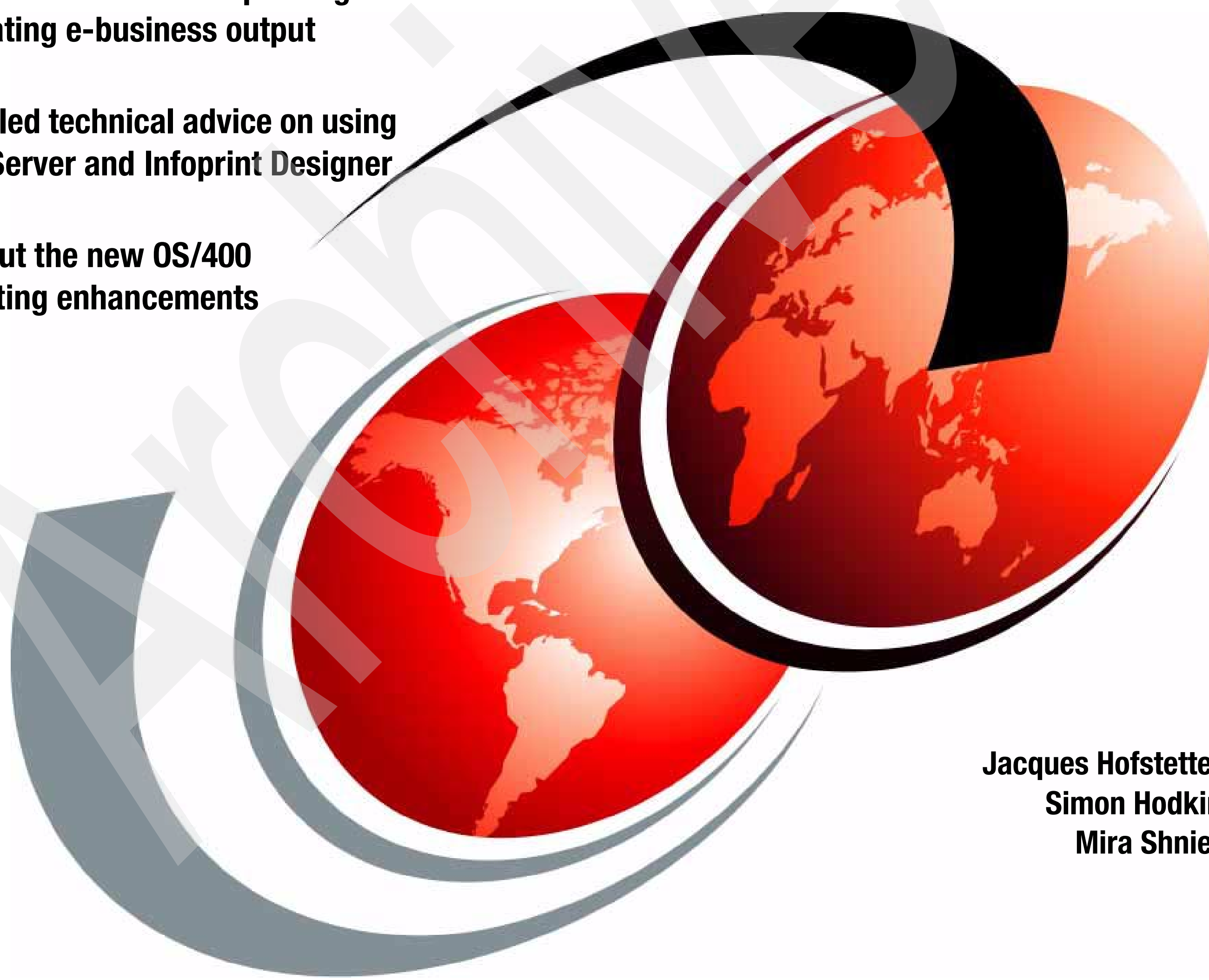
IBM @server iSeries Printing VI

Delivering the Output of e-business

- Learn how to select iSeries printing software for generating e-business output
- Gain detailed technical advice on using Infoprint Server and Infoprint Designer
- Learn about the new OS/400 V5R1 printing enhancements



Jacques Hofstetter
Simon Hodkin
Mira Shnier

ibm.com/redbooks

Redbooks

International Technical Support Organization

SG24-6250-00

IBM @server iSeries Printing VI: Delivering the Output of e-business

May 2002

Take Note!

