=A,CHARS=GT10
     /*
  ```

  Whether the GT10 font will be used depends on how the default page definition is defined for your printer. Your system programmer may have provided this information on the form shown in Appendix C, “Page-Printer Defaults Form.”

  For the examples in this chapter, assume that the printer is defined so that specifications of fonts in the CHARS parameter in the JCL override specifications of fonts in the default page definition.

- In the next example, multiple fonts are listed in the JCL CHARS parameter. This enables you to select fonts by use of table reference characters (TRC) in the print data records. The presence of TRCs in the print data is indicated in the TRC parameter of the OUTPUT statement. If TRCs are not specified in the print data, all the data will print in the first font in the list, GT10.

  ```
  //AFPUSERA JOB ...
  //STEP1 EXEC PGM=USERA
  //OUT1  OUTPUT TRC=YES
  //PRINT DD SYSOUT=A,CHARS=(GT10,GB10)
     /*
  ```

Fonts for TRC selection can also be specified in the page definition. For additional information, see “Using Table Reference Characters to Select Fonts” on page 121.

These examples apply only to fonts used in line data applications. In MO:DCA documents or overlays, fonts are defined in structured fields within the document or overlay. Refer to the reference publication for the AFP utility you used to create the overlay or document, or refer to *Mixed Object Document Content Architecture Reference*.

Changing Formatting within a Document

In the preceding examples, the same formatting specifications are used for all the pages of the line data set. But suppose you want to change some of the formatting in the middle of your data set. For example, suppose you want some pages printed at 6 lines per inch and other, summary pages printed at 8 lines per inch.

To change page definition options such as print direction, line spacing, or record formatting in a print job, you have to use a page definition that contains multiple page formats, and you will have to tell PSF when to use which page format. For an example that shows how to do this, see “Using Multiple Copy Groups or Page Formats.”

Using Multiple Copy Groups or Page Formats

If your print job requires you to change form definition options, such as medium overlays or paper source, for different pages in the data set, do the following:

1. Create a form definition that contains multiple copy groups, with the options you want for different pages coded in different copy groups.
2. Identify that form definition in the JCL of your print job by the method described in “Specifying a Form Definition” on page 91.
3. In your print data, include an Invoke Medium Map (IMM) structured field in front of any page on which you want to change the copy groups.
4. Alternatively, you may be able to use conditional processing in a page definition to trigger the use of a new copy group determined by the content of data fields in the application. This eliminates the need to code IMM structured fields in the data.

Similarly, if your job requires different page definition options, such as lines per inch spacing or print direction, for different pages in the data set, do the following:

1. Create a page definition that contains multiple page formats, with the options you want for different pages coded in different page formats.
2. Identify that page definition in the JCL of your print job, as described in “Specifying a Page Definition” on page 98.
3. In your print data, include an Invoke Data Map (IDM) structured field in front of any page on which you want to change the page formats.
4. Alternatively, you may be able to use conditional processing in a page definition to use a new page formats base on data fields in the application. This eliminates the need to code IDM structured fields in the data.

You can include the IMM and IDM structured fields yourself, or you can invoke them by using the Conditional Processing function of the page definition. For additional information, see “Including AFP Structured Fields in Line Data” on page 127 and “Conditional Processing” on page 85.

EXAMPLES

In the following example, pages 1, 2, and 4 will be printed at a line spacing of 6 lines per inch (lpi) on paper from the main paper source. Page 3 will be printed at a spacing of 8 lpi on blue paper from the alternate paper source.

Form definition F1MUCG has been created containing 2 copy groups:

Page format F2MUMAIN prints on paper from the main paper source.
Page format F2MUALTB prints on paper from the alternate paper source.

Page definition P1MUFMT has been created containing 2 page formats:

Page format P2MU6LPI prints pages at 6 lpi.
Page format P2MU8LPI prints pages at 8 lpi.

This example shows the IMM and IDM structured fields coded in the print data set. For this example, the print records have been included as part of the print job, and the MVS IEBGENER utility is used to send them to the printer. Because the data set contains structured fields, the record format has to be defined with carriage control (RECFM=FBA).

```
//AFPIDM JOB ...
//STEP1     EXEC  PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=*
//OUT1   OUTPUT FORMDEF=MUCG,PAGEDEF=MUFMT
//SYSIN     DD  DUMMY
//SYSUT2    DD  SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=80),OUTPUT=*.OUT1
//SYSUT1    DD  *
   !........F2MUMAIN
   !........P2MU6LPI
   1This is print record 1 of page 1.
    This is print record 2 of page 1.
    This is print record 3 of page 1.
   1This is print record 1 of page 2.
    This is print record 2 of page 2.
    This is print record 3 of page 2.
!........F2MUALTB
   !........P2MU8LPI
   1This is print record 1 of page 3.
    This is print record 2 of page 3.
    This is print record 3 of page 3.
   !........F2MUMAIN
   !........P2MU6LPI
   1This is print record 1 of page 4.
    This is print record 2 of page 4.
    This is print record 3 of page 4.
```

The IMM and IDM structured fields contain unprintable hexadecimal fields, which are represented as periods in the coding shown above. The format of these records is described in *Mixed Object Document Content Architecture Reference*.

If the IMM and IDM structured fields are omitted before page 1 of the data set, PSF begins printing with the first copy group in the form definition and with the first page format in the page definition.

If conditional processing were used for this application, IMM and IDM structured fields would not be required in the print data. For more information about using conditional processing, see “Conditional Processing” on page 85.

Printing Page Segments

You can have page segments printed as part of an overlay resource, or you can have them printed as part of the data by using an Include Page Segment (IPS) structured field. The IPS structured field that names and positions the page segment is included as a record in the print data set.

When using IPS structured fields in line data sets, you can also map the page segment in the page definition to retain it in the printer while your data set is printing. This can speed performance if the page segment is used multiple times in the same print job.

EXAMPLES

- In this example, page segment S1LOGO is specified in an overlay. The overlay is referenced in the form definition F1USERA.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT FORMDEF=USERA
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

  For information about form definitions and overlays, see “Specifying a Form Definition” on page 91 and “Printing a Medium Overlay” on page 96. For information about specifying page segments in an overlay, refer to the publications for the AFP utility product used to create your overlay.

- In the next example, page segments are invoked by use of IPS records in a line data set. Page segment S1LOGO is printed on all 4 pages of the document. Page segment S1MAP is printed only on pages 2 and 4. The user listed S1LOGO in the page definition but did not list S1MAP in the page definition.

  The following is an example of coding a hard page segment by naming it in the page definition Segment List.

```
setunits 10 cpi 6 lpi linesp 6 lpi ;
pagedef nnxx0 width   8.3 in height 10.8 in replace no ;
font fnorm cr10 ;
font fbold cb10 ;

pageformat p2nnxxx0 direction across ;
trcref 0 font fnorm ;
trcref 1 font fbold ;
segment S1LOGO ;

  printline channel 1 repeat 1 position 5 5 font fnorm;
/* name */ field start 6 length 20 position current current ;
/* acct */ field start 1 length 5 position 20 current font fbold;
/* acct */ field start 1 length 5 position 70 -4 direction down ;

printline       repeat    3 position 0 next font fnorm ;

printline channel 2 repeat 55 position 0 next ;
```

  For this example, the print records are included as part of the print job, and the MVS IEBGENER utility is used to send them to the printer. Because structured field records are included in the data set, the record format must be defined with carriage control (RECFM=FBA).

```
//AFPIPS JOB ...
   /*ROUTE PRINT DL3820B
//STEP1     EXEC  PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=*
//OUT1   OUTPUT PAGEDEF=USERB
//SYSIN     DD  DUMMY
//SYSUT2    DD  SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=80),OUTPUT=*.OUT1
//SYSUT1    DD  *
   1This is print record 1 of page 1.
   !........S1LOGO.........
    This is print record 2 of page 1.
    This is print record 3 of page 1.
   1This is print record 1 of page 2.
   !........S1LOGO.........
   !........S1MAP..........
    This is print record 2 of page 2.
    This is print record 3 of page 2.
   1This is print record 1 of page 3.
   !........S1LOGO.........
    This is print record 2 of page 3.
    This is print record 3 of page 3.
   1This is print record 1 of page 4.
   !........S1LOGO.........
   !........S1MAP..........
    This is print record 2 of page 4.
    This is print record 3 of page 4.
```

Because S1LOGO is listed in the page definition, it will be loaded in the printer at the beginning of page 1 and used for all 4 pages. S1MAP, which is not listed in the page definition, will be loaded in the printer at page 2, and loaded again for page 4.

The IPS structured field contains unprintable hexadecimal coding, which is represented by periods in the example. For information about coding the IPS structured field, refer to *Mixed Object Document Content Architecture Reference* and see “Including AFP Structured Fields in Line Data” on page 127.

Printing MO:DCA Data

MO:DCA data is printed according to instructions in the MO:DCA-P structured fields in the data set. To change the formatting of a page document, you change the structured fields. For a description of the structured fields used in page applications, refer to *Mixed Object Document Content Architecture Reference* or the publication for the product used in creating your MO:DCA data.

The following example shows the code for submitting the output of a page application to an AFP printer.

```
//AFPPAGE JOB ...
//STEP1 EXEC PGM=PAGEA
//OUT1  OUTPUT FORMDEF=A10111
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),
//   DCB=(RECFM=VBA,LRECL=1993,BLKSIZE=32760)
   /*
```

No page definition or fonts are specified in the JCL, because the MO:DCA data contains its own internal formatting commands and font definitions. A form definition, however, is required. If you do not specify a form definition in the JCL, PSF uses the default form definition for the printer. For more information, see “Specifying a Form Definition” on page 91.

MO:DCA data must be defined with carriage controls, which are specified in the RECFM parameter. MO:DCA data is usually written as variable-length blocked records, with RECFM coded as either VBA or VBM. Record size and block size may vary.

Using Table Reference Characters (TRCs) to Select Fonts

Following are examples of JCL in which an OUTPUT statement and a DD statement are used to tell PSF that TRCs are specified in the line-data records:

EXAMPLES

- This example specifies that the line data contains TRCs by using the TRC parameter in the OUTPUT statement:

```
//OUT2 OUTPUT CHARS=(GT10,GT12),TRC=YES
//DD2 DD SYSOUT=A,OUTPUT=*.OUT2
```

- This example specifies that the data set contains TRCs by using the DCB subparameter in the DD statement:

```
//OUT2 OUTPUT CHARS=(GT10, GT12)
//DD2 DD SYSOUT=A,OUTPUT=*.OUT2,DCB=OPTCD=J
```

For more details about using TRCs to select fonts, see “Using Table Reference Characters to Select Fonts” on page 121.

Specifying Shift-Out, Shift-In (SOSI) Codes

The following example tells PSF to use a particular data-scanning mode for a data set to be printed on a 3820 printer. Both single-byte fonts and double-byte fonts are to be used for printing. The first font specified is GT10, a single-byte font; the second is G24F, a double-byte font. The single-byte font file should be named X0GT10, and the double-byte font file should be named X0G24F. For more information about specifying SOSI codes, see “Using Shift-Out, Shift-In (SOSI) Codes” on page 126.

EXAMPLE

This example specifies that the SOSI1 process mode is to be set up for a data set that will be printed on a specific 3820 printer:

```
//OUT1 OUTPUT CHARS=(GT10,G24F),PRMODE=SOSI1
//DD1 DD SYSOUT=B,DESTINATION=(REMOTE1),OUTPUT=(*.OUT1)
```

Printing More than 1 Copy

To print more than 1 copy of your AFP data, you can use either of 2 methods:

- You can transmit the data set to the printer any number of times, producing a copy of the entire data set for each transmission. You can specify collated copies either by using the JCL COPIES parameter or by using multiple OUTPUT statements in the JCL.
- You can print multiple copies of each page in turn. You can specify page copies in the form definition, or you can code them as subgroups on the JCL COPIES parameter.

You can include copies of both kinds in a single data set. For more information about these options and the relationships between them, see the descriptions of the COPIES parameter in Chapter 2 and in "Specifying FORMDEF and COPIES or FLASH Parameters in JCL" on page 78.

EXAMPLES

- This example transmits the data set to the printer 3 times, printing 3 complete, collated copies of the data set.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//PRINT DD SYSOUT=A,COPIES=3
   /*
```

- The next example uses a user-created form definition, named F1UCOPY2, that specifies 2 copies of each page. These 2 copies can be defined in the form definition with different overlays, suppressions, or flash options.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//OUT1  OUTPUT FORMDEF=UCOPY2
//DD1 DD SYSOUT=A,OUTPUT=(*.OUT1)
   /*
```

  The data set will be transmitted 1 time and will contain 2 copies of each page.

- The next example uses both the COPIES parameter and the user form definition named F1UCOPY2.

```
//AFPUSERC JOB ...
//STEP1 EXEC PGM=USERC
//OUT1  OUTPUT FORMDEF=UCOPY2
//DD1 DD SYSOUT=A,OUTPUT=(*.OUT1),COPIES=3
   /*
```

  The data set will be transmitted 3 times, as specified in the COPIES parameter. Each of those transmissions will contain 2 copies of each page, as specified in the form definition.

- To print multiple copies of a data set, each with different AFP formatting, use multiple OUTPUT statements in the JCL.

```
//AFPUSERD JOB ...
//STEP1 EXEC PGM=USERD
//OUT1  OUTPUT PAGEDEF=USER1,FORMDEF=USER1
//OUT2  OUTPUT PAGEDEF=USER2
//DD1 DD SYSOUT=A,OUTPUT=(*.OUT1,*.OUT2)
   /*
```

The job will first print the data set according to the page definition and form definition specified in the OUT1 OUTPUT statement. Then the job will print the data set again, this time using the page definition specified in the OUT2 OUTPUT statement. Because this statement does not specify a form definition, the default form definition is used.

Bursting and Stacking Continuous-Forms Paper for the 3800 and 3900 Printers

Bursting paper means separating the continuous-forms paper into individual sheets. The 3800 or 3900 printer must be equipped with an optional burster-trimmer-stacker (BTS) device. If the printer does not have a BTS, the forms go to a continuous-forms stacker.

Use the BURST parameter to specify whether you want the output paper to go to the BTS or to the continuous-forms stacker. If your data set is printed on a cut-sheet-paper printer, PSF ignores the BURST parameter.

EXAMPLES

- This example is for printing on a 3900 printer. The BURST parameter in the DD statement specifies that you want the output to be burst into separate sheets.

  ```
  //DD1 DD SYSOUT=A,DEST=PR3900,BURST=YES
  ```

- In this example, the BURST parameter is specified in the OUTPUT statement.

  ```
  //OUTPUT1 OUTPUT BURST=YES
  //DD1 DD SYSOUT=A,OUTPUT=(*.OUTPUT1)
  ```

Specifying Whether You Want Error Messages to Be Printed

The PIMSG parameter specifies whether you want error messages to be printed and also specifies the maximum number of errors that can occur before printing is stopped. The PIMSG count applies only to “nonterminating” errors, which would not, in themselves, cause the data set to stop printing. Data set printing will always be terminated if a terminating error occurs, regardless of the setting of PIMSG.

Values for the PIMSG parameter are assigned in the PSF startup procedure. To override these values, code the PIMSG parameter in the JCL for your print job.

EXAMPLES

- The first example specifies that all error messages are to be printed and that printing is to be stopped after 10 errors occur.

  ```
  //OUT1 OUTPUT PIMSG=(YES,10)
  //DD1 DD SYSOUT=A,OUTPUT=*.OUT1
  ```

- For the next example, the system will not print messages unless an error occurs that stops printing, and then will print only the message group caused by the terminating error. The PIMSG count, which is not coded, defaults to 16. If 16 nonterminating errors occur, printing will be stopped, because the PIMSG count was exceeded. The only message group printed will indicate that PIMSG count was exceeded.

```
//OUT2 OUTPUT PIMSG=NO
//DD2 DD SYSOUT=A,OUTPUT=*.OUT2
```

- The last example sets the PIMSG count to 0 and specifies that no messages are to be printed unless a terminating error occurs, so that the data set will continue to print regardless of how many nonterminating errors occur. If a terminating error occurs, PSF prints the message group caused by the terminating error.

```
//OUT3 OUTPUT PIMSG=(NO,0)
//DD3 DD SYSOUT=A,OUTPUT=*.OUT3
```

PSF issues messages to 1 or more of the following destinations, depending on the most appropriate:

Operator's console
Messages that the operator or system programmer must address.

Security administrator's console
Messages reporting security violations or errors in the processing of security functions.

Job submitter's console
Messages reporting the completion of a job or its failure to print.

In the printed output
Messages reporting the position from which data is retransmitted during a recovery.

Following the printed output or elsewhere
Messages describing data stream errors. Also, a duplicate copy of any message issued within the printed output.

Printing with Resources from a User Library

If you are using an MVS/ESA system, you can access resources from user libraries that are not defined to other PSF jobs for that printer. You can specify a maximum of 8 user libraries in the USERLIB parameter of the OUTPUT statement.

Because PSF does not retain copies of resources from user libraries between print jobs, user libraries are useful for testing new resources and for protecting secure resources.

EXAMPLE

- This example specifies 2 user libraries that will be searched for resources before the PSF system libraries are searched. The following resources are specified for the print job:
  - Form definition F1USERA is in user library USERA.RESOURCE.
  - Overlay O1USERA, referenced in F1USERA, is also in user library USERA.RESOURCE.
  - Page definition A06462 is in a PSF system library.
  - Font GT15 is in USERA.FONTS. A font named GT15 is also in the PSF system font library.
  - Fonts referenced in the overlay are in the system library.

```
//AFPUSERA JOB ...
//STEP1 EXEC PGM=USERA
//OUT1  OUTPUT FORMDEF=USERA,PAGEDEF=A06462,
//    USERLIB=('USERA.RESOURCE','USERA.FONTS')
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),CHARS=GT15
   /*
```

The above example will find and use the form definition, overlay, and font GT15 in the user libraries. The page definition and other fonts will be used from the system library.

Note: You do not define the system libraries in the JCL for your print job. Your system programmer has already defined system libraries to PSF for all jobs that use the printer.

Printing with Inline Resources

Before 2.2.0, PSF used form definitions and page definitions sent inline with the data set, in addition to using such resources from PSF user libraries and system libraries. PSF 2.2.0 and later can use all types of resources sent inline: fonts (code pages, character sets, and coded fonts), overlays, page segments, and microfilm setup resources, in addition to form definitions and page definitions.

PSF stores inline resources temporarily and uses them only for the data set in which they are sent inline. After the data set finishes processing, PSF deletes the resources to prevent their being used by any other data set.

PSF searches the temporary, inline-resource library for a resource before searching for it in the system libraries or the user libraries. (When you use a resource from a security library, however, PSF searches only the security library. For information about using security resources, refer to *Print Services Facility: Security Guide*.)

PSF can use inline resources (fonts, page segments, and overlays) that have been marked for use with resident fonts or for use with the DPF resource library. Your system programmer must mark these resources, using the APSRMARK program, before you can use them with resident fonts or with the DPF resource library. For more information about using APSRMARK, see “Using APSRMARK to Mark Resources” on page 34 and “Using Resources with the Distributed Print Function (DPF) of PSF” on page 34. If you need to understand the system programmer's role in using resources stored in the DPF library, refer to *Print Services Facility/MVS: System Programming Guide*.

If you send several resources with the same name inline, PSF uses the last resource received for processing the data set.

The maximum length of any record in an inline resource is 32 752. The maximum size of each inline resource that PSF can handle is about 1GB (a gigabyte is 1 073 741 824 bytes).

For more information about using inline form definitions, see page 53. For more information about using inline page definitions, see page 56.

EXAMPLE

- This example specifies 2 inline resources for PSF to use. PSF searches the system libraries for the other resources. The following resources are specified for the print job:
  - Form definition F1INFDEF is sent inline with the data set.
  - Overlay O1INLOV, referenced in F1INFDEF, is also sent inline.
  - Page definition P1A06462 is in a PSF system library.
  - Font GT10 is in the PSF system font library.

```
//AFPUSERB JOB ...
//STEP1 EXEC PGM=USERB
//OUT1  OUTPUT FORMDEF=INFDEF,PAGEDEF=A06462
//PRINT DD SYSOUT=A,OUTPUT=(*.OUT1),CHARS=GT10
   /*
```

  The above example causes PSF to use the inline form definition and overlay and to take the page definition and the font from the system library. You need not define the system libraries in the JCL for your print job; your system programmer has already defined system libraries to PSF for all the jobs that use the printer.

Specifying Notification When the Print Job Has Finished Printing

If you want PSF to send a notification message when the printer has finished your job, specify the NOTIFY option in the OUTPUT statement.

PSF issues a print-complete message with these variables:

- JOBNAME
- STEP
- ID
- SYSTEM

If the NOTIFY message is sent to an MVS node, the message is saved until the user logs on.

EXAMPLES

- The following example specifies that a print-complete message is to be sent to 4 users at the same node.

