Advanced Function Printing: A tutorial

by R. K. deBry
B. G. Platte

Advanced Function Printing (AFP) is an IBM product for printing mixed text, image, and graphics in a system-printing environment. Described is the AFP printing model. We demonstrate the way in which this model is used for existing printing applications, enhanced line printing, and full advanced-function printing.

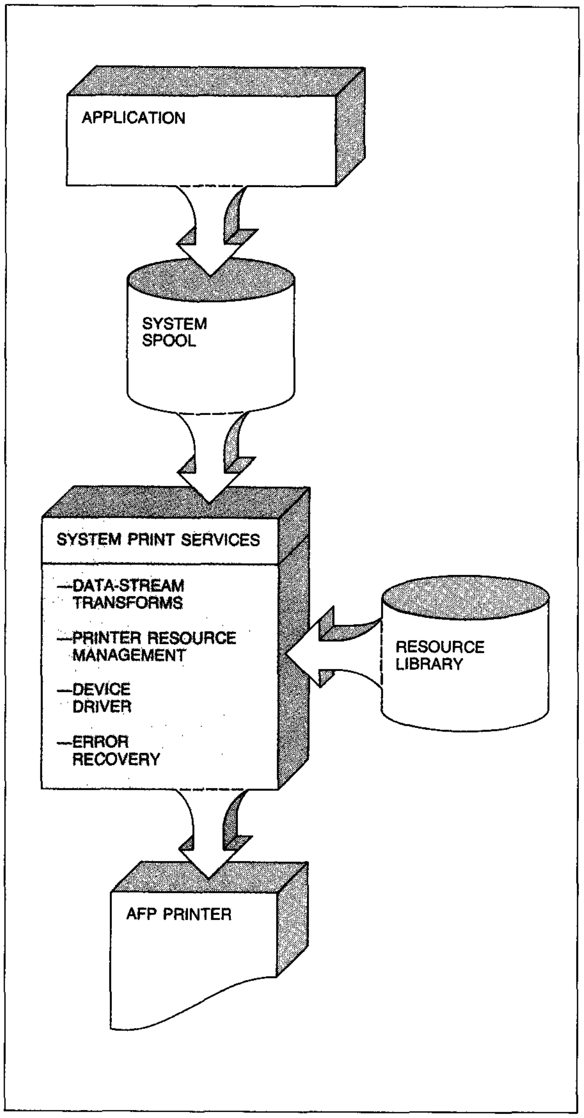
With the beginning of civilization and the development of agriculture and the domestication of animals, people experienced a need to record what they thought, owned, bought, and sold. The earliest examples of written records go back to 3000 B.C., when Mesopotamian scribes stippled the details of business transactions on tablets of soft clay. The first paper used for writing was *papyrus*, which was made of strips of a tall marsh plant glued together into sheets. Papyrus was developed by Egyptians living in the Nile River delta around 600 B.C. Although papyrus was not an easy medium to make and use, it was easier to use than carving in stone. The later invention of parchment meant that people could be much more expressive, and many handwritten documents produced during the Middle Ages are literally works of art which combine elaborate script with decorative scrollwork and pictures. Nevertheless, creating such documents was a tedious and time-consuming process, and only the wealthy could afford such books.

This changed, however, in the 15th century, when Johann Fleischmann von Gutenberg invented a practical printing press, the development of which increased the speed, quality, and availability of the written word. With the introduction of the printing press came the art of typography, the setting of movable and reusable type. Although the earliest type was designed to imitate hand lettering, stylized designs evolved, and people found that type improved their ability to read and thus to communicate. The subject of type and typography is discussed in more detail in a paper by Griffee and Casey[1] in this issue.

Another significant milestone in the printing industry was the invention of the typewriter, which was patented for the first time in 1714. The first commercial machine came later, in 1868. The typewriter finally made it easy for the author to print words on paper quickly, easily, and relatively inexpensively. However, the use of the typewriter still imposed some restrictions on the document creator: limited choice of type styles, fixed pitch, left-to-right and top-to-bottom presentation, and the scissors and glue pot often used to add illustrations.

With the advent of electronic data processing came a dramatic increase in printing speed, but at the expense of quality and flexibility. Computer-generated inventory reports, billing statements, and doc-

Figure 1 AFP print model



umentation could be produced in a fraction of the time that it took previously. However, when a company's image was important, computer-generated output was not acceptable, and expensive typeset forms or other typeset materials were required.

It would be useful to have the advantages of all of the systems just described—the flexiblity and aesthetics of a handwritten document, the ease of reproduction and quality of typesetting, the convenience and ease of use of the typewriter, and the speed of electronic data processing. In recent years, new technologies have become available that make all of this possible. One of the most important of these technologies has been the development of the laser printer, and what we call the electrophotographic printing process. This technology allows us to bring to business applications functions that have previously been available only in sophisticated typesetting and photocomposition applications.

In a laser printer,[2] a laser exposes or writes the information to be printed on the surface of a rotating photoconductive drum. That image is then coated with a toner resembling powdered ink. The rotating drum transfers the toner to the image electrostatically. The paper then passes through a heater or fuser station, where the toner is fused to the paper, thus bonding the image to the paper. The ability to focus the laser extremely accurately and the ability to produce a very small spot size allow the user to effectively draw every dot on the page. By controlling the intensity of the laser, it is possible to smooth character edges and achieve the appearance of a resolution higher than that provided by the actual number of dots per inch. With a resolution of 240 dots per inch (the resolution of the IBM 3800-3 printer), there are over five million dots on a page. It would be tedious indeed if a user had to specify each dot to be printed. IBM has introduced a system that is termed Advanced Function Printing (AFP) software, the purpose of which is to provide a powerful, easy-to-use set of tools with which users can exploit the capabilities of the laser printer without having to get down to the details of writing individual dots on the device. In this paper, we introduce an AFP model and two application environments for AFP. We then demonstrate how the model applies to these environments. Finally, we describe the specific AFP software components as they relate to the functions brought out in the previous discussions.

Advanced Function Printing

Print technologies now exist that allow us to describe every dot on the printed page. Often referred to as All Points Addressability (APA), this technology provides almost limitless possibilities for the preparation and presentation of information on the printed page. In order to harness this power and make it accessible to the user of electronic data processing systems, we have developed the architecture and software products that are the basis of Advanced Function Printing (AFP). Figure 1 illustrates the way in which AFP provides a powerful, easy-to-use set of tools by which the end users can exploit the capabilities of this new technology, while shielding the users' application from the details of managing a specific device.

AFP printers. The devices supported by AFP range from very-high-speed, fanfold printers to low-speed, cut-sheet printers. The print technologies include lasers, light-emitting diode printheads, and wire-matrix printheads. Each printer may have a different set of paper-handling options, recovery considerations, storage for print resources, and so on. Although we want our printers to present as identical an interface as possible, we do not want to inhibit the development of new technologies nor limit the way in which our hardware developers make effective cost/performance/function trade-offs and take innovative steps toward providing better hardware products for our customers. The model shown gives us the flexibility of dealing with new technologies and with printers with varying cost and performance options. At the same time, the model provides the end user with a stable, durable interface to printing by introducing a system-print-services component into the system between the application and the printer.