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

First Edition (May 2002)

This edition applies to Infoprint Designer for iSeries (5733-ID1) available with OS/400 V4R5 and OS/400 V5R1, and Infoprint Server for iSeries (5722-IP1) for use with the OS/400 V5R1.

This book was updated on May 15, 2003.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- eserver™
- eserver™
- Redbooks(logo) ™
- ibm.com®
- iSeries™
- pSeries™
- xSeries®
- zSeries®
- Advanced Function Presentation™
- Advanced Function Printing™
- AFCCU™
- AFP™
- AIX®
- AS/400®
- Balance®
- BookMaster®
- Domino™
- DB2®
- DFS™
- DXT™
- EDMSuite™
- GDDM®
- ImagePlus®
- Infoprint®
- Intelligent Printer Data Stream™
- IBM®
- IMS™
- IPDS™
- Lotus Notes®
- Lotus®
- Network Station®
- Notes®
- OfficeVision/400™
- OfficeVision®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Perform™
- Print Services Facility™
- Redbooks™
- RMF™
- RS/6000®
- S/390®
- Sequent®
- System/36™
- System/38™
- VisualAge®
- VisualInfo™

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

Following on from the successful IBM Redbook *AS/400 Printing V*, SG24-2160, this new printing redbook concentrates on products, methods, and techniques used in delivering the output required by IBM @server iSeries e-business. It introduces and explains many new printing developments with OS/400 V5R1.

In addition, this book offers detailed technical advice on using two of the most significant iSeries printing products for years: Infoprint Server for iSeries and Infoprint Designer for iSeries. *Infoprint Server for iSeries* extends the capabilities of the iSeries beyond printing, to further enable such e-business requirements as data stream and image transforms, spooled file indexing and segmentation, e-mail and Web-enabling. *Infoprint Designer for iSeries* provides the long-awaited, fully-graphical document composition interface to Advanced Function Presentation on the iSeries.

This book will be of value to iSeries customers, IBM Business Partners, and IT specialists who want to learn about the wealth of new printing and output presentation functions available at OS/400 V5R1. It is the essential guide to delivering the output of e-business.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Jacques Hofstetter is a Senior IT Specialist in Switzerland. He has 12 years of experience in Printing on the iSeries and has worked at IBM for 19 years. He has provided implementation, education and consulting services to a wide variety of customers. His areas of printing expertise include Advanced Function Presentation, PostScript, and other data streams across different system platforms. Jacques was also on the *AS/400 Printing V* redbook team.

Simon Hodkin is a Senior IT Specialist with Printing Systems for IBM UK. He has worked at IBM for 15 years. During the last six years, Simon has been the lead UK person for AS/400 and iSeries printing, devising and implementing printing solutions for customers. In addition, he provides printing education and consulting services to customers and IBM Business Partners. He was also on the *AS/400 Printing V* redbook team.

Mira Shnier is a Senior IT Specialist for the Printing Systems Division of IBM Canada. She has worked for IBM since 1977, and for the last 10 years has specialized in printing on the AS/400 and iSeries, providing implementation, education, and consulting services to a wide variety of Canadian customers. Mira has contributed to other AS/400 printing redbooks in the past, including the last version of *AS/400 Printing V*.