```
//GO.OUT1 OUTPUT NOTIFY=(DEST01.USERID1,DEST01.USERID2,
//     DEST01.USERID3,DEST01.USERID4)
//GO.SYSOUT1 DD SYSOUT=N,DCB=(RECFM=VBM,BLKSIZE=8192),
//     OUTPUT=*.OUTPUT1
```

- This example specifies that a print-complete message is to be sent to 16 users at the originating node (the system that runs the print job).

```
//GO.OUTPUT1 OUTPUT NOTIFY=(USERID1,USERID2,USERID3,USERID4)
//GO.OUTPUT2 OUTPUT NOTIFY=(USERID5,USERID6,USERID7,USERID8)
//GO.OUTPUT3 OUTPUT NOTIFY=(USERID9,USERID10,USERID11,USERID12)
//GO.OUTPUT4 OUTPUT NOTIFY=(USERID13,USERID14,USERID15,USERID16)
//GO.SYSUT1 DD    SYSOUT=N,DCB=(RECFM=VBM,BLKSIZE=8192),
//           OUTPUT=(*.OUTPUT1,*.OUTPUT2,*.OUTPUT3,*.OUTPUT4)
```

To determine whether your installation supports the print-notify function, see your system programmer.

A BROADCAST data set must be available to receive messages that are sent to an MVS system. For more information, see your system programmer.

Specifying Duplex Offset

This example shows how to use the duplex offset function. The data is specified first, followed by a form definition created using Page Printer Formatting Aid (PPFA).

```
THIS IS PRINT LINE 1
THIS IS PRINT LINE 2
   .
   .
   .
THIS IS PRINT LINE 70
```

The following example shows the form definition, which specifies that both the back and the front of the duplexed page are to be offset, possibly to accommodate hole-punching or binding.

```
FORMDEF DO11
        REPLACE YES
        DUPLEX NORMAL ;
   COPYGROUP CPNAME
        OFFSET .94 in 1 in .81 in 1 in ;
```

Transmitting a Data Set to an AS/400 System

You can transmit a data set to an AS/400 system and print it on an IPDS printer such as the 3130, 3825, 3829, 3835, 3900, 3930, 3935, 6408, or 6412 printer.

In this example, assume that AS400SYS is the destination name of the AS/400 system. Replace USER.DATASET, *.MYOUT, and class J with your own values.

```
//MYJOB JOB (A,BC),'TEST',MSGCLASS=H
//INSTR PROC
//SPOOL   EXEC PGM=IEBGENER
//MYOUT OUTPUT DEST=AS400SYS.USERID,
//             COPIES=1
//SYSPRINT  DD SYSOUT=*
//SYSUT2    DD SYSOUT=J,OUTPUT=*.MYOUT
//SYSUT1    DD DSN=USER.DATASET,DISP=SHR
//SYSIN     DD DUMMY
//    PEND
//SAMPLE EXEC PROC=INSTR
```

The steps in setting up the AS/400 to receive and print files are as follows:

- Correctly define to the AS/400 system AFP printers.
- Configure the printers according to instructions in *AS/400 Printer Device Programming*.
- Make resources available by using the following commands:
  - CRTFORMDF (form definition)
  - CRTOVL (overlays)
  - CRTPAGSEG (page segments)
  - CRTPAGDEF (page definition)
  - CRTFNTRSC (fonts)
- Use the CRTDEVPRT or CHGDEVPRT command to specify the correct default form definition and font and to specify the printer as an AFP printer.
- Start the printer writer to print the job.

For information on AS/400 commands, refer to *AS/400 Command Language Reference*.

Basic N_UP Printing: Printing Multiple Pages on a Sheet

Beginning with PSF 2.2.0, you can print up to 4 pages on a sheet printed in simplex mode, or up to 8 pages on a sheet printed in duplex mode, by creating a new form definition containing the PPFA N_UP subcommand. To create an application such as that described in the example, you need a printer that supports N_UP printing, such as a 3130, 3825, 3827, 3828, 3829, 3835, 3900 Model 001, 3900 Duplex, 3900-0W1, or 3935 printer. You can use the PPFA program to create the required form definition.

For example, you might want to print pages 1 and 2 of a letter side by side on a single sheet of paper, and then print pages 3 and 4 on a second sheet, thus printing a four-page letter on 2 sheets of paper, as shown in Figure 38. You can also specify different overlays or page segments for each page. For example, page 1 of your letter might contain a company logo, and page 4 a signature. To avoid printing pages too small to read, you might want to print on a 3935 or a 3900-0W1 printer, either of which can print data up to 17 inches wide.

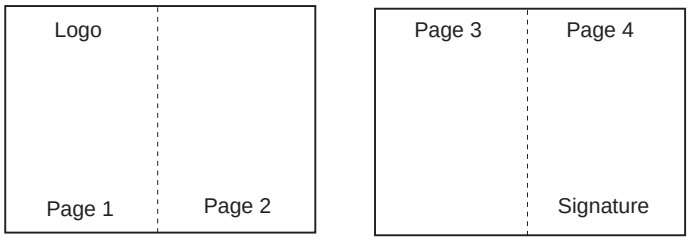


Figure 38. Printing 4 Pages on 2 Sheets

To create a job similar to that diagrammed in this figure, begin by using PPFA to create a form definition that specifies the following:

- The N_UP subcommand
- The ACROSS printing direction
- The PORTRAIT page presentation
- A font of the correct size to allow your text to fit in the page area

Page Printer Formatting Aid: User's Guide and Reference describes the types of N_UP printing you can perform, shows the command syntax, and provides numerous examples for basic N_UP printing as well as the commands and syntax for enhanced N_UP printing. For the N_UP 2 form definitions supplied with PSF, see Table 8 on page 137.

Your JCL might look something like the following. No font is specified in the JCL, because your page definition names a font. In this example, SYSOUT A prints on the 3900-0W1 printer.

EXAMPLE

```
//AFPUSER JOB...
//STEP1 EXEC PGM=USERA
//OUTPUT1 OUTPUT PAGEDEF=nuppdef,FORMDEF=nupfdef
//PRINT DD SYSOUT=A,OUTPUT=(*.OUTPUT1)
/*
```

Specifying JCL Parameters for Microfilm Jobs

When you send output to a microfilm device, you should identify the microfilm setup resource to distinguish the special microfilm options. If you do not, the system will use the default COMSETUP parameter that your system programmer specified in the PRINTDEV. The full name of the microfilm setup resource is coded in the COMSETUP parameter of the OUTPUT statement.

The microfilm setup resource must be in a PSF system library, in a PSF user library, or inline as part of the print data set. You can create your own microfilm setup resource (refer to the publications provided with your microfilm device) or use a resource already available on your system.

This is an example of JCL in which the COMSETUP parameter is specified in the OUTPUT statement for application data sent to a microfilm device. IBM recommends a prefix of H1 for the name of a microfilm setup resource.

```
//OUTPUT1 OUTPUT DEST=name,FORMDEF=fdefname,PAGEDEF=pdefname,
//         COMSETUP=membername
//DD1    DD  SYSOUT=q,OUTPUT=(*.OUTPUT1)
```

This is an example of JCL in which 2 OUTPUT statements are specified for application data sent to a microfilm device and to a printer.

```
//OUTPUT1 OUTPUT CLASS=comname,FORMDEF=fdefname,PAGEDEF=pdefname,
//         COMSETUP=membername
//OUTPUT2 OUTPUT CLASS=prtname,FORMDEF=fdefname,PAGEDEF=pdefname,
//DD1    DD  SYSOUT=q,OUTPUT=(*.OUTPUT1,*.OUTPUT2)
```

This is an example of JCL in which 2 different FORMDEF parameters are specified for application data sent to a microfilm device and to a printer. The example specifies 3 printed copies and 1 microfilm copy.

```
//OUTPUT1 OUTPUT CLASS=comname,FORMDEF=comfdef,PAGEDEF=pdefname,
//         COMSETUP=membername
//OUTPUT2 OUTPUT CLASS=prtname,FORMDEF=prtfdef,PAGEDEF=pdefname,
//         COPIES=3
//DD1    DD  SYSOUT=q,OUTPUT=(*.OUTPUT1,*.OUTPUT2)
```

Ask your system programmer which CLASS or DEST has been set up for sending AFP output to the microfilm device.

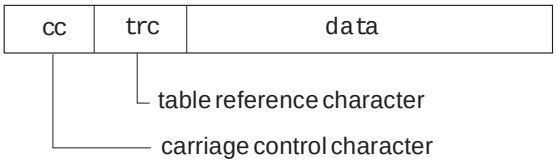
Chapter 5. Printing Line Data

This chapter is intended to help you create line-data records in the format expected by PSF. This chapter also contains information you can use to code MO:DCA-P structured fields. The structured fields are considered general-use programming interfaces and are documented in *Advanced Function Presentation: Programming Guide and Line Data Reference*; refer to that publication before coding structured fields in your program.

Line data is data formatted for printing on a line printer. A line-data record can contain a 1-byte carriage control character and a 1-byte table reference character followed by the data to be printed.[20] Both characters are optional and are defined as follows:

cc Carriage control character, which starts an operation to write, space, or skip

trc Table reference character, which selects the font in which the line is to be printed



and a Table Reference Character

Figure 39. Line-Data Record Containing a Carriage Control Character

Existing applications[21] that generate data consisting entirely of line-data records can be printed on a page printer. However, a page definition is required in place of the forms control buffer (FCB) used for line printers. For information about printing jobs generated for a 3800 line printer and for jobs containing merged lines, see “Merging Data Lines into a Single Print Line” on page 124.

As was noted in “Page Definitions Supplied with PSF” on page 87, PSF page definitions are provided for the following FCB modules: FCBSTD1, FCBSTD2, FCBSTD3, FCB36, and FCB38. If the name of the page definition is the same as that of the FCB used with a line printer, no JCL changes are required, because the FCB parameter is interpreted as the name of a page definition. If the name is different, however, you must specify the name of the page definition with either the PAGEDEF or the JCL FCB parameter.

TRCs are supported for compatibility with 3800 line printers. For line data applications in which the number of lines between font changes is constant, IBM recommends selecting fonts in the page definition. For font selection when the number of lines between font changes is variable, TRCs can provide more flexibility.

---

[20] With a line printer, the maximum number of data bytes in a single input record is 208. With a page printer, the maximum number is 32 768 bytes.

[21] “Existing applications” are applications originally designed for formatting and printing data on line printers. For considerations that apply when these applications are run on a page printer, refer to *Advanced Function Presentation: Printer Information*.

Specifying Carriage Control Characters and Table Reference Characters

Line data can contain carriage control characters (CC), table reference characters, (TRC), both, or neither. You can use either of 2 types of carriage control character: American National Standards Institute (ANSI), or machine code. To indicate that the records contain carriage control characters, and to indicate which type is being used, specify RECFM=A or RECFM=M in the DCB subparameter.

To indicate that TRCs are contained in the records, specify either of the JCL parameters, TRC=YES or DCB=OPTCD=J. For more information, see “Using Table Reference Characters (TRCs) to Select Fonts” on page 109.

Using Carriage Control Characters in Line-Data Records

Carriage control characters are used in line data to control writing, spacing, and skipping operations as the data is being formatted. The forms control buffer (FCB) is used to map carriage control characters to the physical actions that line printers perform to format print data vertically. For page printers, the page definition replaces the function of the FCB. If you are using a page definition with a page printer to replace an FCB, the page definition must specify the same number of lines per inch and have the same actions defined for carriage control characters as the FCB used with the line printer.

You can use the Page Printer Formatting Aid (PPFA) product to create page definitions that specify exactly the same skipping and spacing instructions as the FCBs used with existing line data applications.

Table 3 on page 121 lists the hexadecimal carriage control characters you can use. The decimal representations of the ANSI codes are in parentheses.

[22] These particular carriage control characters are always handled as ANSI carriage control characters.

Table 3. Carriage Control Characters

Action	Machine Code Control Characters Action after Printing	Machine Code Control Characters Action Only (Immediate)	ANSI Control Characters Action before Printing
Print (no space)	01	-	4E (+)
Space 1 line	09	0B	40 ( )
Space 2 lines	11	13	F0 (0)
Space 3 lines	19	1B	60 (-)
Skip to Channel 1	89	8B	F1 (1)
Skip to Channel 2	91	93	F2 (2)
Skip to Channel 3	99	9B	F3 (3)
Skip to Channel 4	A1	A3	F4 (4)
Skip to Channel 5	A9	AB	F5 (5)
Skip to Channel 6	B1	B3	F6 (6)
Skip to Channel 7	B9	BB	F7 (7)
Skip to Channel 8	C1 [22]	C3 [22]	F8 (8)
Skip to Channel 9	C9	CB	F9 (9)
Skip to Channel 10	D1	D3	C1 (A) [22]
Skip to Channel 11	D9	DB	C2 (B)
Skip to Channel 12	E1	E3	C3 (C) [22]

Note: PSF ignores the following hexadecimal machine carriage control characters: 02 through 07, 0A, 12, 23, 43, 63, 6B, 73, 7B, EB, F3, and FB. PSF prints data lines containing these characters in single-spacing mode.

If your application creates line-data records containing carriage control characters that skip to a channel, the page definition used to print the application output must contain instructions for processing those channel codes. If the skipping or spacing actions you have specified in a page definition move the print position past the last line of the current page, PSF starts a new page at the first print position indicated by the page format. PSF does not carry skipping or spacing over to the new page.

An alternative to using carriage control characters or channel codes in line-data records is to define data placement entirely within the page definition, using line count or field formatting.

Using Table Reference Characters to Select Fonts

You can use table reference characters (TRCs) in line-data records to select the font to be used in printing full lines or text in a line. To select fonts for full lines, code a TRC in each output data line. When line data contains both carriage control characters and table reference characters, the carriage control character precedes the table reference character, as in Figure 40 on page 122. The table reference character (0, 1, 2, or 3) selects the font corresponding to the order in which you have specified the font names with the JCL CHARS parameter.

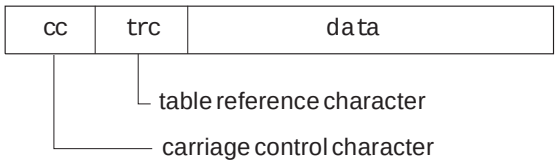


Figure 40. Line-Data Record Containing a Carriage Control Character and a Table Reference Character

You can also code table reference characters that correspond to font names specified in a font list defined for the page definition. Although you can specify 128 fonts in a page definition, the printer allows fewer fonts per page than that.

This example shows how to use table reference characters (TRCs) to specify fonts in line data. The example uses column 1 for the carriage control character and column 2 for the TRC (0, 1, 2, or 3).

EXAMPLE

```
12This line should print in a bold font.
 1This is a normal print line.
```

The next example creates printing resources by using the PPFA PAGEDEF TRCREF command. Values different from the default values are specified in the DIRECTION and ROTATION subcommands of the TRCREF command.

```
SETUNITS LINESP 8 LPI;

FORMDEF rhdr (form definition name)
  REPLACE yes
  OFFSET .5 in .5 in
   DUPLEX NORMAL;

PAGEDEF rhdr (page definition name)

   WIDTH 8.5 IN
   HEIGHT 10.0 IN
   LINEONE 1.0 IN 1.0 IN
   DIRECTION ACROSS
   REPLACE YES;

   FONT normal ST12 ROTATION 0;
   FONT heading GT10 ;

   PAGEFORMAT rhdr;

      TRCREF 1 FONT normal;
      TRCREF 2 FONT heading DIRECTION DOWN ROTATION 270 ;

      PRINTLINE CHANNEL 1
         POSITION 8.0 IN 1.0 IN
         DIRECTION DOWN;

      PRINTLINE
         POSITION MARGIN 2.0 IN
         DIRECTION ACROSS
         REPEAT 60;
============================================================
12Chapter 1 should be printed in font GT10.
 1This is a normal print line and should be printed in font ST12.
 1
 1Some normal text for illustration, chapter 1.
12Chapter 2 should be printed in font GT10.
 1This is a normal print line and should be printed in font ST12.
 1
 1Some normal text for illustration, chapter 2.
============================================================
```

Following are examples of JCL in which an OUTPUT statement and a DD statement are used to indicate that TRCs are specified in the line-data records:

EXAMPLES

- This example uses the TRC parameter in the OUTPUT statement to specify that the line data set contains TRCs:

  ```
  //OUT2 OUTPUT CHARS=(GT10,GT12),TRC=YES
  //DD2 DD SYSOUT=A,OUTPUT=*.OUT2
  ```

- This example uses the CDB subparameter of the DD statement to specify that the data set contains TRCs:

  ```
  //OUT2 OUTPUT CHARS=(GT10, GT12)
  //DD2 DD SYSOUT=A,OUTPUT=*.OUT2,DCB=OPTCD=J
  ```

Rules for Coding Table Reference Characters

PSF uses the table reference character from the output line to select a font. When you are coding a table reference character in an output data line, remember the following:

- For table reference characters corresponding to font names specified in the CHARS parameter:
  - The valid table reference characters are 0, 1, 2, and 3. These are called compatibility TRCs, for compatibility with the 3800 Model 1 printer. PSF ignores the leftmost 4 bits of the table reference character. Thus, X'F0' and X'00' are both valid representations for zero.
  - A table reference character that refers to a font with a number higher than the number loaded by use of the CHARS parameter defaults to 0. For example, if 2 fonts are specified with CHARS, a table reference character of 2 (referring to a third font) defaults to 0 and selects the first font specified. A number that is not valid, such as 4, also defaults to 0.
- For table reference characters corresponding to font names specified in a page definition:
  - Valid table reference characters are from 0 to 127, inclusive. If 4 or fewer fonts are specified, they are treated as compatibility TRCs, and the leftmost 4 bits of the TRC are ignored. In this case, X'F0' and X'00' are both valid representations for 0. If more than 4 fonts are specified, PSF treats them like non-compatibility TRCs and reads all 8 bits. In this case, X'00' is 0, but X'F0' is decimal 240.
  - A table reference character that refers to a font with a number higher than the number of fonts specified in the page definition defaults to the first font in the page definition.
  - A table reference character of 0 selects the first font defined in the page definition.
  - A table reference character higher than 127 selects the first font defined in the page definition.
- If you specify either the TRC=YES or the DCB=OPTCD=J JCL parameter to indicate that table reference characters are specified, but none are in the data, you may get unexpected results.
- If you include table reference characters in the data but do not specify either the TRC=YES or the DCB=OPTCD=J JCL parameter, the table reference characters will be printed in your output as data.

Merging Data Lines into a Single Print Line

PSF can merge multiple input data records to print a single print line. When 2 or more lines of data that contain printable characters are printed in the same line space, the printed line can cause 2 or more characters to be superimposed. Thus you can print composite characters with line data. You can also use this function to create printed lines in which different fields are printed in different fonts.

Some line data applications merge data lines with lines containing mostly blanks and a few vertical bars. On line printers, this is 1 way to merge tabular application data with a line data electronic form. This method can more than double the amount of data per page, but at some cost in performance. On AFP printers, you

can greatly improve performance by using an overlay to print electronic forms. For more information, see “Overlays” on page 29.

If you type a *print-with-no-space* control character at the beginning of an input data line, that line will be superimposed on the next. The print-with-no-space control character indicates that no lines should be skipped before or after this line is printed. You can superimpose as many lines as you like by typing that control character at the beginning of each of those lines, except for the last line. If you are coding with machine-code print control characters, the print-with-no-space control character is X'01'. If you are coding with American National Standards Institute (ANSI), the print-with-no-space control character is X'4E' or the “+” sign.

Note: If a data set containing merged lines is created for printing on a line printer, a page printer may not produce the same results.

When you use PSF to merge lines that use fonts with different pitches or typographic fonts, be aware that lines are merged pel by pel, not character by character. When the job is printed, 10 characters in the first line align with 10 characters in the second line *only* if the characters in both lines have the same pel width. Otherwise overprinting may occur, as in Figure 41, where the data in the third record is superimposed on the data from the second record.

```
I AM THE FIRST LINE TO BE PRINTED
I AM THE SECOND AND LAST LINE MERGED WITH ANOTHER RECORD
```

Figure 41. Output Containing Merged Lines Printed with a Typographic Font

Example of Merging Data Lines

The application in Figure 42 shows 3 input data records, of which 2 are to be merged into a single print line.