Currently used system line-printer data streams are accepted by PSF and converted into IPDS for printing on an AFP printer.

System print services. In the System/370 environment, the system-print-services component for Advanced Function Printing is the Print Services Facility (PSF).[3-5] The major function of PSF is to provide a system- and device-independent interface to printing. In this environment, the application need only have the ability to place data on the system spool. PSF then provides the following important functions: data-stream transformations, printer resource management, device driver, and error recovery.

Data-stream transformations. PSF provides its users with three different data-stream interfaces, which are described more fully in another paper in this issue.[6] Currently used system line-printer data streams are accepted by PSF and converted into IPDS for printing on an AFP printer. In addition, PSF accepts and processes composed-page data streams in Advanced Function Printing Data Stream (AFPDS) format or mixed-mode data streams, which include system line-printer data intermixed with AFPDS control structures. We describe the use of each of these in subsequent sections of this paper.

Printer resource management. PSF manages resources required to print, which include page definitions (PAGEDEFs) and form definitions (FORMDEFs). These are used by PSF to generate the print data stream and the page segments, typographic fonts, and overlays (electronic forms) that are sent to the printer for use during the print process. A set of predefined resources are provided with the AFP product set. Also, more complex functions can be provided with resources that can be created with a set of AFP utility programs, as we discuss later. An application can reference resources in the input data stream to PSF, or these resources can be specified external to the print data when scheduling a data set for printing. PSF either binds these references to an actual resource object that it has taken out of a named resource library and inserted into the output data stream, or moves the reference into the IPDS data stream when it knows that the required resource is resident at the printer. PSF communicates with the printer to query available resources and preload resources using the IPDS data stream.

Device driver. PSF drives the printers, which may attach to the system in a variety of ways. The print services component must understand the communications interface to each printer and provide the appropriate protocol, buffering, and device-management commands so that the data are delivered and executed by the printer. The IPDS data stream provides a two-way communication path to the printer, which is used by PSF to query the capabilities of the printer, initialize the printer, load and manage resources, and validate the receipt and successful processing of each message.

Error recovery. PSF provides device-specific error recovery. Different devices require different levels of error recovery, depending on the attachment, speed, length of paper path, and so on. PSF keeps track of the state of the device, checkpoints data on a user-defined boundary, and executes recovery procedures in a manner transparent to the application program. This means that PSF assumes the responsibility of ensuring that the data will be printed, without application involvement. For example, if the printer has a problem, PSF queries the printer for the number of the last page to be successfully printed. PSF will then automatically reposition to the next page. If the printer runs out of storage, PSF works with the printer to free storage, reload resources, reposition the print data, and retry the operation until it is successful. On a network-attached printer, if the line is dropped, PSF automatically restarts the printer once the line is restored, using checkpoint information. In all cases, print integrity is the number one priority.

In all cases, print integrity is the number one priority.

These services are provided in a centralized software component, so that there is a standard, system-supported interface to printing, independent of the specific AFP printer, its attachment to the system, or the operating system that runs it. This approach offers a number of unique advantages. For example, system print resources can be located in and managed by a central library that is transparent to the application. User exits can be provided for functions such as resource management, separator pages, accounting, etc. in a central place in the system.

Advanced Function Printing applications

Given this model for Advanced Function Printing, we now consider two of the major application environments for which AFP was designed.

Systems printing. Systems printing covers what we traditionally think of as being done in a raised-floor environment, and in fact, many of the AFP printers that IBM markets today are locally attached channel printers. However, products such as the IBM 3820[7] and Remote Print Manager[8] have moved systems printing out of the computer center and into distributed locations. When we think of system printing, we probably most often think of listings and dumps. One could fairly ask the question, "What is the value of Advanced Function Printing in these applications?"

Typographic quality. If no other changes are made, the use of high-quality, high-resolution fonts improves the readability of the output. Using smaller font sizes of good, readable quality allows the reduction of report size from 14⅞ × 11 inches to 11 × 8½ inches. This reduction in paper size makes these pages easier to file and eliminates large binders.

Multiple-up printing. Multiple-up printing simply means that two or more *logical pages* (i.e., numbered pages) are printed on one physical page, as is illustrated in Figure 2. We use the term logical page here to mean the page as the application program views it; the physical page represents the sheet of paper on which the logical page(s) is printed. When PSF prints this additional information on the physical page, a smaller-point-size font is used. In addition, the line spacing is reduced, and the output is rotated to make better use of the space gained.

To carry this a step further, we can print four-up, with four logical pages on one physical sheet of paper, as shown in Figure 3. In addition, duplexing now gives us eight logical pages (four on each side) on a single physical sheet of paper. The benefits are reduced paper costs, reduced usage costs, and increased throughput. In most companies, these savings are often enough to justify the installation of Advanced Function Printing.

To print with the capabilities that we have discussed so far, only the Print Services Facility (PSF) must be installed. Also provided with PSF is a library of predefined, commonly used Page Definitions (PAGEDEFs) and Form Definitions (FORMDEFs) that allow a print job to be specified with various combinations of paper sizes, page rotations, line spacing, two-up and four-up printing, etc. Predefined or customized FORMDEFs and PAGEDEFs may be specified when PSF is installed. This gives the installation the option of selecting a standard output format. Thus, if default PAGEDEFs and FORMDEFs have been installed for two-up, duplexed printing, they are invoked automatically for each print job, without application awareness that a transformation of the output data has taken place. All output would thus be in two-up, duplexed format, unless the defaults