Thanks to the following people for their invaluable contributions to this project:

Ken Dittrich
Brian R Smith
ITSO Rochester

Tim Aden
Miranda C Albracht
Rebecca K. Dimock
Artur K Ford
Patricia Godfrey
Rondi Hermann
Diane Hohense
Jack Klarfeld
Bruce Lahman
Linda Liebelt
Alan Middendorf
Dale Pirie
Bill Shaffer
Howard Turetzki
Farzad M Valad
Nancy Wood
IBM Boulder (US)

Nancy O'Brien
IBM Charlotte (US)

Eddy Gauthier
IBM Belgium

Janice Bosley
Angela Cheng
Mary Curl
Roger Drolet
Kim Hirt
Sayed Jalees
Guy Roy
George Voutsinas
IBM Canada

Kurt Bosshart
Stefan Schmidlin
IBM Switzerland

Larry M Dunn
Dave H Hall
IBM UK

John Holder
J.D. Edwards

Massimo Galizia
Jean-Marc Piaget
La Suisse Assurance, Switzerland

Notice

This publication is intended to help customers, business partners, and IBM system engineers who need to understand the fundamentals of printing on the iSeries server. It will help them develop or advise others about the design and development of iSeries printing applications. The information in this publication is not intended as the specification of any programming interfaces that are provided by Print Services Facility/400, IBM Font Collection, Infoprint Designer, and Infoprint Server. See the PUBLICATIONS section of the IBM Programming Announcement for Print Services Facility/400, IBM Font Collection, Infoprint Designer, and Infoprint Server for more information about what publications are considered to be product documentation.

Comments welcome

We want our redbooks to be as helpful as possible. Send us your comments about this or other redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829

Chapter 1. iSeries printing software to create, manage output

IBM offers a wide variety of iSeries printing software solutions for the creation, formatting, and management of your electronic documents. They range from products designed for an application developer or programmer, to ones that provide an easy to use graphical user interface (GUI). Several products have overlapping functions, while others complement each other. Some are priced program products, some are included as part of other products, and some are even available at no charge.

This chapter helps you understand the role of the different offerings and assists you in the selection of the correct product or products for your environment. We have categorized them in terms of the stages of the development and the types of objects involved.

For the sake of this discussion, we look at the offerings in terms of:

- Formatting your application data
- Creating the external resources
- Driving the printer, including data stream transformations
- Managing the printed output
- Complementary offerings from IBM
- Complementary non-IBM offerings

You can find additional details on most of these offerings in other sources, such as product manuals and users' guides, other IBM redbooks, or on the IBM Web sites, in particular at: `http://www.ibm.com/printers/iseries`

The goal of this chapter is to introduce each of the products to you and provide enough information for you to compare them and choose which might be appropriate. After you make that decision, you could refer to other sources for more in-depth descriptions. For additional information, see the reference materials listed in Appendix G, "Related publications" on page 357.

Naturally, there will be some overlap between the categories in the list above. For example, as explained in 1.1.2, "Infoprint Designer for iSeries" on page 5, the Infoprint Designer package has components for application formatting as well as resource creation within the one offering.

1.1 Formatting your application data

Most output consists of variable data that is generated by some sort of application or program. This section deals with the software tools that work directly on the data. This can be as simple as deciding what setting to use for characters per inch (cpi) or lines per inch (lpi) to use for the entire document.

The next level of complexity may be to add some fixed resources, such as an electronic form (overlay) or a logo (page segment) to the document. (The decision on how to use these resources is part of the overall design. The offerings for purchasing or creating these static resources are described in 1.2, "Building or obtaining external resources" on page 14.)

You may then want to move data fields around, change fonts, suppress data, print multiple copies, or even convert some of the data to a barcode. Some of the offerings support the insertion of logic to choose what is to be done based on the content of the page or line.

In this section, we describe the software offerings that are used to format application data and lay it out on the page. Many of them are built on the Advanced Function Presentation (AFP) architecture. They include:

- Printer files, including those built using Data Description Specifications (DDS)
- Infoprint Designer for iSeries
- AFP PrintSuite - Page Printer Formatting Aid (PPFA/400)
- AFP PrintSuite - Advanced Print Utility (APU)
- AFP PrintSuite - AFP Toolbox
- SAP R/3 AFP PrintSuite feature
- AFP Utilities/400 - Print Format Utility

1.1.1 Printer files and DDS

Printer files and DDS have been enhanced for Version 5 Release 1 of OS/400. The concept of printer files and DDS has been around since the introduction of the IBM System/38 in 1979. Naturally the supported functions have evolved over time as print technology has changed.