```
//MERGEIT JOB
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=S,DCB=(RECFM=FBA,LRECL=80,BLKSIZE=2000)
//SYSUT1 DD *
+THIS IS THE FIRST LINE ON THE PAGE
 THIS IS THE SECOND LINE
+                        - THIS FRAGMENT IS MERGED WITH THE SECOND LINE
/*
//
```

Figure 42. IEBGENER Example of Merging 2 Print Lines

The resulting printed output lines are:

```
THIS IS THE FIRST LINE ON THE PAGE
THIS IS THE SECOND LINE - THIS FRAGMENT IS MERGED WITH THE SECOND LINE
```

In the example in Figure 42, to achieve the *merged* output line, blanks are entered in the third input data record to place “- THIS FRAGMENT IS MERGED WITH THE SECOND LINE” in the correct position.

By using different fonts in lines to be merged, you can highlight or subordinate data in a print line, or you can print different columns of data across a page in different styles or sizes. Or, alternatively, you can select fonts in a page definition and then specify their use for specific fields in a record.

Using Shift-Out, Shift-In (SOSI) Codes

A data set that contains both single-byte and double-byte character codes may necessitate coding the structured fields in the data set to change from 1 kind of character code to the other.[23] To avoid doing this, you can instruct PSF to provide special processing by specifying PRMODE=SOSI1 or PRMODE=SOSI2 in your JCL. The data set can then contain the shift-out code, X'0E', and the shift-in code, X'0F', to indicate to PSF when a font change is required. PSF converts the shift-out and shift-in codes to Set Coded Font Local text controls, described in *Mixed Object Document Content Architecture Reference*.

You can specify a particular single-byte and double-byte font pair for a line or field, or you can use the same single-byte and double-byte font pair for the entire page. If you specify a specific single-byte and double-byte font pair for a line or a field, you must use the font list in a page definition to specify the single-byte fonts and double-byte fonts you want to use for the page. Then you can use more than 1 single-byte font and more than 1 double-byte font per page. If you use the same single-byte and double-byte font pair for the entire page, you must specify the single-byte font first, and the double-byte font second.

Using the JCL CHARS parameter or the font list in a page definition, specify a single-byte font as font 0 and a double-byte font as font 1. As is explained in "Using Table Reference Characters to Select Fonts" on page 121, if you do not place font 0 and font 1 in this sequence, PSF generates an error message and terminates the printing process.

The conversion for PRMODE=SOSI1 is as follows:

1. Each X'0E' is replaced with a blank (X'40'), followed by a structure that contains a Set Coded Font Local text control for the second font.
2. Each X'0F' is replaced with a MO:DCA structure that contains a Set Coded Font Local text control for the first font, followed by a blank (X'40').

The conversion for PRMODE=SOSI2 is as follows:

1. Each X'0E' is replaced with a MO:DCA structure that contains a Set Coded Font Local text control for the second font.
2. Each X'0F' is replaced with a MO:DCA structure that contains a Set Coded Font Local text control for the first font.

The difference between these conversions is that in SOSI1, a blank replaces each code.

PSF assumes that each line starts with a single-byte font, which means that the data is scanned for the codes 1 byte at a time. After PSF processes a shift-out code, PSF scans the data 2 bytes at a time, examining only the first byte of each

[23] The PTX structured field is described in *Mixed Object Document Content Architecture Reference*.

pair. Because the scan starts 1 byte at a time on each line, if a line starts with double-byte font data, the first byte in the line must be a shift-out code, so that scanning is done 2 bytes at a time.

Notes:

1. When you use the same single-byte and double-byte font pair for the entire page, IBM does not recommend mixing the use of shift-out, shift-in processing with the use of table reference characters in line data, because of the rules used in the scanning process and because the fonts used for the shift-in and shift-out codes are always font 0 and font 1.
2. When you use the same single-byte and double-byte font pair for the entire page, IBM does not recommend mixing the use of shift-out, shift-in processing with the use of font lists in page definitions, for the reasons given in note 1. If you follow this procedure, you must know exactly what the data and the font list contain.
3. If your print job consists of multiple steps that alternate between PRMODE=SOSI1 and PRMODE=SOSI2, JES2 and JES3 will reorder the steps to process all those with 1 PRMODE setting first, and then process all those with the other PRMODE setting. The resulting output is different for each of the 2 JES systems.
4. IBM recommends that shift-in codes and shift-out codes alternate in a given record.

Including AFP Structured Fields in Line Data

You can imbed certain AFP structured fields in line data to change the formatting of selected pages in a data set or to include images or blocks of page data on a page. By mixing structured fields with line data, you can do the following:

- Change a copy group to change electronic overlays, duplexing, paper source, or field suppressions.
- Change a page format to change print direction, data formatting, or conditional processing specifications.
- Include page segments from a library.
- Include page overlays from a library.
- Include image data inline with the application print data.
- Use text-control sequences to include MO:DCA data, or to draw vertical and horizontal rules on a page.

For a description of the structured fields that can be mixed with line data, refer to *Mixed Object Document Content Architecture Reference*.

When you mix structured fields with line-data records, the following applies:

- All records in the data set must contain either the X'5A' control character or 1 of the valid carriage control characters listed in Table 3 on page 121. If no control characters are specified, the data set can contain only line-data records.
- The RECFM subparameter of the DCB parameter must specify “A” for American National Standard control characters or “M” for machine-code control characters.

- If the length of the structured field records is greater than or equal to the logical record length defined for the print data set, the record format of the data set must be specified as variable. The following record formats can be specified:

  Variable ANSI (VA)
  Variable machine (VM)
  Variable-blocked ANSI (VBA)
  Variable-blocked machine (VBM)

  If the length of the structured field records is less than the logical record length defined for the print data set, a fixed-length record format can be specified.

- Structured fields that change copy groups or page formats cause PSF to eject to a new page. The first print record following 1 of these structured fields will be the first record on the new page.

- Structured fields other than those that change copy groups or page formats do not affect the placement of line-data records, nor can they affect the font or the orientation used for printing the line-data records. These characteristics of line-data records are defined in the page definition. Line-data records mixed with structured fields print as defined in the page definition, regardless of whether structured fields are present. For example, if line-data record 2 prints on line 2 of the page when the data set contains no structured fields, it still prints in exactly the same position, on line 2 of the page, if a structured field to write MO:DCA data or to include a page overlay is written before it.

  A structured field can affect the printing of line-data records if it contains text-control sequences that change the spacing between characters and between words. Spacing between characters and between words in the line-data record will be changed, because these characteristics are not controlled in the page definition.

- Line-data records can affect the placement, the font, and the orientation of MO:DCA data defined with structured fields. If the font ID, the orientation, or the placement is not defined in the structured field, or if the placement is defined as “relative,” the values specified in the page definition for the current line-data record will be used for the MO:DCA data.

  If ANSI control characters are defined, the current line-data record is the line-data record that immediately precedes the structured field record. If machine-code controls are defined for the print data set, the current line-data record is the line-data record following the structured field record.

- Line-data records can also affect the placement of page segments, page overlays, and images. The structured fields used to include page segments or page overlays can specify absolute position relative to the page origin, or placement at the current print line position. Images included directly in a line data set are always placed relative to the current print line. When you include these objects in line data, you can control their position by controlling the placement of the current line-data record. This may involve writing a blank line-data record at the position desired, or using carriage control characters to place the current line-data record at the correct position.

To include a structured field in a print data set, you can either edit the print data set or include instructions in the application program to put the record in the print data set at the correct location, as in the segment from a sample application program shown in Figure 43 on page 130. A structured field is included between the fifth and sixth line-data records to invoke the page format DATAMAP2, which defines 90° rotated text with an underscored font.

Figure 43 on page 130 shows a segment of a sample application program that is writing an Invoke Data Map (IDM) structured field to change the page format. The IDM structured field is written between the fifth and sixth line-data records to invoke a new page format. The sixth line-data record will become the first print record on the new page to be formatted with the DATAMAP2 page format. Notice that the carriage control byte for the sixth line-data record contains “+” sign, which is the ANSI code to space 0 lines. If the carriage control byte is a blank (ANSI code to skip 1 line before printing), the sixth record will print not on line 1 of the new page, but on line 2.

The coding in Figure 43 shows the layout and the description of the IDM record. The structured field begins with the X'5A' carriage control byte. A record descriptor word (RDW) field is coded in front of the structured field, because this is an Assembler language program that is writing variable-length records. No RDW would be coded if fixed-length records were being written or if the program were written in a high-level language such as COBOL, which does not require user-coded RDW fields.

For information about coding carriage control bytes or variable-length records, refer to the publications for your programming language. For descriptions of the AFP structured fields, refer to *Mixed Object Document Content Architecture Reference*.