Figure 2 **Multiple-up printing**

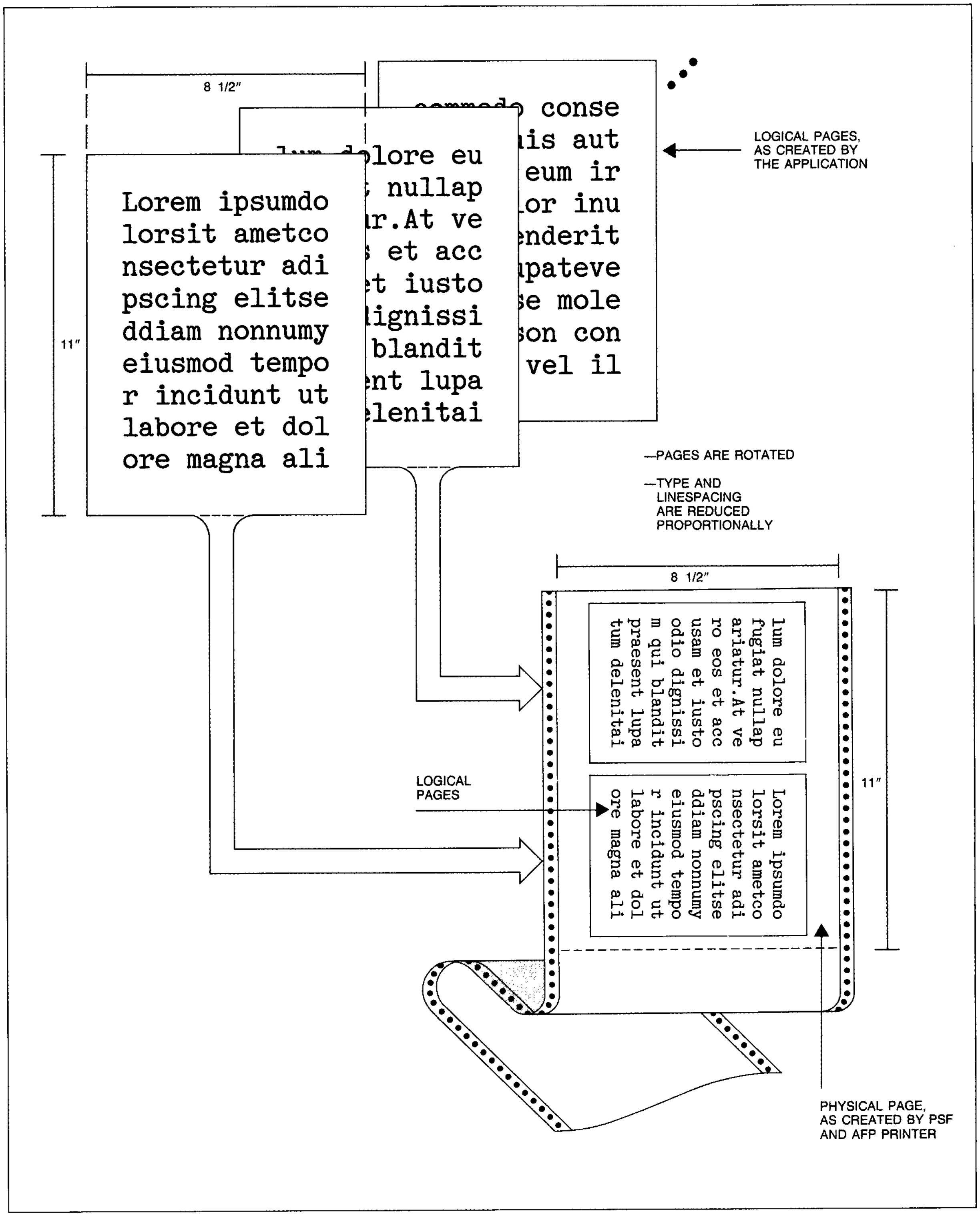
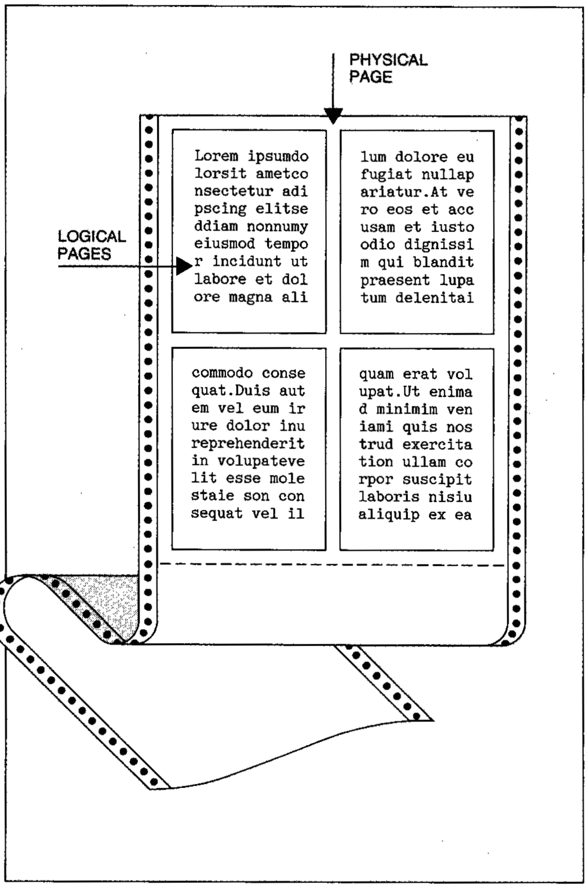


Figure 3 Printing four-up



had been explicitly overridden by specifying a PAGEDEF and/or FORMDEF when scheduling the output for printing, as, for example, parameters in a JCL OUTPUT statement. It is important to note that no changes to the application program are required in order to achieve these benefits. The application generates traditional system line-printer output, but we have provided for transformations of the line data to take place in the Print Services component of the system through the use of externally named resources. This is illustrated in Figure 4.

Before we discuss more complex transformations, we need to take a more detailed look at PAGEDEFs and FORMDEFs.

Form definitions. A form definition, required for all AFP print jobs, consists of two elements: a *document environment group* and a set of *medium maps*. The information contained in the document environment group applies to the entire document. This element defines the position of the logical page on the physical page, names any overlays to be used, and specifies any text suppressions that can be used for line data. Text suppressions provide a means for selectively suppressing the printing of blocks of data, such as in a blind carbon.

One or more medium maps in a form definition define additional print options for a collection of pages (called a *copy group*) in the document. A copy group represents a set of pages in a multipart form that have similar properties (such as number of copies), use of overlays, and duplexing. Thus we can define forms, as shown in Figure 5. These additional print options include the following:

- Selection of text suppressions
- Overlays to be used
- Offset stacking
- Edge marking (3800 and 3835 printers only)
- Forms flash (3800 printer only)
- Horizontal paper adjustment (3800 printer only)
- Paper bin selection for cut-sheet printers
- Duplexing for cut-sheet printers

Note that some of these functions apply to certain printers only. The default form definitions supplied with PSF contain only one medium map per form definition; thus, the same set of print options applies to the entire job. More sophisticated use of form definitions, where medium maps are invoked from within the print data set, are discussed later in this paper.