All print applications use printer files, whether explicitly or implicitly. The process of copying data to a printer file causes a spooled file to be generated and placed in an output queue.

Printer files can contain two different types of information:

- Printer file parameters that control how spooled file data is processed
- Printer Data Description Specifications (DDS) that actually define how application data is positioned on the output page

The first set of output information, the printer file parameters, are part of any printer file on the system. The second set, the printer DDS, is optional and exists if the application program has externally-described output. A compiled printer DDS source file also results in a printer file, but one with more detailed formatting instructions.

1.1.1.1 Printer files

There are over 75 parameters in the Create Printer File (CRTPRTF) command. Any value used here pertains to the entire printer file, unless it is overridden by entries in DDS. Some of the things you can do by using a printer file alone are:

- Select the data stream of the spooled file to be generated
- Select the page size, default characters per inch, and default lines per inch
- Add an offset to move the data down and to the right of the top left corner
- Set print quality
- Select an input drawer and output bin
- Select a default font and code page
- Set page rotation, duplex, or multi-up printing
- Add an electronic overlay to front or back
- Control stapling and finishing
- Set the output queue, form type, page range and copies
- Determine if line data is to converted to AFP data stream

If you consider these options, you will notice that the most detailed level of formatting possible is to the document or file-level. For example, you can change the lines per inch, the font, add an electronic overlay, and so on. But, these changes affect every page of the spooled file.

Some parameters can only be used on spooled files that are in specific formats. For example, you can only use the Front margin (FRONTMGN) parameter if the printer file used generates AFPDS (AFP data stream). For a complete description of values that can be set in a printer file, see the reference material as described in Appendix G, “Related publications” on page 357.

Printer files may be modified separately from the application program that is using them. Use the Change Printer File (CHGPRTF) command to make permanent changes. The Override with Printer File (OVRPRTF) command can be use to make a temporary change. A subset of the parameters may be changed after the spooled file has been generated using Change Spooled File Attributes (CHGSPLFA).

1.1.1.2 DDS keywords

There are over 60 DDS keywords available for use with printer files. They can be used to change the format of the data being printed down to the record or field level, for example, a customer address in a different font, with the zip code printed as a barcode. Some DDS keywords can be used for all spooled file types. Other keywords such as OVERLAY, LINE, or FNTCHRSET are specific to spooled files that are generated using DEVTYPE(*AFPDS).

For an online source of the latest DDS keywords, see the iSeries Information Center at: `http://publib.boulder.ibm.com/pubs/html/as400/`

Then navigate via the Online Library to the V5R1 section and select **Database and file systems-> DB2 UDB for iSeries**. The DDS manual has been extracted into several sections, including one entitled *DDS Reference: Printer files*.

1.1.1.3 Using a printer file with an application program

Figure 1 shows you how a simple printer file with DDS can affect the format of the resulting output. The application program issues an instruction to write a record of data. Using the information in the printer file, the appropriate data stream commands are built into the resulting spooled file and the output will have the desired characteristics. In this example, the printer file was created specifying a default font 11 (Courier medium at 10 cpi). In the DDS, the keyword FONT(46) acts upon just the name field to cause it to print in Courier bold. An additional keyword, UNDERLINE, acts only upon the city field.