```
•
•
•
    OPEN (PRINT,(OUTPUT)) OPEN OUTPUT DATA SET
*******************************************************************************
*   PUT FIRST 5 LINES OF DATA                                                 *
*******************************************************************************
PUT PRINT,LINE1
•
•
PUT PRINT,LINE5
*******************************************************************************
*   PUT A Structured Field TO CHANGE THE PAGE FORMAT SO                       *
*   THAT THE TEXT IS ROTATED 90 AND PRINTED USING AN UNDER-                   *
*   SCORED FONT.  A NEW PAGE IS STARTED.                                      *
*******************************************************************************
PUT PRINT,IDM
*******************************************************************************
*   PUT REMAINING LINES OF DATA                                               *
*******************************************************************************
PUT PRINT,LINE6
•
•
PUT PRINT,LINE10
CLOSE PRINT                                 CLOSE OUT DATA SET
*******************************************************************************
*   DECLARES FOR FIRST 5 LINES OF DATA                                        *
*******************************************************************************
LINE1    DS OCL19                           FIRST LINE
RDW1     DC X'00130000'                     RECORD DESCRIPTOR WORD
DATA1    DC CL15'+THIS IS LINE 1'           CARRIAGE CONTROL AND DATA
•
•
LINE5    DS OCL19                           FIFTH LINE
RDW5     DC X'00130000'                     CARRIAGE DESCRIPTOR WORD
DATA5    DC CL15'+THIS IS LINE 5'           CARRIAGE CONTROL AND DATA
*******************************************************************************
*   DECLARES FOR THE Structured Field                                         *
*******************************************************************************
IDM      DS 0XL21                           INVOKE DATAMAP Structured Field
RDW      DC X'00150000'                     RECORD DESCRIPTOR WORD
CC       DC X'5A'                           5A CONTROL CHARACTER
SFILEN   DC X'0010'                         LENGTH OF Structured Field
SFICODE  DC X'D3ABCA'                       HEX CODE FOR INVOKE DATAMAP
SFIFLAGS DC B'00000000'                     FLAGS
SFISEQ   DC X'0001'                         SEQUENCE NUMBER
PARMS    DC C'DATAMAP2                      DATAMAP NAME
*******************************************************************************
*        DECLARES FOR SECOND 5 LINES OF DATA                                  *
*******************************************************************************
LINE6    DS OCL19                           SIXTH LINE
RDW6     DC X'00130000'                     RECORD DESCRIPTOR WORD
DATA6    DC CL15'+THIS IS LINE 6'           CARRIAGE CONTROL AND DATA
•
•
PRINT    DCB DSORG=PS,DDNAME=PRINT,MACRF=PM,BLKSIZE=25,                   X
              LRECL=21,RECFM=VBA
END
```

Figure 43. Sample Application Program. This segment of program includes a structured field with line data.

Application Development Tool: AFP Conversion and Indexing Facility (ACIF)

AFP Conversion and Indexing Facility (ACIF) is a PSF batch application development utility that you can use for several purposes:

- To convert line data or mixed data into MO:DCA-P data, which is an architected, device-independent data stream used for interchanging documents between different platforms.

  ACIF formats line data according to instructions in a page definition and converts it into an AFP data stream file. ACIF accepts any data stream that PSF Version 2 can process, including line-printer data, the AFP data stream, and MO:DCA-P IS/1. The output can then be sent to any AFP-supported system for printing or to a workstation for viewing.

- To index a document to improve viewing, archiving, or retrieving individual pages or groups of pages from large documents; or to create a separate *index object file* from the indexing tags.

- To retrieve and package AFP resources needed for printing or viewing a document and place them in a separate file, to enable viewing and printing of the exact document, possibly years after its creation.

With APAR PN67975, ACIF identifies and processes microfilm setup resources for creating AFP output to a microfilm device. For information about using ACIF, refer to *AFP Conversion and Indexing Facility: User's Guide*.

Advanced Function Presentation Application Programming Interface (AFP API)

AFP API is an application programming interface that produces the AFP data stream from a high-level programming language, enabling application programmers to format complex output applications without knowing the syntax and semantics of the AFP data stream. AFP API enables development of common, complex, production-printing applications with savings in time and improvements in the appearance of the output, because it provides high-level language access to advanced formatting functions in terms the application programmer can easily understand and use in existing applications. AFP API incorporates the COBOL and PL/I programming languages and is available with PSF 2.1.1 and later. For information about using AFP API, refer to *AFP Application Programming Interface: Programming Guide and Reference*.

Chapter 6. Printing MO:DCA Data

PSF can print data sets formatted as line data, as a mixture of line data and structured fields, or as Mixed Object Document Content Architecture (MO:DCA) data. MO:DCA data sets contain only structured fields, whose type and sequence must meet the specifications of the AFP architecture as defined in *Mixed Object Document Content Architecture Reference*.

You can write your own MO:DCA data, or you can use a product that creates the data for you. The following IBM licensed programs produce MO:DCA data for output on page printers:

Document Composition Facility (DCF)
: A general-purpose text-processing program that supports full-page composition of documents, including graphics and images. DCF includes a text formatter, SCRIPT/VS, that processes documents marked up with its own control words as well as documents marked up with Generalized Markup Language (GML) tags. GML tags, which are shorthand text markup, format the elements of the document.

Publishing Systems BookMaster (BookMaster)
: An implementation of GML that uses the SCRIPT/VS formatter to control the formatting of documents for printing or viewing.

DisplayWrite/370 (DW/370)
: A word-processing program that includes a full-screen text editor and formatter.

AFP Windows Driver
: An AFP Printer Driver, available for Windows and OS/2, that enables any Windows or OS/2 application to generate MO:DCA output for printing on any printer defined to PSF on the host, or on the LAN to PSF/2.

When you use any of these licensed programs to create a document for printing on an AFP printer, the document is created as a data set consisting of structured fields.

For information about formatting with DCF, refer to *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*. For information about formatting with BookMaster, refer to *Publishing Systems BookMaster User's Guide*. For information about formatting with DW/370, refer to *Using DisplayWrite/370*.

Advanced Function Presentation Application Programming Interface (AFP API)

AFP API is an application programming interface that produces the AFP data stream from a high-level programming language, enabling application programmers to format complex output applications without knowing the syntax and semantics of the AFP data stream. The AFP API provides high-level language access to advanced formatting functions in terms the application programmer can easily understand and use in existing applications. AFP API enables programmers to develop common, complex applications in production printing with savings in time and improvements in the appearance of the output. AFP API incorporates the

COBOL and PL/I programming languages and is available with PSF 2.1.1 and later. For information about using AFP API, refer to *AFP Application Programming Interface: Programming Guide and Reference*.

Formatting Data

MO:DCA data created by DCF, BookMaster, or DW/370 contains in its structured fields all the information necessary to describe how the data should be formatted on a page. You do not need a page definition to print MO:DCA data; if you specify a page definition, PSF ignores it.

A form definition is required to define such functions as page positioning, duplex printing, and overlays, among others. Both DCF and DW/370 enable you to change copy groups in the form definition, enabling functions such as duplex printing selectively in a print data set. BookMaster does not provide a tag for changing copy groups.

Unless you specifically code a FORMDEF parameter in the JCL for your job, PSF uses a default form definition. The default may contain controls, such as duplex printing, bin (paper source) selection, and page presentation, different from those you want for your job. For information about the FORMDEF parameter, see "Specifying a Form Definition" on page 91. For information about the contents of a form definition, see "Form Definitions" on page 67.

Selecting Fonts

The page output of DCF, BookMaster, and DW/370 contains the names of the fonts to be used for the print data set. For information about selecting fonts for DCF, refer to *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*. For information selecting fonts for BookMaster, refer to *Publishing Systems BookMaster User's Guide*. For information about selecting fonts for DW/370, refer to *Using DisplayWrite/370*.

Using JCL Parameters

When you print MO:DCA data, the RECFM subparameter of the DCB parameter must specify that the data set contains carriage control characters; for example, RECFM=VBA or RECFM=VBM. Because the only carriage control character used in MO:DCA data is the X'5A' for structured fields, you can specify either ANSI (VBA) or Machine (VBM) controls.

The following JCL parameters are ignored when you print MO:DCA data:

- FCB
- OPTCD=J
- PAGEDEF
- TRC
- UCS

For an explanation of the JCL parameters, see Chapter 2, "Using JCL for Advanced Function Presentation."

Appendix A. Form Definitions Supplied with PSF

This appendix describes the form definitions that IBM supplies with PSF.

Form Definition for the 3800 Printer

Table 4 describes the form definition supplied with PSF for the 3800 printer. The form definition specifies:

- One copy
- No overlays
- No offset stacking or copy marking

Table 4. Form Definition for the 3800

Form Definition Name	Copy Group	Page Position, In Inches	Flash
F10101	F20101	0.0, 0.5	No

Form Definitions for Printers Other than the 3800, PCL4, and PPDS Printers

Table 5 describes the form definitions for printers other than the 3800, PCL4, and PPDS printers, with the name and a description of each. Note that on some printers printing near the edge of the paper can result in poor print quality in the border area. Some of the form definitions enable you to use the third paper source, if the printer has one. For the limitations on your printer, refer to the publications for your printer. Each of these form definitions specifies:

- One copy
- No overlays
- No offset stacking or copy marking

Table 5. Form Definitions for All Printers Other than the 3800, PCL4, and PPDS Printers

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1A10110	F2A10110	0.165, 0.165	Duplex off	Primary
F1A10120	F2A10120	0.165, 0.165	Duplex off	Alternate
F1A10130	F2A10130	0.165, 0.165	Duplex off	Third
F1A10111	F2A10111	0.165, 0.165	Normal duplex	Primary
F1A10121	F2A10121	0.165, 0.165	Normal duplex	Alternate
F1A10131	F2A10131	0.165, 0.165	Normal duplex	Third
F1A10112	F2A10112	0.165, 0.165	Tumble duplex	Primary
F1A10122	F2A10122	0.165, 0.165	Tumble duplex	Alternate
F1A10132	F2A10132	0.165, 0.165	Tumble duplex	Third
F1A10140	F2A10140	0.165, 0.165	Duplex Off	Fourth
F1A10141	F2A10141	0.165, 0.165	Normal Duplex	Fourth
F1A10142	F2A10142	0.165, 0.165	Tumble Duplex	Fourth

Note: The 3820 lines up B4-size paper differently from other paper because of the B4 paper length. To compensate for this, you can create a form definition with a page position of 0.10, 0.00 inch.

Table 6 lists form definitions that enable you to print envelopes and use the manual input bin on the 4028 printer.

Table 6. Form Definitions for Printing Envelopes on the 4028

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1A101E0	F2A101E0	0.165,0.165	Duplex off	Envelope
F1A000E0	F2A000E0	0,0	Duplex off	Envelope
F1A101M0	F2A101M0	0.165,0.165	Duplex off	Manual
F1A000M0	F2A000M0	0,0	Duplex off	Manual

Table 7 lists form definitions that specify a 0,0 offset. These form definitions are for printing on printers other than the 3800 printer.

Table 7. Form Definitions with a 0,0 Offset

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1A00010	F2A00010	0,0	Duplex off	Primary
F1A00011	F2A00011	0,0	Normal duplex	Primary
F1A00012	F2A00012	0,0	Tumble duplex	Primary
F1A00020	F2A00020	0,0	Duplex off	Alternate
F1A00021	F2A00021	0,0	Normal duplex	Alternate
F1A00022	F2A00022	0,0	Tumble duplex	Alternate
F1A00030	F2A00030	0,0	Duplex off	Third
F1A00031	F2A00031	0,0	Normal duplex	Third
F1A00032	F2A00032	0,0	Tumble duplex	Third
F1A00040	F2A00040	0,0	Duplex Off	Fourth
F1A00041	F2A00041	0,0	Normal Duplex	Fourth
F1A00042	F2A00042	0,0	Tumble Duplex	Fourth

Table 8 lists form definitions for N_UP 2 printing. These form definitions define 2 pages on a side of a sheet. For more information on using the N_UP subcommand to create N_UP form definitions, refer to *Page Printer Formatting Aid: User's Guide and Reference*.

Table 8. Form Definitions for N_UP 2 Printing

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1N20110	F1N20110	0.165,0.165	Duplex off	Primary
F1N20111	F1N20111	0.165,0.165	Normal duplex	Primary
F1N20112	F1N20112	0.165,0.165	Tumble duplex	Primary
F1N20130	F1N20130	0.165,0.165	Duplex off	Third
F1N20131	F1N20131	0.165,0.165	Normal duplex	Third
F1N20132	F1N20132	0.165,0.165	Tumble duplex	Third

Table 9 lists form definitions to use with 3-hole-punched paper when printing with page definitions also designed for use with 3-hole-punched paper. You can use any of these form definitions with any of the page definitions for 3-hole-punched paper.

Table 9. Form Definitions for 3-Hole-Punched Paper

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1H10110	F2H10110	1.000,0.165	Duplex off	Primary
F1H10111	F2H10111	1.000,0.165 0.165,0.165	Normal duplex	Primary
F1H10112	F2H10112	1.000,0.165 1.000,0.165	Tumble duplex	Primary
F1H10120	F2H10120	1.000,0.165	Duplex off	Alternate
F1H10121	F2H10121	1.000,0.165 0.165,0.165	Normal duplex	Alternate
F1H10122	F2H10122	1.000,0.165 1.000,0.165	Tumble duplex	Alternate
F1H10130	F2H10130	1.000,0.165	Duplex off	Third
F1H10131	F2H10131	1.000,0.165 0.165,0.165	Normal duplex	Third
F1H10132	F2H10132	1.000,0.165 1.000,0.165	Tumble duplex	Third
F1H10140	F2H10140	1.000,0.165	Duplex Off	Fourth
F1H10141	F2H10141	1.000,0.165 0.165,0.165	Normal Duplex	Fourth
F1H10142	F2H10142	1.000,0.165 0.000,0.165	Tumble Duplex	Fourth

Form Definitions for HP PCL4 and PPDS Printers

The following table describes the form definitions supplied with PSF for printing on HP PCL4 or PPDS printers through PSF/2. These form definitions must be used in combination with page definitions intended for these printers. The form definitions specify:

- One copy
- No overlays
- No offset stacking or copy marking

Table 10. Form Definitions for HP PCL4 and PPDS Printers

Form Definition Name	Copy group	Page Position, In Inches	Duplex	Paper Source
F1CP0110	F2CP0110	0.250,0.200	Duplex off	Primary
F1CP0120	F2CP0120	0.250,0.200	Duplex off	Alternate
F1CP0111	F2CP0111	0.250,0.200	Normal duplex	Primary
F1CP0121	F2CP0121	0.250,0.200	Normal duplex	Alternate
F1CP0112	F2CP0112	0.250,0.200	Tumble duplex	Primary
F1CP0122	F2CP0122	0.250,0.200	Tumble duplex	Alternate

Compatibility Form Definitions for the 3831, 3835, and 3900 Printers

Table 11 describes the compatibility form definitions supplied with PSF for printing on the 3831, 3835, and 3900 printers. Compatibility form definitions make it possible to print data formatted for other printers. Use 1 of the compatibility form definitions when printing:

- Data formatted for landscape presentation on a cut-sheet printer.
- Data formatted for either portrait or landscape presentation on a 3800 printer. The form definition used depends on whether the data was formatted for printing on wide or narrow paper.

These form definitions specify the following:

- One copy group[24]
- 1 copy
- No offset stacking or copy marking
- Primary paper source
- No overlays

[24] The name of the copy group is the same as the name of the form definition, except for the prefix. For example, form definition F10101PA contains 1 copy group named F20101PA.

Table 11. Compatibility Form Definitions for the 3831, 3835, and 3900 Printers

Form Definition Name	Compatible With	Presentation Mode	Print Direction	Duplex	Page Position, In Inches
F1C10110	Cut-sheet printers	Landscape	Down	Duplex off	0.165, 0.165
F1C10111	Cut-sheet printers	Landscape	Down	Normal duplex	0.165, 0.165
F1C10112	Cut-sheet printers	Landscape	Down	Tumble duplex	0.165, 0.165
F10101PD	3800 printer wide forms	Portrait	Down	Duplex off	0.00, 0.50
F10101LA	3800 printer wide forms	Landscape	Across	Duplex off	0.00, 0.50
F10101PA	3800 printer narrow forms	Portrait	Across	Duplex off	0.00, 0.50
F10101LD	3800 printer narrow forms	Landscape	Down	Duplex off	0.00, 0.50

You can also use these form definitions when printing on the printers for which the data was formatted, eliminating the need to change form definitions. For example, you can use form definition F10101PA on a 3800 printer as well as on a 3831, 3835, or 3900 printer.

Table 12. N_UP Compatibility Form Definitions for the 3835 and 3900 Printers

Form Definition Name	Compatible With	Presentation Mode	Print Direction	Duplex	Page Position, In Inches	N_UP
F1N201PD	3800 printer wide forms (N_UP of None)	Portrait	Down	Duplex off	0.00, 0.50	2
F1N201LA	3800 printer wide forms (N_UP of None)	Landscape	Across	Duplex off	0.00, 0.50	2
F1N201PA	3800 printer narrow forms (N_UP of None)	Portrait	Across	Duplex off	0.00, 0.50	2
F1N201LD	3800 printer narrow forms (N_UP of None)	Landscape	Down	Duplex off	0.00, 0.50	2

Note: The use of the above N_UP form definitions on the 3800 printer will cause PSF to issue message APS283I or APS284I. The output will be printed appropriately.

For more information about compatibility form definitions, see “Maintaining Page-Presentation Compatibility” on page 74. For more information about when to use each of these form definitions, refer to *Advanced Function Presentation: Printer Information*.

Form Definitions for Special Purpose Jobs

Table 13 describes the form definitions supplied with PSF for special purposes. You can use these form definitions on any AFP printers. Each of these form definitions specifies:

- One copy
- No offset stacking or copy markings
- Primary paper source
- Duplex off
- No forms flash

Table 13. Form Definitions Supplied for Special Purposes

Form Definition Name	Copy Group	Page Position, In Inches	Overlay
F1IBM	IBM	0.0, 0.5	O1IBM
F1OGL	OGL/370	0.0, 0.0	none

Form definition F1IBM is provided for use in verifying the installation and prints a supplied overlay that contains the IBM logo. This form definition would ordinarily be used only during system installation, to verify that the PSF product is working correctly.

F1OGL is provided for use with the Overlay Generation Language/370 (OGL/370) program. F1OGL to print the overlay sample created during the processing of your OGL/370 statements. The overlay sample is a page, identical with the overlay resource, that can be sent directly to the printer. F1OGL positions the overlay sample page at the medium origin (0.0, 0.0), which ensures that the print position of the overlay sample matches the print position of the overlay resource when it is invoked for printing by a user-specified form definition. For more information on using OGL/370 to produce overlays and overlay samples, refer to *Overlay Generation Language/370: User's Guide and Reference*.

You can use F1OGL for printing any job in which you want the page to be positioned at the medium origin. However, if you position a page at the medium origin, you must ensure that data in the page does not fall outside the printable area of the sheet. For more information about the areas on which your printer can print, refer to *Advanced Function Presentation: Printer Information*.

Appendix B. Page Definitions Supplied with PSF

This appendix describes the page definitions supplied with PSF. Page definitions are available for the various paper sizes used by the printers supported by PSF. For information about the paper sizes your printer uses, refer to the publications for your printer.

Page Definitions for the 3800 Printer

The forms control buffer (FCB) modules available with the 3800 Model 1 (a line printer) have been converted to page definitions for the 3800 page printers (Models 3, 6, and 8). Table 14 lists the FCBs and the names of the corresponding page definitions with their descriptions. In the tables, the abbreviation *lpi* stands for lines per inch.

Table 14. 3800 Model 1 FCBs and Corresponding Page Definitions for 14.88 by 11-Inch Paper

FCB	Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recom-mended Font	Printing Direction	Page Presentation
FCB3STD1	P1STD1	13.87 × 8.5	51 at 6 lpi	29/0	GT10	Across	Landscape
FCB3STD2	P1STD2	13.87 × 10.0	60 at 6 lpi	29/0	GT10	Across	Landscape
FCB3STD3	P1STD3	13.87 × 10.0	80 at 8 lpi	23/0	GT12	Across	Landscape
FCB36	P16	13.87 × 8.5	51 at 6 lpi	29/0	GT10	Across	Landscape
FCB38	P18	13.87 × 8.5	68 at 8 lpi	23/0	GT12	Across	Landscape

Additional page definitions for the 3800 printer are available for formatting 3 of the more common paper sizes, 12 × 8.5, 9.5 × 11, and 14.88 × 11 inches, and for 2-up printing applications (placing 2 pages of application data on a form).

Many of these page definitions have 5-character names following the P1 prefix. To specify such a page definition with an FCB parameter, you must first rename the page definition with a 1- to 4-character name following the P1 prefix.

Each of these page definitions specifies a channel-1 (carriage) control character to position printing at the top of the next page.

[25] Position down and position across are shown in logical units, with 240 logical units per inch. For example, 24 logical units is equal to 0.10 inch. Measurements specified in logical units are the same for all the printers PSF supports, regardless of the resolution or pel-density of the printer.

Table 15 describes the page definitions for paper that measures 12 × 8.5 inches.

Table 15. Page Definitions for 12 by 8.5-Inch Paper

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P104560	11.0 × 7.5	45 at 6 lpi	29/0	GT10	Across	Landscape
P106061	7.5 × 10.0	60 at 6 lpi	29/0	GT10	Down	Portrait
P106080	11.0 × 7.5	60 at 8 lpi	23/0	GT12	Across	Landscape
P108081	7.5 × 10.0	80 at 8 lpi	23/0	GT12	Down	Portrait
P1075A0	11.0 × 7.5	75 at 10 lpi	19/0	GT20	Across	Landscape
P1100A1	7.5 × 10.0	100 at 10 lpi	19/0	GT20	Down	Portrait
P1090C0	11.0 × 7.5	90 at 12 lpi	15/0	GT20	Across	Landscape

Table 16 describes the page definitions for paper that measures 9.5 × 11.0 inches.

Table 16. Page Definitions for 9.5 by 11-Inch Paper

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P106060	8.5 × 10.0	60 at 6 lpi	29/0	GT10	Across	Portrait
P105161	10.0 × 8.5	51 at 6 lpi	29/0	GT10	Down	Landscape
P108080	8.5 × 10.0	80 at 8 lpi	23/0	GT12	Across	Portrait
P106881	10.0 × 8.5	68 at 8 lpi	23/0	GT12	Down	Landscape
P1100A0	8.5 × 10.0	100 at 10 lpi	19/0	GT20	Across	Portrait
P1085A1	10.0 × 8.5	85 at 10 lpi	19/0	GT20	Down	Landscape
P1120C0	8.5 × 10.0	120 at 12 lpi	15/0	GT20	Across	Portrait
P1102C1	10.0 × 8.5	102 at 12 lpi	15/0	GT20	Down	Landscape

Table 17 describes the page definitions for paper that measures 14.88 × 11.0 inches.

Table 17. Page Definitions for 14.88 by 11-Inch Paper

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1L06060	13.88 × 10.0	60 at 6 lpi	29/0	GT10	Across	Landscape
P1L08080	13.88 × 10.0	80 at 8 lpi	23/0	GT12	Across	Landscape
P1L100A0	13.88 × 10.0	100 at 10 lpi	19/0	GT20	Across	Landscape
P1L120C0	13.88 × 10.0	120 at 12 lpi	15/0	GT20	Across	Landscape

Table 18 describes the page definitions for multiple-up printing.

Table 18. Page Definitions for Multiple-Up Printing

Page Definition Name	Form Size: Width by Height, in Inches	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recom-mended Font	Printing Direction	Page Presentation
P1M12060	14.88 × 11.0	13.88 × 10.0	60 at 6 lpi 60 at 6 lpi	29/0 29/1728	GT12	Across	Landscape: Side-by-Side
P1M120C1	12 × 8.5	7.5 × 10.0	60 at 12.5 lpi 60 at 12.5 lpi	15/0 1197/0	GT20	Down	Portrait: Over/Under
P1M120C0	9.5 × 11.0	8.5 × 10.0	60 at 12.5 lpi 60 at 12.5 lpi	15/0 1245/0	GT20	Across	Portrait: Over/Under
P1M16080[26]	14.88 × 11.0	13.88 × 10.0	80 at 8 lpi 80 at 8 lpi	23/0 23/1728	[26]	Across	Landscape: Side-by-Side
P1M13280[27]	12 × 8.5	11.0 × 7.5	66 at 8.8 lpi 66 at 8.8 lpi	15/72 15/1380	GT24	Across	Landscape: Side-by-Side
P1M132C1[28]	12 × 8.5	7.5 × 11.0	66 at 12.1 lpi 66 at 12.1 lpi	15/0 next/0	GT20	Down	Portrait: Over/Under

The phrases *Side-by-Side* and *Over/Under* in Table 18 describe the placement of the subpages, which appear as separate logical pages on the physical sheet, as Figure 44 shows.

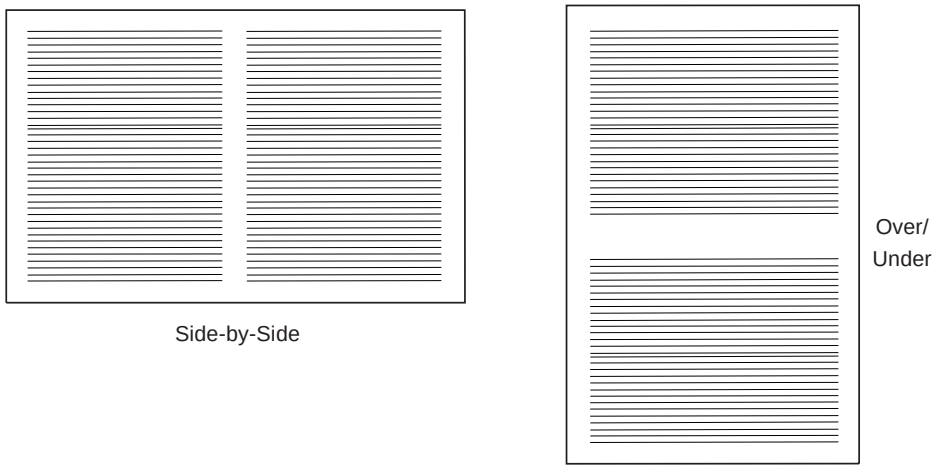


Figure 44. Placement of 2 Subpages on a Single Physical Sheet

[26] P1M16080 prints 80 characters per line when using a 12-pitch font, and 100 characters per line when using a 15-pitch font.

[27] Font GT24 is specified in this page definition; any CHARS option is ignored.