Page definitions. A page definition is a resource that contains formatting specifications for line data or unformatted print data. A page definition contains one or more *page formats*, each of which contains a complete set of page-formatting specifications. A page format contains formatting controls for a user file that indicate where and how text is to be placed on a page. The page format defines for all data specified in the page format the page size, which is specified as the width and height of the page. For line-data records to be printed as received, a line descriptor must be provided that decribes the format of each line in page format. This description includes the following:

- Starting position of each line as it is to appear on the page, which may override the line spacing and margins as described in the application specified for the page

Figure 4 Default PAGEDEF and FORMDEF create two-up output

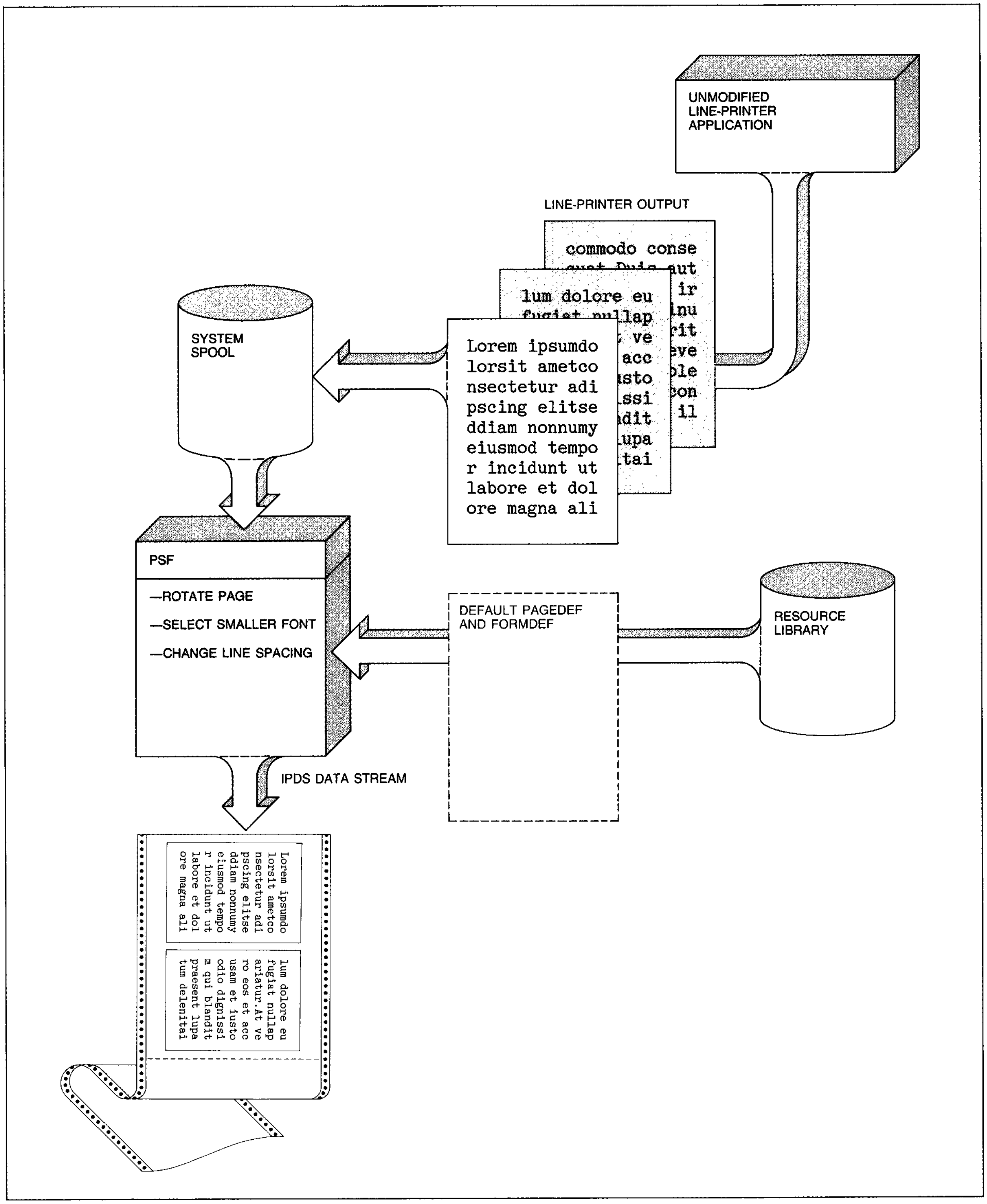
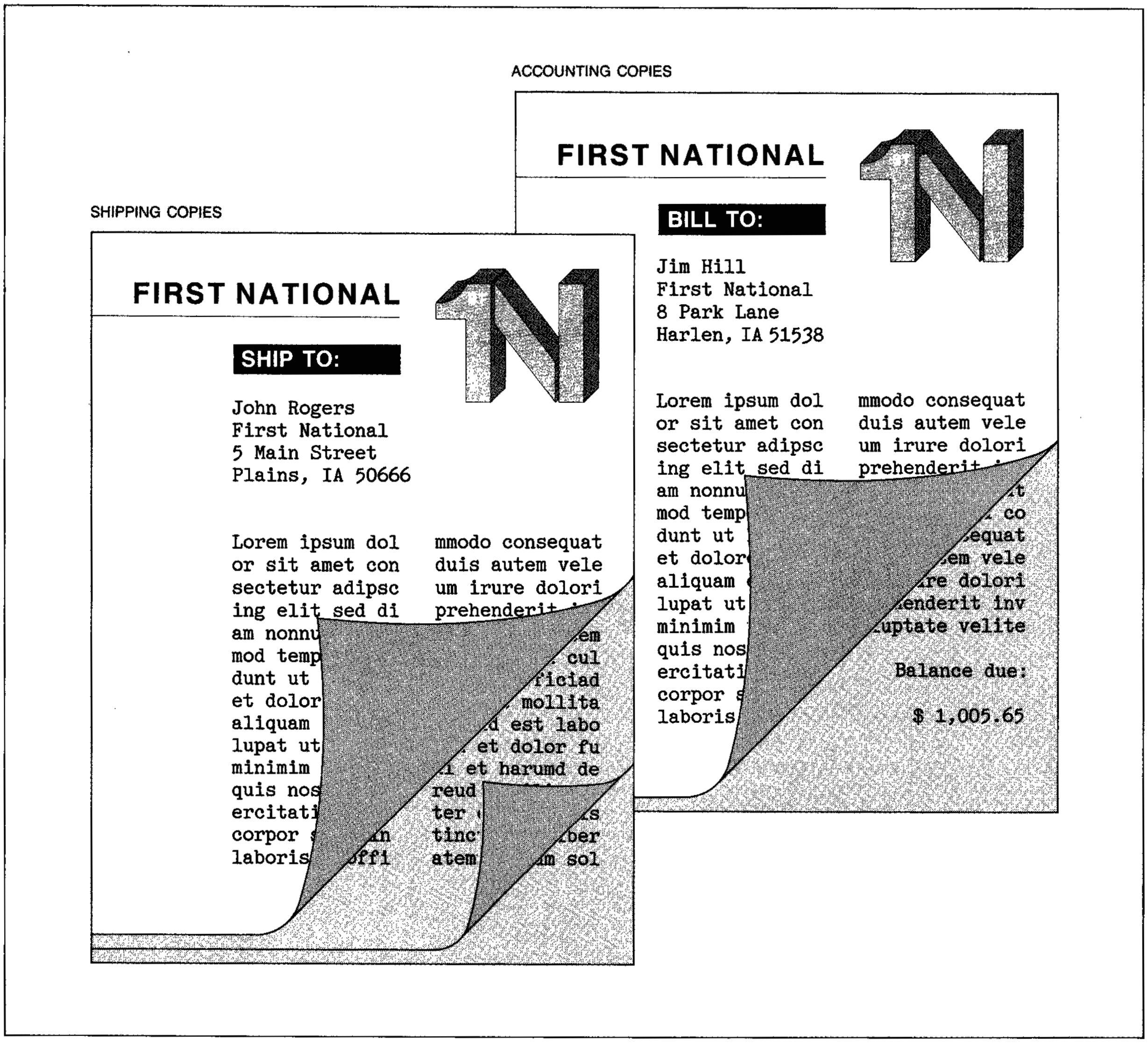