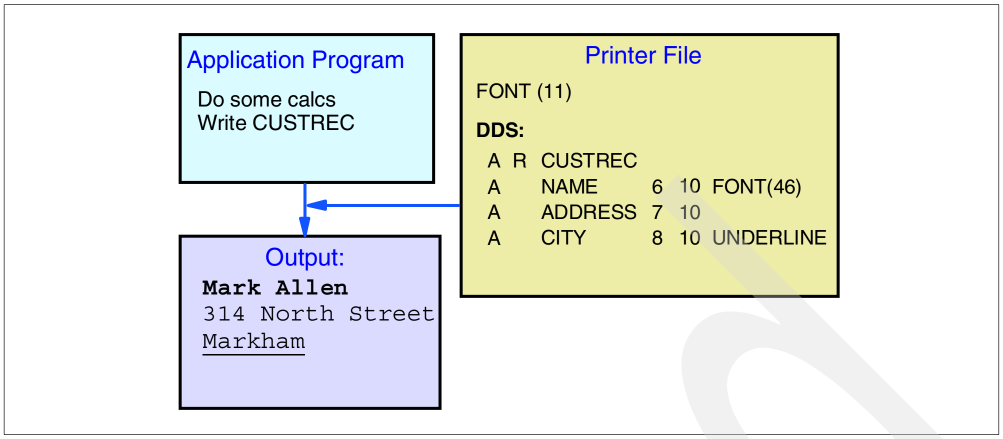


Figure 1. Using a printer file

The layout of the data is not specified explicitly in the program; it is picked up from the DDS when the program is compiled. It is possible to make minor changes to the DDS, such as changing a field's font or position, without recompiling the program. More drastic changes, such as changing the size or number of fields will likely require you to recompile the program, and possibly make corresponding program changes.

1.1.1.4 P-values

Some DDS keywords support the use of variable information through the use of program-to-system fields. These can be used to change the name of an external resource being used or the position of some print elements. Using this technique, the programmer can decide which electronic form (overlay) or image (page segment) to use based on some logic in the program.

Because DDS can be tightly linked to the application program, it is possible to create quite complex pages, often limited only by the skill and imagination of the programmer.

1.1.1.5 When to use printer files and DDS

You can use printer files if you need only to make minor changes that affect the entire file. A common example is to add an overlay or change the bin number. No programming skills are required.

If you have access to the application source and have the need to use high level language logic to control the layout of the data and resources, DDS may provide the flexibility that you need.

You may need DDS if you want to insert the break sequences to allow you to take advantage of the new e-mail capability to send different pages of your data to different recipients. The advantages and disadvantages of DDS are outlined in Table 1.

Table 1. Advantages and disadvantages of DDS

Advantages	Disadvantages
No charge; part of OS/400	Not a graphical user interface (GUI)
Integrated with iSeries applications	Programming skills/resource required

Table 1. Advantages and disadvantages of DDS

Advantages	**Disadvantages**
Powerful document formatting capabilities	May need to consider application requirements while designing output with DDS

1.1.2 Infoprint Designer for iSeries

Infoprint Designer for iSeries is a new program product (5733-ID1), available with OS/400 V4R5 or higher. It provides a new way for you to format your data and create AFP overlays and page segments. It has a true what you see is what you get (WYSIWYG) interface that allows the user to see exactly how the application will look while it is being worked on. The package includes overlay creation and data layout functions that are closely integrated, plus a separate Image Editor program for producing and editing page segments.

This section provides an overview of the data layout component. For an overview of the overlay creation and ImageEditor functions, see 1.2.1, “Infoprint Designer for iSeries” on page 15. Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33, provides a full description of the Infoprint Designer product.

1.1.2.1 Designing the layout

The first step in working with a print application using Infoprint Designer is to import a sample spooled file. This could be an iSeries spooled file, or a file in “line data” format that has been stored on the PC. You then drag the required field or fields from the data window to the layout window. You can easily change the format of these fields, such as fonts, interline spacing, or rotation, at the click of a button.

If you have already created an overlay, it will be displayed on the layout panel in gray. If you need to change part of the overlay, you can easily switch modes. The data in your layout will now be in gray and you can work with the overlay.

For a basic design, you can start with mapping the data to the new layout. This replicates your existing stationery, with the overlay replacing the pre-printed elements. A more advanced user can add additional copies, constant back, multi-up, or conditional processing. For example, you can choose different page layouts based on the contents of pre-defined fields.

Under the covers of Infoprint Designer, AFP objects called *form definitions* and *page definitions*, are being created. However, an end-user does not need to be concerned with that until some of the more advanced functions are used.