[28] Font GT20 is specified in this page definition; any CHARS option is ignored. When used with line data that contains no carriage controls, this page definition prints with no break between subpages.

Page Definitions for the 4224, 4230, 4234, and 4247 Printers

Page definitions for the 4224, 4230, 4234, and 4247 printers are supplied for formatting some of the more common paper sizes, as the following tables show. These page definitions are designed for printing on continuous-forms paper on any of the 4 printers named. They are not designed for printing on a 4224, 4230, or 4247 printer with cut-sheet paper or with the document-on-demand feature.

The 4224, 4230, 4234, and 4247 printers support different fonts, depending on the print-quality level set for the printer. Therefore, select a font that is supported at the print-quality level set for your printer. For more information, refer to *Advanced Function Presentation: Printer Information*. Each of these page definitions assigns a channel-1 (carriage) control character to the first line of the page.

Table 19 describes the page definitions for continuous-forms paper that measures 12 × 8.5 inches.

Table 19. Page Definitions for Continuous-Forms Paper 12 by 8.5 inches

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1J04964	10.67 × 8.17	49 at 6 lpi	30/0	GT10	Across	Landscape
P1J06484	10.67 × 8.17	64 at 8.01 lpi	30/0	GT12	Across	Landscape

Table 20 describes the page definitions for continuous-forms paper that measures 9.5 × 11 inches.

Table 20. Page Definitions for Continuous-Forms Paper 9.5 by 11 inches

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1A06462	8.17 × 10.67	64 at 6 lpi	30/0	GT10	Across	Portrait
P1A08584	8.17 × 10.67	85 at 8.01 lpi	30/0	GT12	Across	Portrait

Table 21 describes the page definitions for continuous-forms paper 14.88 × 11 inches. The width of the printable area in these page definitions is 13.2 inches, because the 4224, 4230, 4234, and 4247 printers have a maximum line length of 13.2 inches. To print records having a line length of 13.2 inches (for example, 132-byte records printed with a GT10 font), you must use a form definition that positions the page at the left margin. For example, you can use form definition F1OGL, described on page 140.

Table 21. Page Definitions for Continuous-Forms Paper 14.88 by 11 inches

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1L06464	13.2 × 10.67	64 at 6 lpi	30/0	GT10	Across	Landscape
P1L08584	13.2 × 10.67	85 at 8.01 lpi	30/0	GT12	Across	Landscape

Page Definitions for HP-CL4 and PPDS Printers

Page definitions for HP-CL4 and PPDS printers are supplied for formatting some of the more common paper sizes, as the following tables show. For information about the paper sizes supported for your printer, refer to the publications for your printer.

Each of these page definitions specifies a channel-1 (carriage) control character to position at the top of the next page.

Table 22 describes the page definitions for cut-sheet A4 paper, which is 8.27 inches wide by 11.69 inches high.

Table 22. Page Definitions for A4 Paper

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1Q09182	7.77 × 11.29	91 at 8.2 lpi	25/0	GT12	Across	Portrait
P1X04863	10.60 × 7.77	48 at 6.1 lpi	30/0	GT10	Down	Landscape
P1X06483	10.60 × 7.77	64 at 8.2 lpi	24/0	GT12	Down	Landscape
P1X06683	10.60 × 7.77	66 at 8.5 lpi	24/224 (28)	GT15	Down	Landscape.

Table 23 describes the page definitions for cut-sheet B4 paper, which is 10.12 inches wide by 14.33 inches high.

Table 23. Page Definitions for B4 Paper

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1T08362	9.62 × 13.93	83 at 6 lpi	30/0	GT10	Across	Portrait
P1T11382	9.62 × 13.93	113 at 8.2 lpi	24/0	GT10	Across	Portrait
P1T05963	13.93 × 9.62	59 at 6.1 lpi	30/0	GT10	Down	Landscape
P1T07983	13.93 × 9.62	79 at 8.2 lpi	24/0	GT 12	Down	Landscape

Table 24 describes the page definitions for cut-sheet letter paper, which is 8.5 inches wide by 11 inches high.

Table 24. Page Definitions for Cut-Sheet Letter Paper

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1P06362	8.00 × 10.60	63 at 6 lpi	30/0	GT10	Across	Portrait
P1P08682	8.00 × 10.60	86 at 8.2 lpi	24/0	GT12	Across	Portrait
P1X04863	10.60 × 7.77	48 at 6.1 lpi	30/0	GT10	Down	Landscape
P1X06483	10.60 × 7.77	64 at 8.2 lpi	24/0	GT12	Down	Landscape
P1X06683	10.60 × 7.77	66 at 8.5 lpi	24/224 (28)	GT15	Down	Landscape

Table 25 describes the page definitions for cut-sheet legal paper, which is 8.5 inches wide by 14 inches high.

Table 25. Page Definitions for Cut-Sheet Legal Paper

Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1R08162	8/00 × 13.60	81 at 6 lpi	30/0	GT10	Across	Portrait
P1R11082	8.00 × 13.60	110 at 8.2 lpi	24/0	GT12	Across	Portrait
P1R04863	13.60 × 8.00	48 at 6 lpi	30/0	GT10	Down	Landscape
P1R06683	13.60 × 8.00	66 at 8.2 lpi	24/0	GT12	Down	Landscape

Page Definitions for All Other Printers Supported by PSF

Table 26 describes the page definitions supplied with PSF for all printers other than the 3800, 4224, 4230, 4234, 4247, PCL4, and PPDS printers. PSF provides common page definitions for printers other than these three, to promote the interchange of documents between different printers.

PSF provides page definitions to format the commonly used cut-sheet and continuous-forms paper sizes. The following tables show the page definitions for each paper size. For information about the paper sizes supported for your printer, refer to the publications for your printer.

Each of these page definitions specifies a channel-1 (carriage) control character to position at the top of the next page.

Table 26 describes the page definitions for cut-sheet A4 paper, which is 8.27 inches wide by 11.69 inches high.

Table 26. Page Definitions for A4 Paper

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1C09182	7.94 × 11.36	91 at 8.2 lpi	25/0	GT12	Across	Portrait
P1V04863	10.67 × 7.94	48 at 6.1 lpi	30/0	GT10	Down	Landscape
P1V06483	10.67 × 7.94	64 at 8.2 lpi	24/0	GT12	Down	Landscape
P1V06683	10.67 × 7.94	66 at 8.5 lpi	24/224 [29]	GT15	Down	Landscape

[29] The user-printable area is 9.74 × 7.94 because of the 224-logical-unit offset in Position Down/Position Across column.

Table 27 describes the page definitions for cut-sheet B4 paper, which is 10.12 inches wide by 14.33 inches high.

Table 27. Page Definitions for B4 Paper

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1D08462	9.79 × 14	84 at 6 lpi	30/0	GT10	Across	Portrait
P1D11382	9.79 × 14	113 at 8.2 lpi	24/0	GT12	Across	Portrait
P1D06063	14 × 9.79	60 at 6.1 lpi	30/0	GT10	Down	Landscape
P1D08083	14 × 9.79	80 at 8.2 lpi	24/0	GT12	Down	Landscape

Table 28 describes the page definitions for any of the following paper sizes:

- Cut-sheet: Letter, which is 8.5 inches wide by 11 inches high
- Continuous-forms: 12 inches wide by 8.5 inches high
- Continuous-forms: 9.5 inches wide by 11 inches high

Table 28. Page Definitions for Letter and Continuous-Forms Paper 12 x 8.5 inches or 9.5 x 11 Inches

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1A06462	8.17 × 10.67	64 at 6 lpi	30/0	GT10	Across	Portrait
P1A08682	8.17 × 10.67	86 at 8.2 lpi	24/0	GT12	Across	Portrait
P1V04863	10.67 × 7.94	48 at 6.1 lpi	30/0	GT10	Down	Landscape
P1V06483	10.67 × 7.94	64 at 8.2 lpi	24/0	GT12	Down	Landscape
P1V06683	10.67 × 7.94	66 at 8.5 lpi	24/224 [29]	GT15	Down	Landscape

Table 29 describes the page definitions for any of the following paper sizes:

- Cut-sheet: Legal, which is 8.5 inches wide by 14 inches high
- Continuous-forms: 14.88 inches wide by 11 inches high

Table 29. Page Definitions for Legal and Continuous-Forms Paper 14.88 x 11 inches

Page Definition Name	Printable Area Width by Height, In Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1B08262	8.17 × 13.67	82 at 6 lpi	30/0	GT10	Across	Portrait
P1B11082	8.17 × 13.67	110 at 8.2 lpi	24/0	GT12	Across	Portrait
P1B04963	13.67 × 8.17	49 at 6 lpi	30/0	GT10	Down	Landscape
P1B06683	13.67 × 8.17	66 at 8.2 lpi	24/0	GT12	Down	Landscape

Table 30 describes the page definitions for multiple-up printing on any of the following paper sizes:

- Cut-sheet: Letter, which is 8.5 inches wide by 11 inches high
- Cut-sheet: A4, which is 8.27 inches wide by 11.69 inches high
- Continuous-forms: 12 inches wide by 8.5 inches high
- Continuous-forms: 9.5 inches wide by 11 inches high

Table 30. Multiple-up Page Definitions

Page Definition Name	Printable Area: Width by Height, in Inches	Print Lines Per Page	Page Position:[25] Down/Across	Recommended Font	Printing Direction	Page Presentation
P1W120C2	7.94 × 10.67	60 at 12 lpi 60 at 12 lpi	16/160 1344/160	GT20	Across	Portrait: Over/Under
P1W12883	10.67 × 7.94	64 at 8.2 lpi 64 at 8.2 lpi	24/0 24/1281	GT15	Down	Landscape: Side-by-Side
P1W240F3	10.67 × 7.94	60 at 15.2 lpi 60 at 15.2 lpi 60 at 15.2 lpi 60 at 15.2 lpi	16/48 968/48 16/1322 968/1322	GT24	Down	Landscape: Over/Under Side-by-Side

The phrases *Side-by-Side* and *Over/Under* in Table 30 on page 148 describe the placement of the subpages, which appear as separate logical pages on the physical page, as seen in Figure 45.

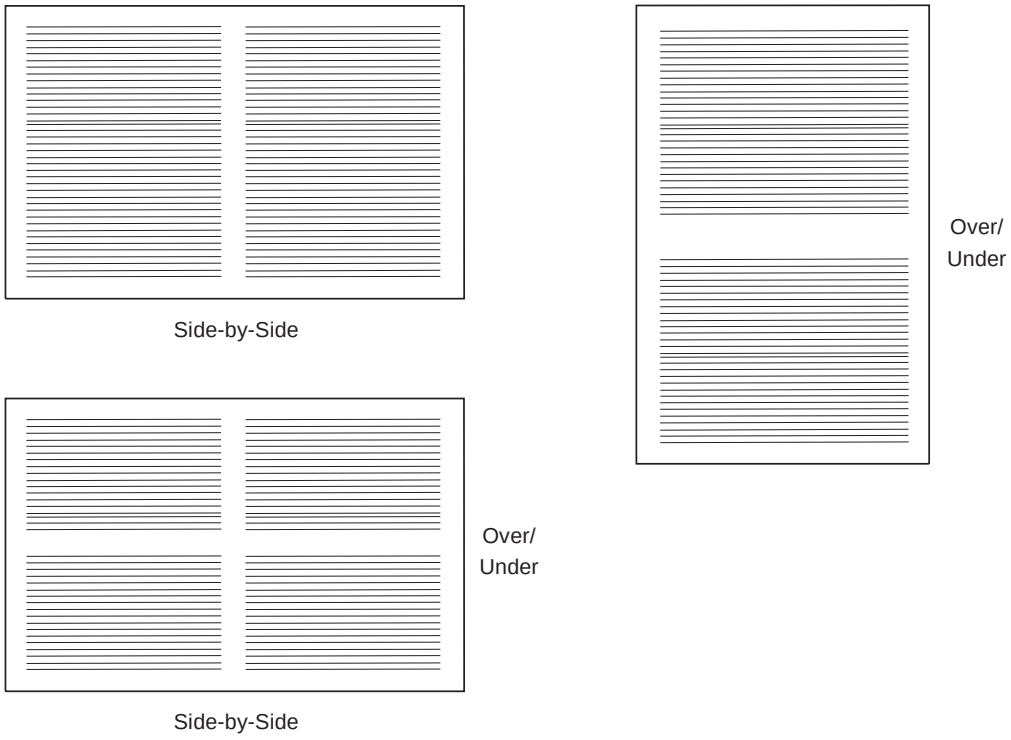


Figure 45. Placement of Multiple-Up Logical Pages on the Physical Sheets

Table 31 lists page definitions for printing on a variety of sizes of 3-hole-punched paper. The page definitions specify that the printing direction is Up, and they offset printing to accommodate 3-hole-punched paper. You must use these page definitions with the form definitions whose names begin with F1H.

Table 31. Page Definitions for Printing on 3-Hole-Punched Paper

Page Definition Name	Printable Area: Width by Height, In Inches	Print Lines Per Page	Page Position: Down/Across [25]	Recommended Font	Printing Direction	Page Presentation
P1B0446B	13.67 × 7.34	44 at 6 lpi	30/0	GT10	Up	Landscape
P1B0608B	13.67 × 7.34	60 at 8.2 lpi	24/0	GT12	Up	Landscape
P1D0556B	14.0 × 8.96	55 at 6.1 lpi	30/0	GT10	Up	Landscape
P1D0748B	14.0 × 8.96	74 at 8.2 lpi	24/0	GT12	Up	Landscape
P1V0436B	10.67 × 7.11	43 at 6.1 lpi	30/0	GT10	Up	Landscape
P1V0588B	10.67 × 7.11	58 at 8.2 lpi	24/0	GT12	Up	Landscape
P1V0608B	10.67 × 7.11	60 at 8.5 lpi	24/224	GT15	Up	Landscape
P1W1168B	10.67 × 7.11	58 at 8.2 lpi 58 at 8.2 lpi	24/0 24/1281	GT15	Up	Landscape
P1W216FB	10.67 × 7.11	54 at 15.2 lpi 54 at 15.2 lpi 54 at 15.2 lpi 54 at 15.2 lpi	16/48 890/48 16/1322 890/1322	GT24	Up	Landscape

Page Definition Line-Spacing Values and Fonts

The following tables list the line-spacing values for some of the page definitions supplied with PSF. The tables also list fonts by group according to the lines per inch at which each font is printed.

Table 32 shows the line-spacing values for the 3800 printer.

Table 32. Cross-Reference of Line Spacing and Page Definitions for the 3800 Printer

Line Spacing	Page Definition Name
6 lpi	P104560 P105161 P106060 P106061 P1L06060 P1M12060 P1STD1 P1STD2 P16
8 lpi	P106080 P106881 P108080 P108081 P1L08080 P1M16080 P1STD3 P18
10 lpi	P1075A0 P1085A1 P1100A0 P1100A1 P1L100A0
12 lpi	P1090C0 P1120C0 P1102C1 P1L120C0
12.5 lpi	P1M120C0 P1M120C1
Special	P1M13280 [30] P1M132C1 [31]

[30] P1M13280 specifies font GT24, which is printed at 8.8 lines per inch. Any CHARS option is ignored.

[31] Fonts AE20 and GT20 are printed at 12.8 lines per inch. Page definition P1M132C1 defines printing at 12.1 lines per inch, specifying font GT20. If you specify any other font with a CHARS option, the specification is ignored. To print an application using font AE20, use PPFA to change the source for P1M132C1, specifying font AE20. Rename the object and store it in the system page definition library.

Table 33 shows the line-spacing values for each of the page definitions for the 4224, 4230, 4234, and 4247 printers.

Table 33. Cross-Reference of Line Spacing and Page Definitions for the 4224, 4230, 4234, and 4247 Printers

Line Spacing	Page Definition Name
6 lpi	P1J04964 P1J06462 P1L06464
8.01 lpi	P1J06484 P1A08584 P1L08584

Table 34 shows the line-spacing values for each of the page definitions for cut-sheet A4 paper for the PCL4 and PPDS printers.

Table 34. Cross-Reference of Line Spacing and Page Definitions for Cut-Sheet A4 Paper for the PCL4 and PPDS Printers

Line Spacing	Page Definition Name
6.1 lpi	P1X04763
8.2 lpi	P1Q09182 P1X06483
8.5 lpi	P1X06683

Table 35 shows the line-spacing values for each of the page definitions for cut-sheet B4 paper for the PCL4 and PPDS printers.

Table 35. Cross-Reference of Line Spacing and Page Definitions for Cut-Sheet B4 Paper for the PCL4 and PPDS Printers

Line Spacing	Page Definition Name
6 lpi	P1T08362
6.1 lpi	P1T05963
8.2 lpi	P1T11382 P1T07983

Table 36 shows the line-spacing values for each of the page definitions for all other printers.

Table 36. Cross-Reference of Line Spacing and Page Definitions for Other Printers

Line Spacing	Page Definition Name
6 lpi	P1D08462 P1B08262 P1B04963 P1A06462 P1B0446B
6.1 lpi	P1D06063 P1V04863 P1D0556B P1V0463B
8.2 lpi	P1C09182 P1D11382 P1D08083 P1A08682 P1V06483 P1B11082 P1B06683 P1W12883 P10608B P1D0748B P1V0588B P1W1168B
8.5 lpi	P1V06683 P1V0708B
12 lpi	P1W120C2
15.2 lpi	P1W240F3 P1W116FB

Any font that can be printed at the specified line spacing (or at a larger line-spacing value) can be used with a page definition. Table 37 describes the line spacing for commonly used monospace fonts provided with PSF.

Table 37. Cross-Reference of Line Spacing and Commonly Used PSF Monospaced Fonts

Line Spacing	Fonts
6 lpi	CE10 CE12 CI10 CR10 GB10 GF10 GS10 GT10 LB12 LR12 PB12 PI12 PR10 PR12
8 lpi	GB12 GF12 GF15 GI12 GP12 GS12 GS15 GT12 GT15
10 lpi	GC15
12 lpi	GFC GSC GUC
Special	AE20 [32] GT20 [32] GT24 [33]

For more information about these fonts or about fonts not listed in this table, refer to the font technical references and the font samples publications listed under Appendix E, “Related Publications.”

[32] Fonts AE20 and GT20 are printed at 12.8 lines per inch. Font GT20 is specified in page definition P1M132C1. To print an application using font AE20, you can use the PPFA program to change the source for P1M132C1 to specify font AE20. Rename the page definition and store it in the system page definition library.

[33] Font GT24 is printed at 8.8 lines per inch as defined in page definition P1M13280.

Appendix C. Page-Printer Defaults Form

You can copy the page-printer defaults form on the next page and complete it with the defaults assigned to the printers in your installation. Obtain the default information from your system-support group, and complete a copy for each different CLASS, DEST, FLASH, and FORM value for each printer or group of printers that has a unique set of defaults.

PAGE-PRINTER DEFAULTS FORM

PAGE-PRINTER Type is: _____

Printer selection parameters: CLASS = _____

DEST = _____

FLASH = _____

1. Media or form usually loaded in printer:
   For continuous-forms printer, size of form usually loaded: _____

   For continuous-forms printer with BTS, output usually burst? YES | NO

   For cut-sheet printer, type of media usually in each source:

   Bin 1 _____ Bin 2 _____

   Bin ___ _____ Bin ___ _____
2. DATACK = BLOCK | UNBLOCK | BLKCHAR | BLKPOS
3. CKPTPAGE = _____ or CKPTSEC = _____
4. PIMSG = NO | YES Message count = _____
5. FORMDEF = _____

   &Copy Group = _____

   Page Position = _____

   Paper source for cut-sheet printers: BIN1 | BIN2 | BIN ___

   Duplex: NO | NORMAL | TUMBLE | RNORMAL | RTUMBLE

   For 3800: Flash activated? YES | NO
6. PAGEDEF = _____

   Form size = _____

   No. of print lines = _____

   Printable area = _____

   Lines per inch = _____

   Print direction: ACROSS | DOWN | BACK | UP
7. Default fonts are (listed in TRC sequence):

   0 = _____ 1 = _____ 2 = _____ 3 = _____

   If PAGEDEF is not specified in JCL:

   Fonts CAN | CANNOT be changed by using CHARS[34]
8. Fonts supported by printer: Resident | Host | Both

   Double-byte fonts supported? YES | NO

   If yes, default PRMODE: SOSI1 | SOSI2

   PSF exit APSUX07 has raster fonts automatically mapped to outline fonts for this printer: YES | **NO**
9. Other characteristics of this printer:

   _____
10. COMSETUP = _____ (&md.s only)
11. Where are the messages sent: **Printed with output** | Printed on printer _____ | On output class _____

[34] Fonts can be changed unless a JES default FCB (page definition) defined for this printer specifies fonts.

Appendix D. Microfilm Device Considerations

With PSF/MVS APAR OW10067, you can use the same data set for printing jobs on paper output or on microfilm. You should specify the COMSETUP parameter in your JCL to transfer microfilm setup resource data to the microfilm device for processing; otherwise the host default setup resource that your system programmer created will be used, and your output may not appear as you expected. For information on creating a microfilm setup resource, refer to *Print Services Facility/MVS: System Programming Guide*.

This appendix presents considerations for when you are creating a microfilm job both for printing to paper and for sending to microfilm devices.

Images on a Page

Image data representing characters cannot be accessed as extracted text on the microfilm device. Data used in text extraction processing must be codepoint-defined.

Carefully consider the shading patterns used in the job. Jobs printed on paper can use much finer gradations of shading than jobs printed on microfilm. For microfilm jobs, IBM recommends that jobs use only 3 shading patterns: none, 15% for 50% shading, and 30% to 100% for 100% shading.

Complicated images using different gray levels may not print on microfilm with the same quality as on paper. This quality consideration applies to logos created as bitmaps (raster images) or with graphic drawing orders.

Graphics on a Page

Graphic data representing characters cannot be accessed as extracted text on the microfilm device.

Complicated graphics using different gray levels may not print on microfilm with the same quality as on paper. This quality consideration applies to logos created as bitmaps (raster images) or with graphic drawing orders.

Form Definitions

If you are printing on cut-sheet printers, use a form definition with N_UP 1 or PORTRAIT ACROSS. Otherwise your output can rotate incorrectly.

If you are printing on a 3800 printer, your application must be compatible with the 3900 continuous-forms printers. Otherwise your output can rotate incorrectly. See “Maintaining Page-Presentation Compatibility” on page 74.

Consider using different form definitions and page definitions for AFP print jobs and for microfilm jobs. For example, print jobs may need to contain constant forms, but microfilm jobs may not. Also, microfilm devices print in simplex mode.

A form definition, a page definition, and the microfilm setup resource data interact together to present an AFP page on microfilm. Coordination between resources is

necessary, and in some cases a test run is recommended. Work with your system programmer to set up a test run.

If you use N_UP to place multiple pages in partitions on a sheet, your output may be unreadable on a microfilm device.

Using Fonts

Carefully consider the point size of the fonts used in your job. Fonts smaller than 6 points may produce unreadable data.

Work with your system programmer to set up different job classes for 120-pel or 240-pel resolution jobs and for 300-pel resolution jobs.

Using 120-pel resolution will give a higher printing rate than a higher resolution, but it may not produce the quality of output you need. Run some valid samples before archiving the data at 120-pel resolution.

Fonts for printing on microfilm devices must contain global resource identifier (GRID) data. If you are using fonts in user libraries or inline in your data stream, consult your system programmer to make sure that the fonts include GRID data. Refer to *Advanced Function Presentation: Printer Information* for more information on GRID's.

Page Segments

Text data in a page segment that is not in AFP text (PTOCA) format cannot be used for text extraction on a microfilm device.

Overlays

Text data (image or graphic) in an overlay that is not in AFP text (PTOCA) format cannot be used for text extraction on a microfilm device.

Specifying Parameters

Specify COMSETUP=your comsetup in your JCL. This COMSETUP parameter must reside in a system library, in a specified user library, or inline in your data stream. For more information on the COMSETUP parameter, see page33 .

To prevent data checks from being reported, specify DATACK=BLOCK in your JCL.

Avoid using checkpointing (the CKPTPAGE and CKPTSEC JCL parameters) for microfilm jobs.

Specify COPIES=1; specifying multiple copies will produce extra copies on the microfilm.

Additional Parameters to Help in Distributing Output

The following information can be passed to a microfilm device for extraction and is accessible through commands in your microfilm setup resource.

- NAME=*name* (See “Additional Parameters to Help in Distributing Output” on page 62.)
- ROOM=*name* (See “Additional Parameters to Help in Distributing Output” on page 62.)

Printing Constant Forms

Specifying CONSTANT BACK and COPIES in your form definition causes extra pages on microfilm. You may not want constant data on microfilm.

Other Considerations

Your system programmer may have caused the error messages that PSF prints at the end of your print data sets to be redirected to another destination. To ensure that the data set printed as you expected, consult your system programmer to determine the destination.

If you are using Print Services Facility for AIX, use Advanced Facility Presentation: Conversion and Indexing Facility (ACIF) to package the resources. For more information, refer to the publications on Print Services Facility for AIX.

Appendix E. Related Publications

The titles and the order numbers for IBM publications can change from time to time. To verify the current title or order number for a publication, consult your IBM marketing representative.

Advanced Function Presentation

Publication	Order number
Advanced Function Presentation: Printer Information	G544-3290
Advanced Function Presentation: Printer Summary	G544-3135
Advanced Function Printing: Diagnosis Guide	LH40-0201
Guide to Advanced Function Presentation	G544-3876

Application System/400

Publication	Order number
AS/400 Printer Device Programming	SC41-3713
AS/400 Data Description Specifications	SC41-9620
AS/400 Command Language Reference	SC41-3722

Architecture

Publication	Order number
Advanced Function Presentation: Programming Guide and Line Data Reference	S544-3884
Bar Code Object Content Architecture Reference	S544-3766
Graphics Object Content Architecture Reference	SC31-6804
Font Object Content Architecture Reference	S544-3285
Image Object Content Architecture Reference	SC31-6805
Presentation Text Object Content Architecture Reference	SC31-6803
Intelligent Printer Data Stream Reference	S544-3417
Mixed Object Document Content Architecture Reference	SC31-6802

Fonts

Publication	Order number
IBM AFP Fonts: Font Summary	G544-3810
IBM AFP Fonts: Introduction to Typography	G544-3122
IBM AFP Fonts: Font Samples	S544-3792
IBM AFP Fonts: Type Transformer User's Guide	G544-3796
IBM AFP Fonts: Technical Reference for IBM Expanded Core Fonts	S544-5228
IBM AFP Fonts: Technical Reference for Code Pages	S544-3802

MVS/ESA

Publication	Order number
OS/390 MVS JCL Reference	GC28-1757
MVS/ESA JCL Reference	GC28-1829

Print Services Facility/MVS

Publication	Order number
Print Services Facility/MVS: Diagnosis Guide and Reference	G544-5462
Print Services Facility/MVS: System Programming Guide	S544-3672
Print Services Facility: Security Guide	S544-3291
AFP Conversion and Indexing Facility: User's Guide	S544-5285
AFP Application Programming Interface: Programming Guide and Reference	S544-3872
AFP Application Programming Interface: COBOL Language Reference	S544-3873
AFP Application Programming Interface: PL/1 Language Reference	S544-3874

Print Services Facility/2

Publication	Order number
IBM Print Services Facility for OS/2: An Installation Cookbook for System/370 and Token Ring Networks	G544-3965
IBM Print Services Facility for OS/2: An Installation Cookbook for AS/400 and Token Ring Networks	G544-3966
IBM Print Services Facility for OS/2: Network Configuration Guide for System/370 and Communication Manager/2	S544-3911
IBM Print Services Facility for OS/2: A Guide to Using PSF/2	G544-5225
IBM Print Services Facility for OS/2: Facts about PSF/2	G544-3890
IBM Print Services Facility for OS/2: Printer Attachments Guide	G544-5215

Printers

Publication	Order number
Advanced Function Presentation: Printer Information	G544-3290

Programs That Create Resources

Publication	Order number
Overlay Generation Language/370: User's Guide and Reference	S544-3700
Page Printer Formatting Aid: User's Guide and Reference	S544-5284

Text Processing

Publication	Order number
Document Composition Facility: SCRIPT/VS Text Programmer's Guide	SH35-0069
Document Composition Facility: Bar Code User's Guide	S544-3115
Publishing Systems BookMaster General Information	GC34-5006
Publishing Systems BookMaster User's Guide	SC34-5009
DCF/DLF General Information	GH20-9158
Using DisplayWrite/370	SH12-5172

Glossary

Source Identifiers

This publication includes terms and definitions from the *IBM Dictionary of Computing,* ZC20-1699.

Each definition reprinted from the *American National Dictionary for Information Processing Systems* is identified by the symbol (A) at the end of the definition.

Each definition reprinted from a published section of the International Organization for Standardization (ISO) *Vocabulary—Information Processing* or from a published section of *Vocabulary—Office Machines,* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1), is identified by the symbol (I) at the end of the definition. Because many ISO definitions are also reproduced in the *American National Dictionary for Information Processing Systems,* ISO definitions may also be identified by the symbol (A).