Figure 5 Multipart, multiple-copy forms



- Starting position on the page, when a defined channel code is encountered in the line data
- Name of the font for that line
- Printing direction and character orientation for the line

Up to this point, we have talked about simple transformations of line data to page format. The controls that we have just discussed provide for the functions of line-printer emulation, condensed printing, and multiple-up printing. However, if the user wants to format line data in a way other than originally generated by the application, additional information must be provided in the page format. This additional information defines the following:

- Locations and lengths of fields in the input record
- Placement, direction, and font for each field, as it is mapped into page format
- Suppression of fields, which is usually specified if multiple-page copies are printed with field suppressions on selected copies

- Constant data to be printed along with the fields from the input records

Figure 6 shows a typical line-printer output, and a possible remapping of that output onto a preprinted form. Notice how the remapping of the line data and the use of different fonts increase the usability of the document. Once again, this function was provided without altering the existing application program.

Recall that PSF is installed with predefined sets of page definitions and form definitions to provide standard output forms, such as two-up. When more sophisticated formatting is required, page definitions and form definitions can be created using an IBM utility program called Page Printer Formatting Aid (PPFA).[9] PPFA is a batch program that takes a user-written specification of a page definition or form definition and converts it into the AFPDS format necessary for processing by PSF. PPFA runs on MVS, VM, and VSE.

Electronic forms. In our previous examples, we showed information being printed on a preprinted form. There are a number of costs associated with preprinted forms that can be saved through the use of electronic forms. It may take days or weeks to have a new preprinted form designed and supplied in quantity for a production application. If a form needs rework or modifications, we must allow additional turnaround time and perhaps lose the money invested in the on-hand inventory of forms. Electronic forms, on the other hand, can be created inhouse in a matter of hours, and can easily be edited or modified, if necessary, before being used. Because the forms are generated as they are used, if changes to a form are required, there is no loss of inventoried forms.

In addition, throughput time can be saved with the use of electronic forms. With preprinted forms, an operator must stop the printer, load new forms, and restart the printer. The system has to be notified that

Figure 6 Line data remapped onto preprinted form

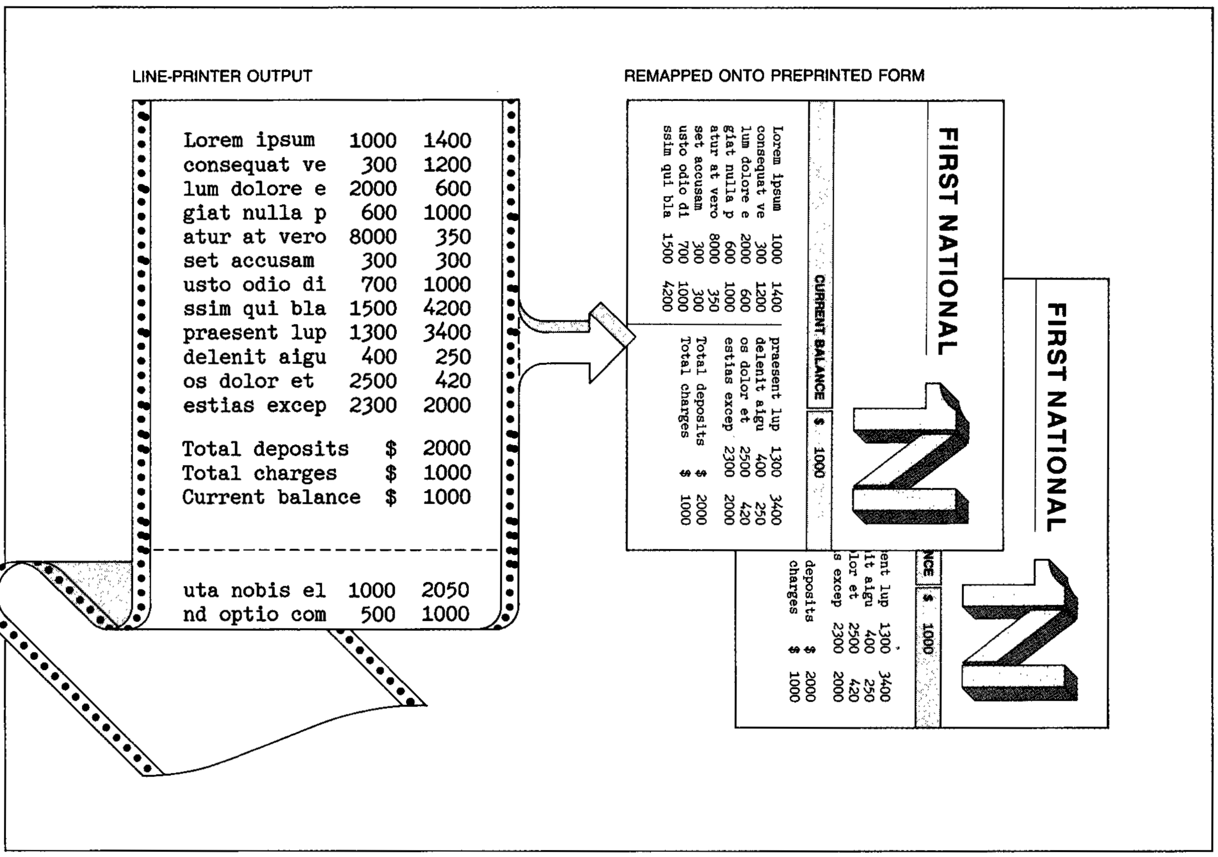
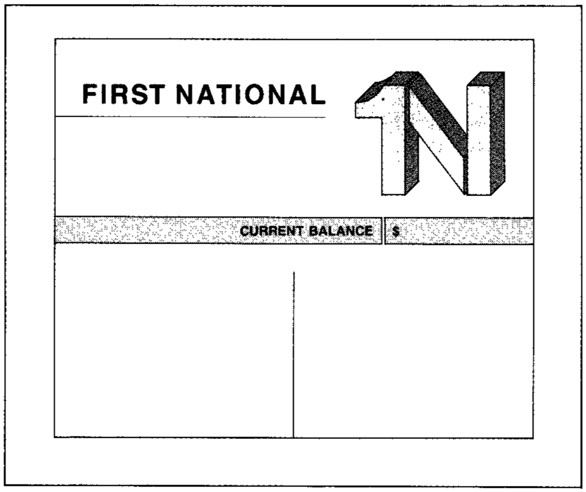


Figure 7 An electronic form



new forms are loaded, so that the right set of print jobs can now be scheduled for printing. With AFP and electronic forms, all job setup is handled by the system.

Electronic forms are generated in AFP as *overlays*, which are AFP resources that may contain vertical and horizontal rules (or boxes) with varying line weights, shaded boxes, and graphics elements that we define as page segments. Character data on the overlay may be defined with different orientations and with any of the typographic fonts available with AFP. A simple overlay, shown in Figure 7, contains boxes, graphics elements, shaded areas, and fixed text. Figure 8 shows the results of reformatting the line data and using an overlay.

Recall that overlays are specified as one of the optional parameters in a form definition. Multiple overlays are specifiable on a single page, providing the ability to build up a form from a number of predefined individual building blocks which may be used in different forms. Overlays are sent by PSF to the printer, where they are stored and used across multiple pages, just as a preprinted form would be. Forms that contain significant amounts of constant data can save considerable data transmission time, which is particularly important when the printer is not locally attached to the system.

Overlays can be generated by the Overlay Generation Language (OGL),[10] an IBM utility program. OGL is also a batch utility that accepts user-written overlay description statements and generates the overlay resource in AFPDS format.

Mixed documents. So far, we have not required any changes to the application program to achieve the AFP function. We have been able to use either predefined page definitions and form definitions or to build more sophisticated definitions that we install as defaults or invoke when scheduling the print. Obviously, there are many circumstances where we must alter the use of page definitions or form definitions after very little usage, and perhaps even allow the application to make these changes dynamically on the basis of decisions taken during the processing of the application data. This can be done with AFP by inserting AFPDS structures into the output data stream. This action invokes predefined page and form definitions.

Recall that a form definition can contain multiple *medium maps*. The medium map defines such elements as page position, duplex printing, and number of copies. By coding an *invoke-medium-map*-structured field into the line-printer data stream, a new medium map can be invoked. When a new medium map is named in this way, printing continues, using the new specification. If we want to invoke the medium map named "GROUP2," the encoding of the structured field in the data stream is the following:

5A 0010 D3ABCC 00 0002 "GROUP2"

where

5A indicates a structured field,
0010 specifies the length in bytes of the structured field,
D3ABCC identifies this as an *invoke-medium-map*-structured field,
00 is a reserved field,
0002 assigns a sequence number for recovery purposes,
and
"GROUP2" is the name of the medium map to be invoked.

Similarly, we can change the name of a page format to be used by inserting an *invoke-data-map*-structured field into the output data stream. The invoke-data-map-structured field, which is similar to the invoke-medium-map-structured field previously discussed, selects one of a set of prespecified page formats from the active page definition. Other AFPDS

Figure 8 Merging remapped line data and overlay

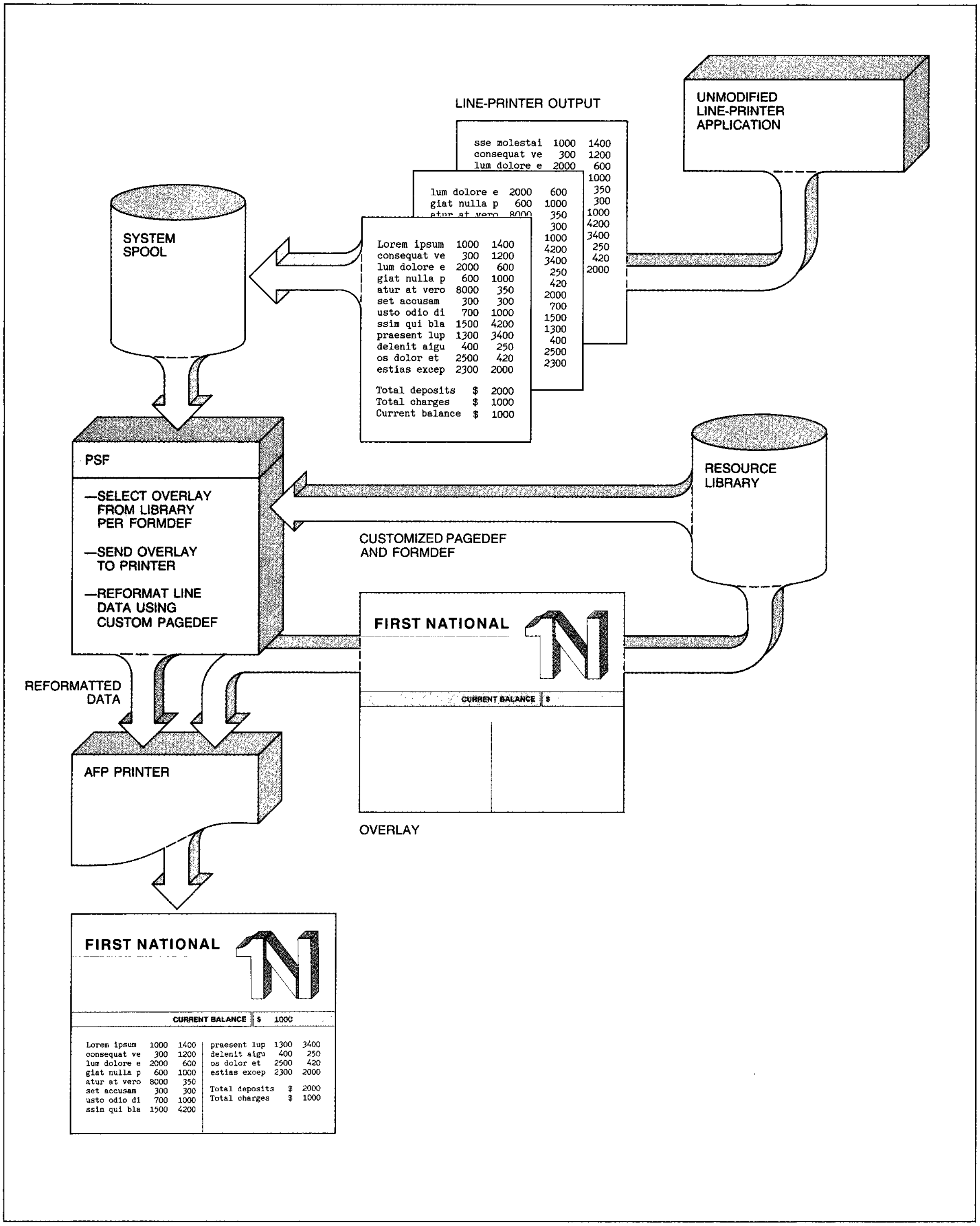


Table 1 AFPDS structured field that can be mixed with line data

Abbreviation and Name	Identifier (hexadecimal)	Record Length (bytes in hexadecimal code)	Description
IMM: Invoke Medium Map	D3ABCC	10	Selects the copy-group definition
IDM: Invoke Data Map	D3ABCA	10	Selects the page format
CTX: Composed-Text Data	D3EE9B	8–7FEF	Includes text on a page
Image definition controls:			
BIM: Begin Image Block	D3A87B	10	Begins an image definition
IOC: Image Output Control	D3A77B	20	Positions an image on a page
IDD: Image Input Descriptor	D3A67B	2C	Specifies the size of an image
IRD: Image Raster Data	D3EE7B	8–7FEF	Describes an image raster pattern
ICP: Image Cell Position	D3AC7B	14	Specifies size, position, and repeating of image cells
EIM: End Image Block	D3A97B	10	Ends an image definition
IPS: Include Page Segment	D3AF5F	16	Specifies a page segment

structures that can be included in a mixed-line document are shown in Table 1.