Each definition reprinted from working documents, draft proposals, or draft international standards of ISO Technical Committee 97, Subcommittee 1 (Vocabulary), Joint Technical Committee 1, is identified by the symbol (T) at the end of the definition, to indicate that final agreement has not yet been reached among its participating members.

Definitions specific to IBM products are so labeled; for example, “In SNA,” or “In the 3820.”

References

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposite or substantively different meaning.

See. This refers to multiple-word terms in which this term appears.

See also. This refers to related terms that have similar, but not synonymous, meanings.

Synonym for. This appears in the commentary on a less desirable or less specific term, and identifies the preferred term that has the same meaning.

Synonymous with. This appears in the commentary on a preferred term and identifies less desirable or less specific terms that have the same meaning.

A

abend. An abnormal end of task before the task's completion, which is due to an error condition and which cannot be recovered while the task is running.

access control. In computer security, the methods and facilities used to ensure that a computer system and the data, system software, and application programs stored in it can be accessed only by authorized users in authorized ways.

access method. A method for moving data between main storage and I/O devices.

accumulator. (1) A 3800 hardware feature that supplies a separate storage area to hold data in raster form. The storage area can be used either for composing a sheet of data that combines a large amount of variable and constant data, or for storing an electronic overlay in raster form so that the overlay is merged with variable data while the page is printed. (2) A register in which the result of an operation is formed.

action code. A code, generated by software or hardware, that indicates a recovery action. The hardware action code is byte 2 of the sense data.

active environment group. A collection of mapping structured fields, positioning controls, and data descriptors that define the environment for a page. These structured fields form a internal object in MO:DCA data, a page definition, or an overlay.

addressable point. For page printers, any point on a presentation surface that can be addressed. Synonymous with *picture element*.

Advanced Function Presentation (AFP). A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information. See *presentation device*.

AFP. See *Advanced Function Presentation.*

all points addressability (APA). The ability to address, reference, and position text, overlays, and images at any defined position or pel on the printable area of the paper. This capability depends on the ability of the hardware to address and to display each picture element.

APA. See *all-points-addressability.*

APAR. See *authorized program analysis report.*

application program. A program written for or by a user that applies to the user's work, such as a program that performs inventory control or payroll tasks.

asynchronous. Without regular time relationship; unexpected or unpredictable with respect to running program instructions.

authorized program analysis report (APAR). A report of a problem caused by a suspected defect in a current, unaltered release of a program.

auxiliary data set. A data set that contains job header, data set header, job trailer, or message data. Contrast with *print data set*.

auxiliary resource. Fonts, page segments, overlays, page definitions, or form definitions associated with auxiliary data sets.

B

baseline. In a font, the imaginary line on which successive characters are aligned in the inline direction.

bin. A paper supply on a cut-sheet printer. See also *cassette*.

BIND. The command that VTAM sends to a logical unit to establish a VTAM program-to-logical unit session.

bounded-box font. A font designed to use bounded character boxes. Contrast with *unbounded-box font*.

bounded-box relative metrics. Units of measure that are expressed in 1000 units per Em-square.

bounded character box. A character-box design that does not require character-positioning information to be included (as untoned pels) in the box. Contrast with *unbounded character box*.

BTS. See *burster-trimmer-stacker.*

buffer pool. An area of storage in which all the buffers of a program are kept.

buffered pages. Pages kept in printer control storage, waiting to be printed.

burst. To separate continuous-forms paper into single sheets.

burster-trimmer-stacker (BTS). A printer hardware feature that separates continuous forms into single sheets, trims the carrier strip from both edges of the forms, and stacks the sheets.

byte multiplexer channel. A multiplexer channel that interleaves bytes of data.

C

carriage control character. An optional character in an input data record that specifies a write, space, or skip operation.

cassette. In a cut-sheet printer, a movable enclosure for paper supply. See also *bin*.

CCW. See *channel command word.*

channel attached. In PSF, linked to the host system only by means of System/390 channel protocols. For example, a 3800 printer cabled to the host system with a System/390 channel adapter is considered a channel-attached printer. Contrast with *SNA-attached*.

channel code. A number from 1 to 12 that identifies a position in the forms control buffer or a page definition. A carriage control character can select a position defined by a particular channel code.

channel command word (CCW). A command that directs data channel operations. CCWs built by Print Services Facility are sent to channel-attached printers.

channel counter. A device that counts the pages that have been successfully received. Synonymous with *received page counter*.

channel station. In the printer, a device that counts the pages queued for printing.

character. (1) A symbol used in printing. For example, a letter of the alphabet, a numeral, a punctuation mark, or any other symbol that represents information. (2) A byte of data.

character box. The area that completely contains the raster pattern comprising the graphic character.

character data. Data in the form of letters and special characters such as punctuation marks. See also *numeric data*.

character graphic. A visual representation of a character, other than a control character, that is normally produced by writing, printing, or displaying. (T)

character group. Any number of character graphics and character properties.

character identifier. The standard identifier for a character regardless of its style. For example, all uppercase A's have the same character identifier. Synonymous with *graphic character identifier*.

character increment. The distance between the current print position and the next print position.

character rotation. The alignment of a character with respect to the baseline, measured clockwise in degrees. Examples are 0°, 90°, 180°, and 270°.

character set. (1) A finite set of characters upon which agreement has been reached and that is considered complete for some purpose; for example, each of the characters in ISO Recommendation R646 "6- and 7-bit coded character sets for information processing interchange." (2) For page printers, the font library member that contains the character graphics and their descriptions.

checkpoint. A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later. For any print job, the user can specify that PSF take checkpoints on the basis either of the number of logical pages processed or the number of processing seconds elapsed.

coded font. A font library member that associates a code page and a font character set. For double-byte fonts, a coded font associates more than 1 pair of code pages and font character sets.

coded font local identifier. A 1-byte identifier that the Map Coded Font structured field assigns to each coded font it selects. The identifier is then specified in the text-control sequence that precedes the string of text to be printed with the particular font.

coded font section. A font character set and code page pair. A single-byte coded font consists of only 1 coded font section; a double-byte coded font can consist of more than 1 coded font section.

coded overlay. An overlay that is stored in the printer in a coded (not raster) format. Contrast with *raster pattern overlay*.

code page. A font component that associates code points with character identifiers. A code page also identifies how undefined code points are handled.

code point. A 1-byte code representing 1 of 256 potential characters.

color selection. The ability to specify a color other than black in order to print data in more than 1 color. Some printers support the selection of several colors, depending on the color of the ribbon installed in the printer. Other printers support the selection of black or "color of media," which can cause white lettering on a background that has been shaded black, for example.

command. A request from a terminal or a specification in a batch-processing job for the performance of an operation or the execution of a particular program.

committed copy counter. A device that keeps count of the copies of a page that have been committed to printing but have not been counted by the committed page counter.

committed page counter. A device that keeps count of the pages that have been committed to printing and have been removed from the page buffer. Synonym for *transfer counter*.

common-use forms. A set of paper sizes selected as the most commonly used throughout the world.

communication. See *data communication*.

communication-attached. In PSF, a device that is SNA-attached and that uses a communication controller. For example, a 3812 printer attached to a 3174 control unit that is attached to a 3175 Communication Controller can be considered a communication-attached printer. Contrast with *local attached*.

compatibility font. An AFP font designed to emulate the uniformly spaced fixed-pitch fonts used with line printers.

compatibility mode. A mode of operation for printing the output of 3800 Model 1 application programs on a 3800 page printer with little or no change to the application or the JCL.

completion code. An indication that reports the completion of a task.

compressed pattern storage (CPS). Storage that holds the extended (double-byte) fonts for the 3800 Model 6 and Model 8 printers.

computing system RPQ. A customer request for a price quotation on alterations or additions to the functional capabilities of a computing system, hardware product, or device. The RPQ can be used with programming RPQs to solve unique data processing problems. See also *programming RPQ* and *RPQ*.

concatenate. (1) To link together. (2) To join 2 character strings.

concatenated data set. In MVS, a group of logically connected data sets that are treated as a single data set for the duration of a job step. See also *data set*, *partitioned data set*, and *library*.

condition code. A code that reflects the result of an input/output, arithmetic, or logical operation.

conditional processing. A page definition function that enables input data records to partially control their own formatting.

console. A part of a computer used for communication between the operator or maintenance engineer and the computer. (A)

continuous forms. A series of connected forms that feed continuously through a printing device. The connection between the forms is perforated to enable a user to tear them apart. Before printing, the forms are stacked, folded along the perforations. Contrast with *cut-sheet paper*.

continuous-forms stacker. An output assembly on a continuous-forms printer that refolds and stacks the continuous forms after printing.

control character. A character that starts, changes, or stops any operation that affects recording, processing, transmitting, or interpreting data (such as carriage return, font change, and end of transmission).

control record. A subset of structured fields that can be mixed with line data records in a print data set.

copy group. (1) In a form definition, a complete set of printing controls specifying the format for a print job or part of a print job. A form definition may contain 1 or more copy groups, no 2 exactly alike. (2) One or more copies of a page of paper. The copies in a copy group can be made to differ by modifications such as text suppression, page position, forms flash, and overlays.

copy modification. The process of adding, deleting, or replacing data on selected copies of certain pages of a print job.

copy separation. The method for distinguishing among consecutive copies of a single data set. A continuous-forms stacker distinguishes copies by printing 1, 2, or 3 vertical bars on the left carrier strip between copies. A burster-trimmer-stacker, or a printer that uses cut-sheet paper, distinguishes copies by offset stacking.

core interchange font. A uniformly spaced, typographic font with specialized characters for different languages.

CPS. See *compressed pattern storage.*

current print position. The picture element that defines the character reference point or the upper left corner of an image.

cut-sheet paper. Paper that is cut into separate sheets before it is printed on. Contrast with *continuous-forms paper*.

D

DASD. See *direct-access storage device.*

data check. A synchronous indication of a condition caused by invalid data or incorrect positioning of data. You can suppress the reporting of some data checks.

data communication. Transfer of information between functional units by means of data transmission according to a protocol. (I)

data control block (DCB). A control block used by access method routines in storing and retrieving data.

data link. Connection between 2 terminals operating according to a link protocol. Contrast with *telecommunication line*.

data map. An internal object in a page definition that specifies fonts, page segments, fixed text, page size, and the placement and orientation of text. Synonymous with *page format*.

data map transmission subcase. An internal object that specifies the information for use in printing line data. Each data map of a page definition must contain 1 data-map transmission subcase.

data set. A named set of records stored and processed as a unit. Synonym for *file*.

data set header. A page in printed output that separates multiple data sets or multiple copies of a data set within a print job. See also *job header*.

DCB. See *data control block.*

DCF. See *Document Composition Facility.*

default. An attribute, value, or option that is assumed when none is explicitly specified. (I)

deferred-printing mode. A printing mode that spools output through JES to a data set instead of printing it immediately. Output is controlled by JCL statements. Contrast with *direct-printing mode*.

device manager. For channel-attached printers, the PSF subcomponent that manages the interface to the printer, builds CCWs, issues I/O requests, and processes I/O interrupts. For SNA-attached printers, the PSF subcomponent that loads the PPCC during device initialization, processes completed I/O, establishes and controls an SNA session through the PPCC, and supplies printer information in the CCB.

direct-access storage device (DASD). A computer storage device in which access time is, in effect, independent of the location of the data.

Direct Printer Services Subsystem (DPSS). The PSF subcomponent that serves as the interface between PSF and an application program when the job entry subsystem is not spooling jobs for a printer. DPSS attaches PSF as a subtask for use in direct-printing mode.

direct-printing mode. A printing mode that allows PSF exclusive use of a channel-attached printer. Output is printed immediately and is not spooled through JES. Contrast with *deferred-printing mode*.

disabled mechanism. PSF support that enables jobs to print with alternative options if the printer selected for the job does not support a requested option.

Distributed Print Function (DPF). A component of PSF/2 that you can install and use to print jobs from MVS, VSE, VM, or OS/400 systems. DPF receives host PSF output and resources for spooling and printing with PSF/2. DPF also stores PSF/MVS and PSF/VSE resources in the DPF resource library, so that the host system does not have to send PSF resources each time documents are spooled.

document. (1) A publication or other written material pertaining to a specific subject or related subjects. (2) In word processing, a collection of 1 or more lines of text that can be named and stored as a separate entity.

Document Composition Facility (DCF). An IBM licensed program that provides a text formatter called SCRIPT/VS. SCRIPT/VM can process files marked up with a unique set of controls and tags.

document environment group. An internal object that identifies the use of overlay and suppression and defines the placement of the page or pages on the form. A document environment group is a required part of every form definition.

double-byte coded font. A font in which each character is defined by 2 bytes, of which the first defines a coded font section, and the second defines a code point in that section. Double-byte coded fonts are needed for the support of languages requiring more than 256 graphic characters; 2 bytes are required to identify each graphic character. Kanji is printed by using a double-byte font. Contrast with *single-byte coded font*.

double-dot image. In Advanced Function Presentation, an image that is enlarged by doubling the pel pattern horizontally and vertically.

download. To transfer data from a processing unit to an attached device such as a microcomputer for processing.

DPF. See *Distributed Print Function.*

DPSS. See *Direct Printer Services Subsystem.*

drain. An operator action to halt the flow of jobs to a printer, usually to stop the printer or to change print options.

drained state. The condition in which job flow to the printer stops until the operator issues a command, such as START, directing the system to begin sending jobs to the printer, enabling it to resume printing.

DRM. Dynamic reconfiguration management.

duplex printing. Printing on both sides of a sheet of paper. Contrast with *simplex printing*. See also *normal duplex printing* and *tumble duplex printing*.

E

EBCDIC. Extended binary-coded decimal interchange code.

ECB. Event control block.

EID. Event ID.

electronic forms. A collection of constant data that is electronically composed in the host processor and can be merged with variable data on a page during printing.

electronic overlay. A collection of constant data, such as lines, shading, text, boxes, or logos, that is electronically composed in the host processor and stored in a library, and that can be merged with variable data during printing. Contrast with *page segment*. See also *page overlay* and *medium overlay*.

embedded text control. One or more bytes of control information contained in a set of text. The text control itself is not printed, but it controls certain operations on the text that follows it.

enabled. (1) Pertaining to a state of the processing unit that allows certain types of interruption. (2) A condition of the printer (physically selected) in which the printer is available to the host processor for normal work. Contrast with *disabled mechanism*.

end-user interface. A method by which a customer can obtain the services of a product, for example, coding samples, commands and command lists, control statements, display panels, job control language, listings of data streams, messages, program link-editing control statements, and text-formatting commands. Every product does not have an end-user interface; some products provide their services through programming interfaces, some provide services through a command line interface, and others provide their services only to

other products. Contrast with *programming interface for customers*.

equipment check. An asynchronous indication of a printer malfunction.

ERP. See *error recovery procedures*.

error recovery procedures (ERP). Procedures designed to help isolate and, where possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record information on machine malfunctions.

ESA. Enterprise System Architecture.

escape character. The control character X'2BD3' in a text-control sequence that indicates the beginning of the sequence and the end of any preceding text.

ESTAE. Extended specify task abnormal exit, a routine that receives control when a task terminates abnormally.

ETFF. Eject to front facing.

event control block (ECB). A control block used to represent the status of an event.

EWS. Early warning system.

exception. A condition that exists when the printer:

- Detects an invalid or unsupported command, order, control, or parameter value from the host
- Finds a condition of which the host system must be notified
- Detects a condition that requires the host system to resend data

exception highlighting. The markings placed on the printed page to indicate the location of an error in the data stream.

execution. The process of carrying out an instruction or instructions of a computer program by a computer. (I) (A)

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 8-bit characters.

ES of OS/2. Extended Services of Operating System/2.

extended specify task abnormal exit (ESTAE). A macroinstruction that enables a user to intercept an abend.

F

FCB. See *forms control buffer.*

fixed metrics. Measurement information, in specific units such as pels, inches, or centimeters, for individual graphic characters or collections of them. Contrast with *relative metrics*. See also *font metrics*. .

floating overlay. Synonym for *page overlay*.

font. (1) A family or assortment of characters of a given size and style; for example, 9-point Bodoni Modern. (A)

font character set. Synonym for *character set*.

font metrics. Measurement information that defines individual character values such as height, width, and space, as well as overall font values such as averages and maximums. Font metrics can be expressed in specified fixed units, such as pels, or in relative units that are independent of both the resolution and the size of the font. Also called *character metrics*.

font section. Synonym for *coded font section*.

form. A division of the physical medium; multiple forms can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing a form. An envelope is an example of a physical medium that has only 1 form. The IPDS architecture defines 4 types of form: cut-sheets, continuous forms, envelopes, and computer output on microfilm. Each type of form has a top edge, a front side, and a back side. Synonymous with *sheet*.

format. (1) A specified arrangement of such things as characters, fields, and lines, usually used for displays, printouts, or files. (2) To arrange such things as characters, fields, and lines. (3) To prepare a document for printing in a specified format.

formatted print records. Line data made up of records formatted for printing on line printers. PSF uses a page definition to print formatted records on page printers.

form definition. A resource that PSF uses to define the characteristics of a form; it specifies overlays to be used (if any), paper source (for cut-sheet printers), duplex printing, text suppression, the position of MO:DCA data on the form, and the number of copies and modifications of a page.

FORMDEF. A JCL parameter that specifies a form definition. See *form definition*.

forms control buffer (FCB). A buffer for controlling the vertical format of printed output. The forms control buffer is a line-printer control similar to the punched-paper, carriage-control tape used on IBM 1403 printers. On AFP page printers, the forms control buffer is replaced by the page definition. See *page definition*.

forms flash. In the 3800 printer, the function that prints user-prepared photographic images with variable MO:DCA data. When you specify use of a forms flash, the printer operator inserts a frame containing a photographic negative into the printer.

FSA. See *functional subsystem application.*

FSI. See *functional subsystem interface.*

FSS. See *functional subsystem.*

functional subsystem (FSS). The PSF address space that JES creates.

functional subsystem application (FSA). An area in the PSF functional subsystem that drives and manages a single printer.

functional subsystem interface (FSI). A set of services that enables communication between the JES address space or direct printer services subsystem and the PSF functional subsystem.

fuser counter. A device that counts the pages that have successfully passed the printer-defined jam-recovery point.

fuser station. In a printer, the assembly that bonds the toned image to the paper by heat and pressure.

G

generalized trace facility (GTF). An optional OS/VS service program that records important system events, such as supervisor calls and start I/O operations, for use in problem determination.

Global resource identifier (GRID). An 8-byte identifier that identifies a coded font resource. A GRID contains the following fields, in the order shown:

- Graphic character set global identifier (GCSGID) of a minimum set of graphic characters required for presentation. The GCSGID can be a character set that is associated with the code page, the font character set, or both.
- Code page global identifier (CPGID) of the associated code page.
- Font typeface global identifier (FGID) of the associated font character set.
- Font width, in 1440ths of an inch.

graphic character. See *character graphic*.

graphic character identifier. See *character identifier*.

GRID. See *global resource identifier*.

GTF. See *generalized trace facility.*

H