Document publishing. The other major application area for which AFP is designed is in-house document publishing. This includes such functions as the preparation and printing of technical correspondence and proposals, presentations, newsletters, telephone directories, parts lists, and maintenance manuals. These kinds of applications normally generate fully-composed-page AFPDS data streams. IBM offers two companion products to the AFP product set that help in creating such composed-page documents.

Document Composition Facility. Document Composition Facility (DCF) prepares input for printing by formatting text into columns, changing headings and footlines, placing footnotes, creating a table of contents, index, and title page, and by performing other text-formatting functions.

DCF supports the use of a Generalized Markup Language (GML) which allows the user to identify logical document structures, such as chapters and paragraphs. Specific formatting can then be defined for each logical structure. For example, the control **:H1** defines a heading level 1, which typically corresponds to the chapter heading in a book. The **:H1** control can result in the following specific formatting taking place:

- Skip to a new page.
- Skip down one inch.
- Turn on centering.
- Put the text of the heading in uppercase 14 point bold italic Times Roman.
- Skip one inch after the heading.
- Put the heading text into the table of contents.
- Put the heading text in a running footline at the bottom of odd-numbered pages in this chapter.

DCF has access to the font descriptions used by PSF in printing a document; thus, the text of the document can be laid out very precisely, with page endings and column endings carefully computed on the basis of the fonts that are used when the document is printed. DCF allows the formatting of text, the insertion of graphics images into documents, and the drawing of horizontal and vertical rules of varying thicknesses. In addition to DCF, IBM offers a set of complementary publishing programs for this environment, including MARKUP,[11] DRAWMASTER,[12] and BOOKMASTER.[13]

Graphic Data Display Manager. The Graphic Data Display Manager (GDDM)[14] is an IBM component that provides an application interface for creating graphics images. GDDM is used by a number of other IBM applications and by user-written applications to support graphics workstations on the System/370 host. One of the functions provided by GDDM is the conversion of any graphics data in its database into an AFPDS image page segment. These page segments can then be stored as resources available to PSF. Page segments can be identified in fully composed documents, such as those produced by DCF, or they may be included in a document by referencing the page segment from an overlay definition.

AFP product summary. We now expand our original print model to include the various components that we have discussed up to this point, as shown in Figure 9. At the heart of the model is the system print services component PSF. Application products, such as DCF, build fully composed AFPDS data streams that PSF transforms to IPDS for printing. Other major IBM software systems that support this interface are DisplayWriter/370 and GDDM's CDPU program. In addition, user-written applications that fully exploit the capabilities of AFP are written to this interface.

These functions can be accomplished without any impact on existing application code.

On the right side of Figure 9 are shown those applications that today generate line data or mixed line data, as we have described it. PSF uses externally defined resources called page definitions and form definitions to transform the line data into page format. A number of utility programs exist to aid the application developer in defining these resources. We have already discussed Overlay Generation Language (OGL) and Page Printer Formatting Aid (PPFA). In addition, there is a utility called Print Management Facility (PMF)[15] that also provides many of these functions. PMF provides a menu-driven interface for creating page definitions and form definitions. PMF takes image data which may have been generated by some external source, such as a scanner, and converts them into an AFPDS page segment. In addition to these functions, PMF provides an interactive font editor and a set of font-library-management functions. Another font utility is the Font Library Service Facility (FLSF).[16] FLSF lets the user add characters to a font, rotate characters, customize code pages, and perform a number of other font-library-management functions.

Finally, IBM has provided a menu-driven interface for invoking print services. Print Services Access Facility (PSAF)[17] gives the user a set of interactive menus that allow the description of a print job. PSAF then creates the appropriate job-scheduling commands for the operating system and builds simple page definitions and form definitions for a print job, with a few simple instructions.

As we have shown, a user does not need all of these components to begin using Advanced Function Printing. PSF and the IBM-supplied page definitions and form definitions provide an extensive set of functions, high-quality fonts, condensed printing, and multiple-up layout capabilities. Furthermore, these functions can be accomplished without any impact on existing application code.

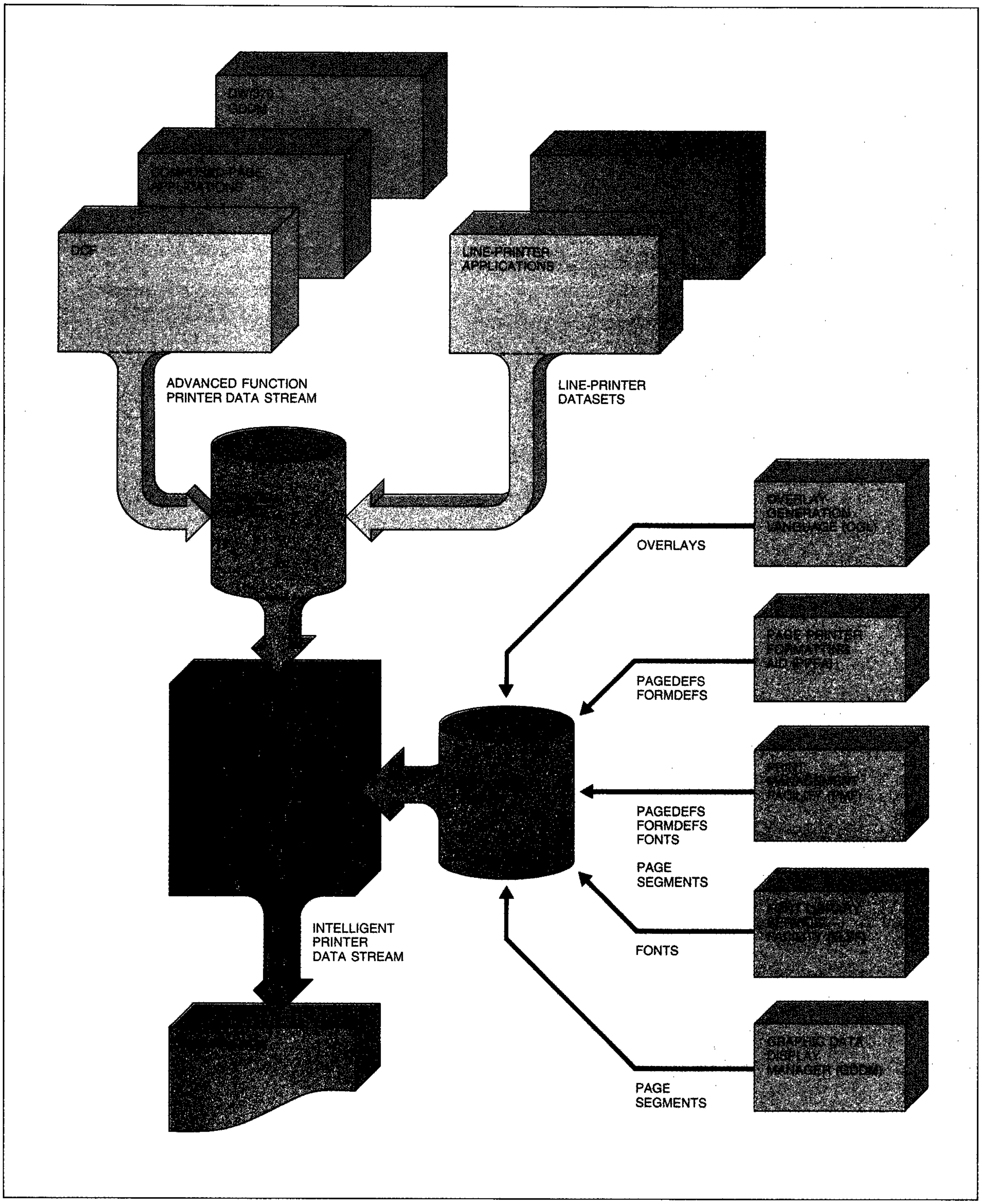
More of the AFP capabilities can be exploited by adding PPFA and creating one's own page definitions and form definitions, by using OGL to create one's own electronic forms, or by using one of the font utilities to create or customize one's own fonts.

Concluding remarks

Advanced Function Printing is an IBM product for systems printing. We have demonstrated how the structure of the AFP print model and the architectures that support AFP have been created to support the user's existing application investment and provide smooth migration and growth into new functions. In what ways might we expect AFP to grow in the future?

Customers have noted a number of requirements that point in the direction in which we also believe we must proceed. There is a need for the ability to print documents in an office location, data center, or individual workstation with identical results. There is also a requirement for compatibility, so that any output generated on a central computer, mini, or workstation can be printed on any printer in the enterprise. Implicit in the understanding of the architecture is that IBM program products can be created to serve the entire range of printers. By maintaining an open architecture and using the layering described in this paper, we can distribute more and more print function into the network while maintaining a consistent application interface to AFP.

Figure 9 AFP print model



Cited references

1. A. W. Griffee and C. A. Casey, "An introduction to typographic fonts and digital font resources," *IBM Systems Journal* **27,** No. 2, 206–218 (1988, this issue).
2. C. D. Elzinga, T. M. Hallmark, R. H. Mattern, Jr., and J. M. Woodward, "Laser electrophotographic printing technology," *IBM Journal of Research and Development* **25,** No. 5, 767–773 (September 1981).
3. *Print Services Facility/VM User's Programming Guide,* 5664-198, IBM Corporation; available through IBM branch offices.
4. *Print Services Facility/MVS User's Programming Guide,* 5665-275, IBM Corporation; available through IBM branch offices.
5. *Print Services Facility/36 User's Guide and Reference,* S544-3485, IBM Corporation; available through IBM branch offices.
6. R. K. deBry, B. G. Platte, C. L. Berinato, and J. W. Marlin, "Architectures of Advanced Function Printing," *IBM Systems Journal* **27,** No. 2, 234–245 (1988, this issue).
7. *IBM 3820 Page Printer Reference Manual,* S544-3175, IBM Corporation; available through IBM branch offices.
8. *IBM Personal Computer Remote PrintManager User's Guide,* S544-3439, IBM Corporation; available through IBM branch offices.
9. *Page Printer Formatting Aid User's Guide and Reference,* G544-3181, IBM Corporation; available through IBM branch offices.
10. *Overlay Generation Language User's Guide and Reference,* SH35-2079, IBM Corporation; available through IBM branch offices.
11. *Markup User's Guide and Tutorial,* S544-3350, IBM Corporation; available through IBM branch offices.
12. *Drawmaster User's Guide and Reference,* SC34-5022, IBM Corporation; available through IBM branch offices.
13. *Bookmaster User's Guide and Reference,* SC34-5009, IBM Corporation; available through IBM branch offices.
14. *Graphical Display Data Manager General Information Manual,* GC33-0100, IBM Corporation; available through IBM branch offices.
15. *Print Management Facility User's Guide and Reference,* SH35-0059, IBM Corporation; available through IBM branch offices.
16. *Font Library Service Facility Installation and Operation Manual,* SC33-6166, IBM Corporation; available through IBM branch offices.
17. *Print Service Access Facility User's Guide and Reference,* S544-3407, IBM Corporation; available through IBM branch offices.

Roger K. deBry *IBM Information Products Division, 6300 Diagonal Highway, Boulder, Colorado 80301.* Dr. deBry received his Ph.D. from the University of Utah in 1972 in electrical engineering. He subsequently took an assignment in Kingston, where he was responsible for the IBM 3270 display data-stream architecture definition, and in 1979 received an Outstanding Innovation Award for his work. In 1985 he published the book *Communication with Display Terminals,* McGraw-Hill Book Company, Inc., New York. Dr. deBry worked for two years on the Corporate Programming Staff in Purchase, New York, where he focused on architecture and system design issues relating to device support. During this assignment, he became interested in software support for laser printers. As a result, at the completion of his corporate staff assignment he moved to Boulder as technical assistant to the programming center manager. Dr. deBry has been in this position for two years, and has been responsible for developing a printer software strategy. He is a member of ACM and an active participant in IFIPS, where he is the U.S. representative to IFIPS TC2, Programming.

Brian G. Platte *IBM Information Products Division, 6300 Diagonal Highway, Boulder, Colorado 80301.* Mr. Platte joined IBM in 1970 as a programmer, and is currently a senior programmer in the Page Printer System Design Department, where he is responsible for page printer system design. For the past nine years he has worked on APA printers and Advanced Function Printer software. Mr. Platte's responsibilities have included IBM 3820 software support, page printer software planning, system design, and data-stream architecture. Prior to working on printers, he worked on the IBM 3850 Mass Storage System, where he was responsible for the microcode. Mr. Platte earned a B.S. degree in mathematics at the Central Michigan University, Mount Pleasant, Michigan. He received an M.S. degree in computer science from the University of Colorado, Boulder, in 1975. Mr. Platte has received two patents, one of which was for the IPDS architecture, which also earned him his first Invention Achievement Award, an IPD President's Award, and an Outstanding Technical Achievement Award.

Reprint Order No. G321-5321.