hard page segment. A page segment declared in the Map Page Segment structured field and loaded in the printer as a resource that can be reused during the job without being reloaded to the printer. Contrast with *soft page segment*.

hardcopy. (1) A copy of a display image that is generated on an output device such as a printer or plotter and that can be carried away. (T) (2) A printed copy of machine output in a visually readable form, for example, printed reports, listings, documents, and summaries.

hardware default font. The font that the printer uses if no other font is specified.

HCD. Hardware configuration definition.

hexadecimal. Pertaining to a numbering system with base of 16; valid numbers use the digits 0 through 9 and characters A through F, where A represents 10 and F represents 15.

host font. See *host resource*.

host processor. The processing unit to which the page printers are attached through a data-transfer interface.

host resource. A resource found in a system library, in a user library, or inline in the print data set.

host system. (1) A data processing system that prepares programs and operating environments for another computer or controller. (2) The data processing system to which a network is connected and with which the system can communicate.

I

identification label. In PSF, a protected set of resources (fonts, overlays, page segments) that are used to label PSF output for security purposes. Contrast with *security label*.

image. A pattern of toned and untoned pels that form a picture.

image block. A structure that contains the raster pattern and the instructions for placing the pattern on the page.

image cell. A portion of an image that saves storage by defining only part of a raster pattern. Each image cell must also contain information that defines the placement of its raster pattern within the complete image. An image cell can be replicated to fill a defined area.

image data. A pattern of bits, with values of 0 and 1, that defines the pels in an image. (A 1-bit is a toned pel.)

imbedded text control. See *embedded text control*.

IML. See *initial microprogram load.*

impact printer. A printer in which printing results from mechanical impacts. Contrast with *nonimpact printer*.

initial microprogram load (IML). (1) The action of loading microprograms into computer storage. (2) The action of loading licensed internal code from a diskette or some other medium to writable control storage.

initial program load (IPL). (1) The initialization procedure that causes an operating system to begin. (2) The process of loading system programs and preparing a system to run jobs.

initialize. (1) In programming languages, to give a value to a data object at the beginning of its lifetime. (I) (2) To set counters, switches, addresses, or the contents of storage to zero or other starting values at the beginning of, or at prescribed points in the operation of, a computer routine. (A) (3) To prepare for use; for example, to initialize a diskette. See also *initial program load*.

inline. The direction of successive characters in a line of text. Synonymous with *inline direction*.

inline direction. The direction of successive characters in a line of text.

inline resource. A resource contained in the print data set.

input/output (I/O). Pertaining to a device whose parts can perform an input process and an output process at the same time. (I)

installation exit. The subcomponent that calls exits; provides defaults for job header, trailer, and data-set header separator-page exits; and supports customer-written exits for logical records, SMF records, message processing, and resource management.

Intelligent Printer Data Stream (IPDS). (1) The data stream generated by PSF to send to an IPDS page printer. (2) An all-points-addressable data stream that enables users to position text, images, and graphics at any defined point on a printed page.

interface. A shared boundary. An interface can be a hardware component used to link 2 devices, or it can be a portion of storage or registers accessed by 2 or more computer programs.

interleaving. (1) The simultaneous accessing of 2 or more bytes or streams of data from distinct storage units. (2) The alternating of 2 or more operations or functions by means of the overlapped use of a computer family.

intermediate device. In the IPDS architecture, a device that operates on the data stream and is situated between a printer and a presentation services program in the host. Examples include devices that capture and cache resources and devices that spool the data stream.

internal object. Groups of structured fields that can be included as part of a resource or a print data set but that cannot be accessed separately.

intervention-required exception. An error that causes printing to stop until an operator performs a required action.

I/O. Input/output.

I/O error manager. The PSF subcomponent that analyzes I/O errors, determines the recovery action, and directs I/O error recovery and cleanup.

IOS. I/O supervisor.

IPDS. See *Intelligent printer data stream.*

IPL. Initial program load.

ISO forms. A set of paper sizes selected from those made standard by the International Organization for Standardization for use in data processing.

J

jam-recovery copy counter. A device that counts the copies of a page that have successfully passed the printer-defined jam-recovery point but have not been counted by the jam-recovery page counter.

jam-recovery page counter. A device that counts the pages that have successfully passed the printer-defined jam-recovery point. See also *fuser counter*.

JCL. See *job control language.*

JES. See *job entry subsystem.*

job control language (JCL). A language of control statements used to identify a computer job or describe its requirements to the operating system.

job entry subsystem (JES). An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system.

job header. A page in the printed output that indicates the beginning of a user job. A job can contain 1 or more data sets. See also *data set header*.

job separation. The ability to identify job boundaries by placing marks or sheets of paper between successive jobs.

job trailer. A page in the printed output that indicates the end of a user job.

K

kanji. Nonphonetic Chinese characters used in Japanese written language. In a font representing kanji characters, each character is represented by a double-byte font.

kerning. The printing of adjacent graphic characters so that they overlap on the left or right side.

L

landscape page presentation. The position of a printed sheet with its long edges as the top and bottom and its short edges as the sides. Contrast with *portrait page presentation*.

LASI. Library access system interface.

library. A file or a set of related files, for example, a page definition library containing 1 or more page definition files. A library often exists as a partitioned data set in MVS, and as a minidisk in VM.

library access system interface (LASI). The PSF subcomponent that gets resources from libraries and stores and obtains records in a message data set.

library member. Synonym for *resource object*.

ligature. A single character representing 2 or more input characters. For example, *ff* and *ffi* are character combinations that can be presented as ligatures.

line data. Data prepared for printing on a line printer such as a 3800 Model 1 Printing Subsystem. Line usually characterized by carriage control characters and table reference characters. Contrast with *MO:DCA-P data.*

line descriptor. Specifications describing how input data records are to be formatted into individual print lines. PSF interprets line descriptors when formatting printed output.

line merging. Printing 2 or more records of line data at the same location on the page. Line merging is used with line data to mix different fonts on the same line, to underscore or overstrike, and, on impact printers, to make print darker.

line printer. A device that prints a line of characters as a unit. (I) (A) Contrast with *page printer*.

lines per inch (lpi). (1) A unit of measurement used in specifying baseline placement. (2) A measure of the number of lines per vertical inch of paper.

local-attached. Pertaining to a SNA-attached device used with PSF that does not have a communications controller in its configuration. For example, a 3812 printer connected to a channel attached 3174 control unit that is defined to the host system through VTAM is considered a local-attached printer. Contrast with *communication-attached*.

local identifier. A 1-byte identifier assigned to parts of the data stream to facilitate PSF processing. For example, the Map Coded Font structured field assigns each coded font a local identifier. When a coded font is required for processing, this identifier is specified in the Set Coded Font Local text control.

logical page. A presentation space. One or more object areas or data blocks can be mapped to a logical page. A logical page is rectangular and has specifiable characteristics such as size, shape, orientation, and offset. Orientation and offset are specified relative to a coordinate system for the medium.

logical page origin. (1) The point on the logical page from which the positions of images, graphics, page overlays, and text with 0-degree inline direction are measured. (2) The point on the logical page represented by Xp=0, Yp=0 in the Xp coordinate system.

logical unit (LU; L-unit). (1) In SNA, a port through which an end user accesses the SNA network to communicate with another end user, and through which the end user accesses the functions provided by system services control points (SSCP). A logical unit can support at least 2 sessions, 1 with an SSCP and one with another logical unit, and may be able to support many sessions with other logical units. (2) A unit of linear measurement. For example, in Mixed Object

Document Content Architecture (MO:DCA) the following logical units are used:

1 L-unit = 1/1440 inch
(unit base = 10 inches,
units per unit base=14 400)
1 L-unit = 1/240 inch
(unit base = 10 inches,
units per unit base = 2400)

logon mode. In VTAM, a subset of session parameters specified in a logon-mode table for communication with a logical unit.

logon-mode table. In VTAM, a table of macro-generated constants that associate a logon-mode name with a set of session parameters. Each logon mode is identified by a logon-mode name.

lpi. Lines per inch.

LU (logical unit) base. The linear measurement base. The value defines, for any object, the meaning of the L-unit values. See also *logical unit*.

L-unit. See *logical unit*.

LU type 1. An SNA logical unit type that provides a communication protocol among host application programs and terminals. Some printers also use this protocol to communicate with host application programs.

LU type 6.2. An SNA logical unit type that converges functions from existing LU types to provide a single, interchangeable communication protocol. See also *Page Printer Communication Component*.

M

macro. Synonym for *macroinstruction*.

macroinstruction. An instruction that causes the execution of a predefined sequence of instructions.

magnetic ink character recognition (MICR). Recognition of characters printed with ink that contains particles of a magnetic material.

mandatory print labeling (MPL). A class, defined to the Resource Access Control Facility (RACF), that causes PSF to automatically label separator pages and data pages and to enforce the user-printable area.

marking. Using the APSRMARK utility to update the BEGIN structured field in a resource to identify it as PUBLIC or PRIVATE. A PUBLIC mark instructs PSF to use a resident version of the resource if 1 is available; PRIVATE signals PSF to use the host version of the resource.

media origin. The first hardware-addressable point on the physical page; the point from which the logical page origin is positioned by the medium map. This point is represented by Xm=0, Ym=0 in the Xm, Ym coordinate system. The media origin is defined with respect to the top edge of the media.

medium map. An internal object in a form definition that controls the modifications to a form, page placement, and overlays. Synonymous with *copy group*.

medium overlay. An electronic overlay invoked by the medium map of a form definition for printing at a fixed position on the form. Contrast with *page overlay*.

member ID. The member name of a resource object, minus the 2-character prefix. For example, BITR is the member ID of the font whose member name is X1BITR.

member name. The name under which a file is stored in a library. For example, X1BITR is the member name of a font in the font library.

message data set. (1) A data set on disk storage that contains queues of messages awaiting transmission to particular terminal operators or to the host system. (2) In PSF, a virtual data set built in memory by the LASI component to store error messages for printing at the end of the document.

MICR. See *magnetic ink character recognition.*

metrics. Synonym for *font metrics*.

Microfilm device. An output device that presents a hardcopy on microfilm.

microfilm setup resource. A setup file that contains information used to present AFP data on microfilm.

Mixed Object Document Content Architecture. A strategic, architected, device-independent data stream for interchanging documents.

mixed-pitch font. A font that simulates a typographic font. The characters are in a limited set of pitches, for example, 10 pitch, 12 pitch, and 15 pitch.

MO:DCA. See *Mixed Object Document Content Architecture.*

MO:DCA data. Print data that has been composed into pages. Text-formatting programs such as DCF can produce composed text data consisting entirely of structured fields.

MO:DCA data definition. A resource containing a set of formatting controls for printing logical pages of data. This resource includes controls for the number of lines per printed sheet, font selection, print direction, and

mapping individual fields in the data to position on the printed sheets.

MO:DCA data page. A page of print data consisting entirely of structured fields.

MO:DCA print data set. A print data set consisting entirely of structured fields.

MO:DCA-P. Mixed Object Document Content Architecture for Presentation.

modified-default form definition. A form definition that was the default specified in the PRINTDEV statement of the PSF startup procedure and modified by the groupvalue parameter of the COPIES parameter or by the FLASH parameter in the JCL statement.

modified-default page definition. A page definition that was the default specified in the PRINTDEV statement of the PSF startup procedure and modified by a font list specified in one of the following places:

- The CHARS parameter from the user's JCL or PRINTDEV statement
- The UCS parameter from the user's JCL
- The JES default font in the current printer setup

MOF. Metric-only font. See *4028 Font Metrics.*

monospaced font. A font in which the graphics characters have a uniform character increment. Synonymous with *uniformly spaced font*. Contrast with *proportionally spaced font*.

MPL. See *mandatory print labeling.*

multiple-entry font. A font with multiple entries in the Map Coded Font structured field. The only fonts that have multiple entries are double-byte fonts that are defined dynamically. (The MCF points directly to a set of code page and font character set pairs.)

multiple-up. The printing of more than 1 page of application data on a single surface of a sheet of paper.

Multiple Virtual Storage (MVS). An IBM operating system consisting of MVS/System Product Version 1 and the MVS/370 Data Facility Product, operating on a System/390 processor.

multiplexer. (1) A device that combines several input signals into a single output signal in such a manner that each of the input signals can be recovered. (T) (2) A device capable of interleaving events of 2 or more activities, or capable of distributing events of an interleaved sequence to the respective activities. (A)

multiplexer channel. A channel designed to operate with a number of I/O devices simultaneously. Several I/O devices can transfer records at the same time by interleaving items of data.

multiplexing. In data transmission, a function that permits 2 or more data sources to share a common transmission medium so that each source has its own channel. (I) (A)

multipoint network. Three or more devices sharing a transmission line at the same time.

MVS. See *Multiple Virtual Storage.*

MVSCP. MVS Configuration Program.

MVS/ESA. Multiple Virtual Storage/Enterprise System Architecture.

N

NACK. Negative acknowledgment.

NCP. See *Network Control Program.*

NCP generation. The process performed in the host processor of assembling and link-editing a macroinstruction program to produce a Network Control Program.

nested resource. A resource mapped in an overlay.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

nonimpact printer. A device in which printing does not result from mechanical impacts, for example, thermal printers, electrostatic printers, and photographic printers. (I) (A) Contrast with *impact printer*.

nonprocess runout (NPRO). An operation that moves paper through the paper path without printing.

nonswitched line. A telecommunication line on which communications do not have to be established by dialing. Contrast with *switched line*.

normal duplex printing. Printing on both sides of the paper to enable sheets to be bound on the long edge of the paper. Contrast with *simplex printing*. See also *tumble duplex printing*.

notify subtask (NST). (1) A PSF subcomponent that returns processed data sets to JES or the Direct Printer Services Subsystem and performs checkpoint processing on data sets as they are printed. (2) An external trace that contains information pertaining to the releasing and checkpointing of data sets by PSF.

NPRO. See *nonprocess runout.*

NST. Notify subtask.

null name. A token name with X'FFFF' in the first 2 bytes.

null value. A parameter for which no value is specified.

numeric data. (1) Data represented by numerals. (2) Data in the form of numerals and special characters, for example, a date represented as 81/01/01. See also *character data*.

N_UP printing. In basic N_UP printing, the dividing of a side of a sheet into a fixed number of equal-size partitions. For example, N_UP 4 divides each side of the sheet into 4 equal partitions. In enhanced N_UP printing, the sheet can be divided into 8 partitions, which can be arranged anywhere on a sheet.

O

object. A resource or a sequence of structured fields in a larger entity, such as a page segment or a composed page.

object container. A set of structured fields used to carry object data for a variety of objects.

offset stacking. A function that offsets printed output pages for easy separation of print jobs.

OGL/370. See *Overlay Generation Language/370.*

option. (1) A specification in a statement that can be used to influence the execution of the statement. (2) A choice offered from a list of possibilities.

orientation. The number of degrees an object is rotated relative to a reference, for example, the orientation of an overlay relative to the logical page origin. Orientation usually applies to blocks of information. Character rotation applies to individual characters. See also *text orientation*.

origin. A pel position from which the placement and orientation of text, images, and page segments are specified. For example, pages, overlays, and page segments have origins.

outline font. A font technology in which the graphic character shapes are represented in digital form by a series of mathematical expressions that define the outer edges of the strokes. The resulting graphic character shapes can be either solid or hollow. Outline fonts can be scaled (sized) to any size. The IBM outline font character sets have a prefix of CZ. Contrast with *raster font*.

overlay. See *electronic overlay*.

Overlay Generation Language/370 (OGL/370). An IBM licensed program used to create overlays.

P

page. A collection of data that can be printed on a physical sheet of paper.

page environment. The size of the page or overlay control data to be used in MO:DCA data text blocks, and the page segments or fonts to be used.

page format. A subset of a page definition, containing controls governing the arrangement of text on a page.

page mode. The mode of operation in which a page printer can accept a page of data from a host processor for printing on an all-points-addressable output medium. Data can consist of pages composed of text, images, overlays, or page segments.

page origin. See *logical page origin*.

page overlay. An electronic overlay that can be invoked for printing, and positioned at any point on the page, by an Invoke Page Overlay structured field in the print data. Contrast with *medium overlay*.

page position. A control in the copy group that assigns the top left boundary point of the logical page on a sheet for a data set. The page position is determined from the media origin.

page printer. Any of a class of printers that accepts MO:DCA-P pages. Contrast with *line printer*.

Page Printer Communication Component (PPCC). The access method that provides the SNA communication interface between printers and PSF.

Page Printer Formatting Aid (PPFA). An IBM licensed program you can use to create and store form definitions and page definitions. You can use PPFA on the following operating platforms: OS/400, OS/390, MVS, VM, VSE, and AIX. PSF/2 can use the form definitions and page definitions created by PPFA, but the PPFA product does not run on the OS/2 platform.

page segment. A resource containing MO:DCA data and images, prepared before formatting and included as part of the input for a print job.

PAGEDEF. A JCL parameter that specifies a page definition. See *page definition*.

paging. (1) The transfer of pages between real storage and auxiliary storage. (I) (A) (2) In

System/390 virtual storage systems, the process of transferring pages between real storage and external page storage.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

partial page. A page that does not contain all the intended data. Partial pages can be printed after an error is sensed.

partition. In basic N_UP printing, the division of the medium presentation space into a specified number of equal-sized areas in a manner depending on the current physical medium.

partitioned data set (PDS). A data set in direct-access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Contrast with *sequential data set*.

pass-through. To gain access to another network.

pattern storage (PST). An area of storage that holds the raster patterns for fonts and images.

PDS. See *partitioned data set.*.

pel. See *picture element.*

pending page queue (PPQ). In PSF, a list of pages that have been processed by PSF but have not been released from JES.

Personal System/2. IBM's personal computers that use an 80286, 80386, or 80486 processor.

physical medium. A physical entity on which information is presented. Examples of physical media are display screens, paper, foils, microfilm, and labels.

physical unit (PU). The part of a communication controller or cluster controller that manages the resources attached to the node.

picture element. An element of a raster pattern about which a toned area on the photoconductor might appear. See also *raster pattern*. Synonym for *pel*.

pipeline. (1) A serial arrangement of processors or a serial arrangement of registers within a processor. Each processor or register performs part of a task and passes the results to the next processor; several parts of different tasks can be performed at the same time. (2) The hardware path between the channel station or received-page station and the stacker.

pitch. A unit of measurement for the width of a printed character, reflecting the number of times a graphic character can be set in 1 linear inch; for example, 10-pitch has 10 graphic characters per inch. Contrast with *point*.

PLR. See *priority-level resource.*.

point. A unit of measurement about 1/72 of an inch; used in measuring the height of a font. Contrast with *pitch*.

point size. The height of a font in points.

point-to-point network. An arrangement in which 2 devices share the same transmission line at the same time.

POR. Power-on reset. See *initial microprogram load*.

portrait page presentation. The position of a printed sheet that has its short edges as the top and bottom and its long edges as the sides. Contrast with *landscape page presentation*.

power-on reset (POR). See *initial microprogram load*.

PPCC. See *Page Printer Communication Component.*.

PPFA. See *Page Printer Formatting Aid.*

PPQ. See *pending page queue.*

PPWTR. Page printing writer.

preprinted form. A sheet of paper containing a preprinted design of constant data into which variable data can be printed. See also *electronic overlay* and *forms flash*.

presentation device. A device that produces character shapes, graphics pictures, images, or bar code symbols on a physical medium. Examples of physical media are display screens, paper, foils, microfilm, and labels.

presentation text. Synonym for *MO:DCA data*.

print data set. A named set of records stored and processed as a unit. See *data set.*

print data stream. The data stream created by PSF and transmitted to the printer.

print file. A file created by an application program that contains the actual information to be printed and some of the data that controls the format of the printing. Print files can contain MO:DCA-P data, line data, or a combination of MO:DCA-P and line data.

print direction. (1) The direction in which characters are added to a line. (2) In PSF, the specification of inline direction for the printing of text.

print job. The data that a user submits to PSF for printing.

print labeling. A controlled method of placing identification labels on each page of output printed by PSF.

print position. Any location on a medium where a character can be printed.

print quality. (1) The measure of printed output against existing standards and in comparison with jobs printed previously. (2) The ability of some page printers to print data at more than 1 level of print quality, such as “draft” and “near-letter” quality.

Print Services Facility (PSF). An IBM licensed program that produces printer commands from the data sent to it. PSF programs run on the AIX, MVS, VM, VSE, OS/2, OS/390, and OS/400 operating platforms.

printable area. The area on a sheet of medium on which print can be placed.

printer. A presentation device that produces hardcopy output. See *presentation device*.

priority level resource (PLR). A resource that is in the RLST chain when PSF returns from Resource Exit (APSUX07) and that has the RLSTPLR bit turned on. When PSF is deleting resources at the end of a data set, PSF deletes all non-PLR resources before deleting any PLR resources. A frequently used resource can be marked as PLR with the Resource Exit, thus improving its chances of being retained after the data set is finished.

processor. In a computer, a functional unit that interprets and executes instructions. (I) (A)

program status word. An area in storage used to indicate the order in which instructions are executed, and to hold and indicate the status of the computer system.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

programming interface for customers. Any product method that enables a customer-written program to obtain the services of the product, for example, CSECT names, data areas or control blocks, data sets or files, exits, macros, parameter lists, and programming languages. Not all products have programming interfaces for customers; some products provide their services through end-user interfaces, and others provide their services only to other products. Contrast with *end-user interface*.

programming request for price quotation. A customer request for a price quotation on alterations or additions to the functional capabilities of system control programming or licensed programs. The RPQ can be used in conjunction with computing system RPQs to solve unique problems in data processing.

proportionally spaced font. A typographic font, or in some usages a mixed-pitch font. See *typographic fonts* and *mixed-pitch fonts*.

protocol. The meanings of and the sequencing rules for requests and responses by which network-addressable units (PU, LU, SSCP, and VTAM programs) in a communication network coordinate and control data transfer operations and other operations.

PRPQ. See *programming request for price quotation.*

PSF. See *Print Services Facility.*

PSF Direct. A function of PSF/2 or PSF for AIX that enables another PSF program (PSF/VM, PSF/MVS, PSF/VSE, or PSF/400) to print remotely, using the SNA LU 6.2 protocol, on printers supported by PSF/2 or PSF for AIX. The PSF program sends the print data stream directly to the PSF/2 or PSF for AIX printer, bypassing the OS/2 or AIX spool. The operator of the originating system controls printing on the PSF/2 or PSF for AIX printers as though the printers were attached to the originating system.

PST. Pattern storage.

PSW. Program status word.

PS/2. The IBM Personal System/2 operating system.

PTF. See *program temporary fix.*

PU. Physical unit.

R

RACF. See *Resource Access Control Facility.*

raster font. A font technology in which the graphic characters are defined directly by the raster bit map. Contrast with *outline font*.

raster overlay. See *raster pattern overlay.*

raster pattern. A series of picture elements (pels) in scan lines to form an image. See also *page segment.*

raster pattern overlay. An overlay loaded in the printer as a raster pattern rather than as a sequence of printer commands. Contrast with *coded overlay*.

raster pattern storage (RPS). An area of storage that holds raster patterns for fonts and images.

RCB. See *resource control block.*

RDW. See *record descriptor word.*

real storage. The main storage in a virtual storage system. Physically, real storage and main storage are identical. Conceptually, however, real storage represents only part of the range of addresses available to the user of a virtual storage system. Formerly, the total range of addresses available to the user was provided by the main storage. (I) (A)

reasonable resource loaded value. A value used by PSF to keep the number of resources loaded at a manageable level. At the end of a data set, PSF deletes resources of a particular resource type until this value is reached.

received page counter. A device in a printer that counts the pages that have been successfully received and syntax-checked. Synonymous with *channel counter*.

record descriptor word. Data preceding a variable record that specifies the length of the entire record, including the RDW.

record interface. Coordinates the transmission of printer resources needed to print a document.

recovery termination manager (RTM). A program that handles all normal and abnormal terminations of tasks by passing control to a recovery routine associated with the terminated function.

region size. The amount of main storage available for executing programs.

relative metrics. Measurement information that is defined in relation to some other units. Relative values are expressed as fractional parts of a unit square design space (Em-square), whose sides correspond to the vertical size of the font. This enables the same value to be used for different point sizes and different raster-pattern resolutions. The use of relative metrics requires the definition of the units of measure for the Em-square, the point size of the font, and the pel resolution of the raster. Contrast with *fixed metrics*. See also *font metrics* and *bounded-box relative metrics*.

reload function. The Resource Exit can request that a resource be “reloaded.” PSF will not use an existing version of the resource but will load the resource from a host library.

Remote PrintManager Version 2.0. A program that runs on a personal computer and stores MVS and VSE printer resources (fonts, overlays, and page segments) in the personal computer's fixed disk for use by an attached printer. As of September, 1994, RPM 2.0 is no longer marketed, because it has been replaced by the PSF Direct function of PSF/2 and PSF for AIX. See *PSF Direct*.

Remote PrintManager Version 3.0. A program that runs on a personal computer connected to a Local Area Network (LAN) and that connects printers to the LAN. ASCII files created on the personal computer and files sent from an MVS, VM, VSE, or OS/400 host are placed on the remote print spool by RPM, which then acts as a print server and schedules the converted ASCII files and host files for printing. RPM 3.0 also stores MVS and VSE resource files in its resource library, thus eliminating the need to download the resources each time a file using these stored resources is printed. RPM 3.0 is no longer sold or serviced, because it has been replaced by the Distributed Print Facility (DPF) of PSF/2.

repositioning. A process in which Print Services Facility, following an indication from the printer or from JES of a potentially recoverable error, locates the correct spool record for recomposing 1 or more pages for printing.

request for price quotation. See *computing system RPQ*.

request unit (RU). In SNA, a message unit that contains control information such as a request code, or function management (FM) headers and end-user data, or both. Synonymous with *request*.

resident resource. A resource that is stored in a printer (fonts or symbol sets) or in an intermediate device (fonts, page segments, and overlays). Resident resources are identified by the APSRMARK utility of PSF or by the resident font resource table APSRFTBL.

resolution. In computer graphics, a measure of the sharpness of an image, expressed as the number of lines and columns on the display screen or the number of pels per unit of linear measure.

resource. (1) A collection of printing instructions used by PSF in addition to the print data set to produce printed output. PSF resources include coded fonts, font character sets, code pages, page segments, overlays, form definitions, and page definitions. (2) Any source of aid used for performing a task, such as disk storage space, computer processing time, and communication lines.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging any detected unauthorized attempts to enter the system, and logging any detected accesses to protected resources.

resource-constrained condition. A condition in which the printer does not have enough storage for the resources required to print the current page.

resource manager. Any control-program function responsible for allocating a resource. In PSF, the resource manager maintains the resource control structure used to manage resources, directs the retrieval of resources from libraries, deletes resources from the printer, calls for Resource Exit functions, and includes the APSRMARK utility that enables programmers to mark resources as printer-resident.

resource name. The name under which a resource object is stored, the first 2 characters of which indicate the resource type:

X0-XG,XZ	Coded font
T1	Code page
C0-CG,CZ	Font character set
S1	Page segment
F1	Form definition
P1	Page definition
O1	Overlay
H1	Recommended for microfilm

resource object. A member in a resource library. Synonym for *library member*.

rotation. Synonym for *character rotation*. See also *orientation*.

routine. A program or sequence of instructions called by a program that may have some general or frequent use. (I) (A)

RPM. See *Remote PrintManager Versions 2.0 or 3.0.*

RPQ. Request for price quotation. See *computing system RPQ*.

RPS. See *raster pattern storage.*

RRLV. See *reasonable resource loaded value.*

RTM. See *recovery termination manager.*

RU. Request unit.

rule. A solid or patterned line of any weight, extending horizontally or vertically across a column, a row, or a page.

S

SDLC. See *Synchronous Data Link Control.*

security definition. In MVS, a member containing the definitions for one identification label. These definitions include instructions for the overlay name and the size and origin of paper to be used.

security definitions library. In MVS, a partitioned data set, or a series of concatenated partitioned data sets, containing the security definitions for an entire system.

security label. In a “trusted” computing base, a security label used to maintain multiple levels of security on a system. This label is a combination of a security class and a security level. Contrast with *identification label*.

security overlay. An overlay, such as 1 created by using Overlay Generation Language/370, that resides in a secure library and is used to place security resources on a page.

segment. Synonym for *page segment*.

sense data. (1) Data describing an I/O error. Sense data is presented to a host system in response to a Sense I/O command. (2) In SNA, explanatory data sent with a negative response.

sequence number. A 2-byte field in the structured field introducer that identifies the position of the structured field in the data set.

sequential data set. In MVS, a data set whose records are organized on the basis of their physical positions, such as on magnetic tape. Contrast with *partitioned data set*.

sheet. A division of the physical medium on which data is presented. The IPDS architecture defines 4 types of sheet: cut-sheet forms, continuous forms, envelopes, and computer output on microfilm. Each sheet has a front side and a back side. Some types of media consist of multiple sheets; for example, a roll of continuous forms can be divided at the perforations into rectangular sheets. Each sheet usually has carrier or tractor-feed strips, also. Microfilm is another example of a medium comprising multiple sheets, whereas an envelope has only 1 sheet. Synonymous with *form*.

shift-out, shift-in (SOSI). Special EBCDIC control codes in the data stream to indicate switches between double-byte fonts and single-byte fonts.

simplex printing. Printing on only 1 side of the paper. Contrast with *duplex printing*.

single-byte coded font. A font in which the characters are defined by a 1-byte code point. A single-byte coded font contains only 1 coded font section. Contrast with *double-byte coded font*.

single-entry font. Fonts with a single entry in the Map Coded Font structured field, including all single-byte fonts and those double-byte fonts defined as coded fonts.

skip. (1) To ignore 1 or more instructions in a sequence of instructions. (2) A move of the current print position to another location.

SMF. See *system management facilities.*

SMP/E. See *System Modification Program/Extended.*

SNA. See *Systems Network Architecture.*

SNA-attached. In PSF, pertaining to a device that is linked to the host system through VTAM and that uses an SNA protocol to transfer data. The device does not need to be physically connected to the host; some printers are attached to a control unit, a communication controller, or both, and they can transfer data over telecommunication lines. For example, a 3820 that is attached to a communication controller and that uses the LU 6.2 communication protocol to transfer data to a communication controller is considered an SNA-attached printer. Contrast with *channel attached*.

soft page segment. A resource that is not declared in the Map Page Segment structured field but is sent to the printer inline with data. Contrast with *hard page segment*.

SOSI. See *shift-out, shift-in.*

spot carbon. Paper from which carbon is omitted in certain areas to suppress the printing of data on specific copies.

SRM. System Resource Manager.

SSCP. See *system services control point.*

SSI. See *functional subsystem interface.*

stacked-page counter. A device that counts the pages that have been successfully stacked.

startup procedure. A procedure used to start PSF and to specify PSF initialization parameters and libraries that contain system resources.

station ID. For the 3800 printer, a 2-byte pipeline counter that is incremented when the last copy of a page reaches the point in the hardware represented by that counter.

storage. (1) A unit into which recorded text can be entered, in which it can be retained and processed, and from which it can be retrieved. (T) (2) The action of placing data in a storage device. (I) (3) A storage device. (A)

structured field. A self-identifying, variable-length, bounded record that can have a content portion that provides control information, data, or both.

structured field introducer. The first 8 bytes of a structured field, which indicate its length, type, and number.

subgroup. A set of modifications in a copy group that applies to a certain number of copies of a form. A copy group can contain more than 1 subgroup.

subpage. A grouping of line data records. A line data record is identified as belonging to a particular subpage with the subpage identifier byte in the Line Data Descriptor (LND) structured field. Conditional processing can be used with a page definition to select a new data map and/or medium map to take effect before or after a current subpage is printed.

supervisor call (SVC). An instruction that interrupts a program being run and passes control to the supervisor to perform a service that the instruction specifies.

suppression. See *text suppression*.

suppression local ID. A value assigned in the Map Suppression coded field to a suppression named in a Line Descriptor structured field. This value is contained in the Begin and End Suppression text controls.

suppression number. An identification number from 1 to 127 generated by the Map Suppression structured field and assigned to text designated for suppression in the data map transmission subcase.

swapping. A process by which the contents of an area of real storage are interchanged with the contents of an area in auxiliary storage.

switched line. A telecommunication line on which communications are established by dialing. Contrast with *nonswitched line*.

switched major node. In VTAM, a major node whose minor nodes are physical and logical units attached by switched SDLC links.

SVC. See *supervisor call*.

symbol set. A type of font that resides in a printer but has fewer attributes than can be specified for resident coded fonts.

synchronous. (1) Pertaining to 2 or more processes that depend on the occurrences of a specific event, such as common timing signals. (I) (A) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous information transfer over a data link connection.

SYSGEN. System generation. (A)

SYSIN. System input stream.

SYSOUT. System output stream.

system input stream (SYSIN). A data definition (DD) statement used to begin an in-stream data set.

system management facilities (SMF). An optional control program that provides the means for gathering and recording information that can be used to evaluate system use.

System Modification Program/Extended (SMP/E). An IBM licensed program used to install software and software changes.

system output stream (SYSOUT). A data definition (DD) statement used to identify a data set as a system output data set.

Systems Network Architecture (SNA). In IBM networks, the description of the layered logical structure, formats, protocols, and operational sequences that are used for transmitting information units through networks, as well as for controlling the configuration and operation of networks.

systems services control point (SSCP). In SNA, a focal point with an SNA network for managing the configuration, coordinating network-operator and problem-determination requests, and providing directory support or other session services for end users of the network.

T

table reference characters (TRC). An optional control character in an input record that identifies the font with which the record is to be printed. The table reference character corresponds to a font number defined in a page definition font list or to the order of font names listed in the CHARS parameter in the JCL.

telecommunication line. The part of a data circuit, external to the equipment, that connects to a data-switching exchange. Contrast with *data link*.

terminate. (1) In SNA products, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure for ending 1 or more designated LU-LU sessions. (2) To stop the operation of a system or device. (3) To stop execution of a program.

text. A graphic representation of information on an output medium. Text can consist of alphanumeric characters and symbols arranged in paragraphs, tables, columns, or other shapes.

text control. Structured-field data that controls the format, placement, and appearance of text.

text control sequence. A text control and its associated data.

text orientation. A description of the appearance of text as a combination of inline and baseline directions and character rotation.

text suppression. The intentional omission of portions of text from copy groups specified in a parameter in the form definition.

throughput. (1) A measure of the amount of work performed by a computer system over a period of time, for example, the number of jobs per day. (I) (A) (2) In data communication, the total traffic between stations per unit of time.

token name. An 8-byte name that can be given to all internal objects and resources.

token ring. A network configuration in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

trace. A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

transaction ID. An identifier assigned by the Document Processor component, associated with an I/O buffer, and representing a form of resource.

transfer counter. A device that counts the pages that have been committed to printing and have been removed from the page buffer. Synonym for *committed page counter*.

transmission. (1) The sending of data from 1 place for reception at another. (A) (2) The sending of a print data set to a printer.

TRC. See *table reference character*.

tumble duplex printing. Duplex printing for sheets that are to be bound on the short edge of the paper regardless of whether the printing is portrait or landscape. Contrast with *normal duplex printing*.

2-channel switch. A hardware feature that enables you to attach an I/O device to 2 channels. Adding a dynamic switch enables both interfaces at the same time; channel selection is then determined by programming.

type size. (1) A measurement in pitch or points of the height and width of a graphic character in a font. (2) One of the many attributes of a font; others are weight and width.

typeface. A collection of fonts all having the same style, weight, and width. Each font differs from the others by point size or type family.

typographic font. A font in which the distance between characters varies. The distance from 1 character to another is adjusted to improve the visual flow of text by eliminating excess space.

U

UCS. See *universal character set.*

unbounded-box font. A font designed to use unbounded character boxes. Contrast with *bounded-box font*.

unbounded character box. A character box design that requires the character position information to be included (as untoned pels) in the box. Contrast with *bounded character box*.

unformatted print records. Line data comprised of fields of data that have not been formatted into print lines. PSF uses a page definition to format these records for printing on page printers.

uniformly spaced font. A font in which every character has the same character increment. Contrast with *proportionally spaced font*.

units of measure. Counting methods that are either specific or relative. See also *font metrics* and *bounded-box relative metrics*.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays. Synonymous with *font*.

UPA. See *user-printable area.*

user library. A private print-resource library owned by an individual user, accessed only when the owner specifies the name in the JCL USERLIB parameter.

user-printable area (UPA). The area within the valid printable area (VPA) in which user-generated data can print without causing an exception condition. See also *valid printable area*.

utility program. (1) A computer program in general support of computer processes, for example, a diagnostic program, a trace program, or a sort program. (T) Synonymous with *service program*. (2) A program designed to perform a routine task such as copying data from 1 storage device to another. (A)

V

valid printable area (VPA). The intersection of the partition for basic N_UP with the current logical page or the current overlay. See also *user-printable area*.

value. A quantity assigned to a constant, a variable, a parameter, or a symbol in a command.

virtual storage. (1) The storage space that the user of a computer system can regard as addressable main storage in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (I) (A) (2) Addressable space from which the instructions and the data are mapped into the processor storage locations.

Virtual Telecommunications Access Method (VTAM). A set of programs that maintains control of the communication between terminals and application programs running under the DOS/VS, OS/VS1, and OS/VS2 operating systems.

VPA. See *valid printable area.*

VTAM. See *Virtual Telecommunications Access Method.*

W

WCS. See *writable control storage.*

writable control storage (WCS). Printer storage for entering, holding, and retrieving data. Writable control storage contains licensed internal-code instructions and other control information, such as the print buffer.

write-to-operator (WTO). An optional user-coded service that enables the writing of a message to inform the system console operator of errors and system conditions that may need correcting.

WTO. See *write-to-operator.*

X

X-axis. In printing, an axis perpendicular to the direction in which the paper moves through the printer. See also *Y-axis*.

X-extent. A measurement along the X-axis.

Xm, Ym coordinate system. The coordinate system for a medium.

Xp, Yp coordinate system. The coordinate system for a logical page.

Y

Y-axis. In printing, an axis parallel to the direction in which the paper moves through the printer. See also *X-axis*.

Y-extent. A measurement along the Y-axis.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

Readers' Comments — We'd Like to Hear from You

Print Services Facility/MVS
Application Programming Guide
Version 2 Release 2.0

Publication No. S544-3673-03

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

Note: IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

	Yes	No
• Does the publication meet your needs?	____	____
• Did you find the information:		
Accurate?	____	____
Easy to read and understand?	____	____
Easy to retrieve?	____	____
Organized for convenient use?	____	____
Legible?	____	____
Complete?	____	____
Well illustrated?	____	____
Written for your technical level?	____	____
• Do you use this publication:		
As an introduction to the subject?	____	____
As a reference manual?	____	____
As an instructor in class?	____	____
As a student in class?	____	____
• What is your occupation?		

_____

Thank you for your input and cooperation.

Note: You may either send your comments by fax to 1-800-524-1519, or mail your comments. If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

Comments:

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
S544-3673-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
The IBM Printing Systems Company
Department H7FE Building 003G
P O Box 1900
BOULDER CO 80301-9817

Fold and Tape | **Please do not staple** | Fold and Tape

S544-3673-03

Cut or Fold
Along Line

File Number: S370-40
Program Number: 5695-040

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

S544-3673-03