At the current release, Infoprint Designer only produces traditional record formats, based on line data. That is, the data must be in a predictable format, so you can define fields as “total’, “customer number”, and so on. Future enhancements to the product *may* incorporate advances in data formatting such as Record Format Architecture. IBM understands that supporting the Record Format Architecture is a requirement, but has not yet committed a date for delivery of this function.

1.1.2.2 Using the objects created by Infoprint Designer

After the design is complete, the resources are uploaded to the iSeries server. The user is responsible for integrating them into the applications. You do not need to make any changes to the programs that generate the data. However, the printer files being used need to be changed or overridden to generate line data and to reference the form definition and page definition created. This is described in Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33. These resources can be used on any AS/400 or iSeries server from V3R2 and V3R7 onwards.

Note that Infoprint Designer is not required at run time. Therefore you could run Infoprint Designer on your development system and use it to produce AFP resources for use on this and any production systems.

If you are printing to an IPDS printer that is configured with AFP(*YES), you can have the application generate LINE data. The page and form definitions, fonts, overlays, and so on, are processed by Print Services Facility/400 (PSF/400) at print time. If you want to use the AFP Viewer, or to send the output to an ASCII printer using Host Print Transform, use the Convert Line Data (CVTLINDTA) parameter in the printer file. This causes the output to be generated as fully-composed AFP data stream (AFPDS). The CVTLINDTA command is described in 3.2.1, “CVTLINDTA” on page 96.

1.1.2.3 When to use Infoprint Designer

Infoprint Designer is a very easy to use data layout tool, especially for the basic layout functions. The WYSIWYG view provides instant feedback and can improve the designers productivity versus the iterative approaches needed by the other “programming” tools, and even the “green screen” GUI used by Advanced Print Utility (APU). Figure 2 shows the product running under Windows.

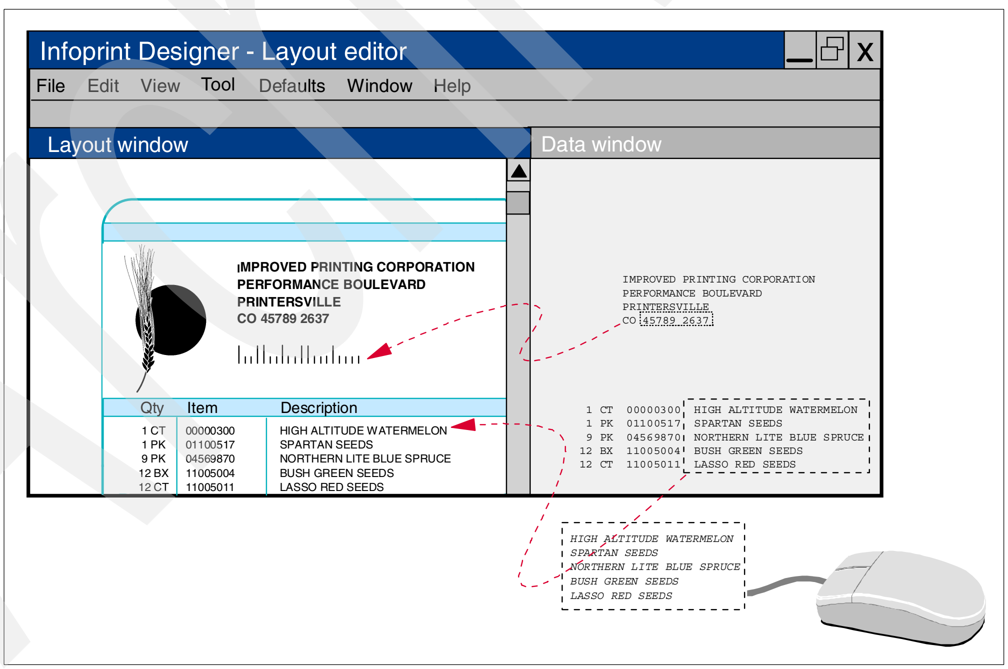


Figure 2. Drag and drop data mapping with Infoprint Designer for iSeries

Infoprint Designer can be used whenever the source data is generated in a predictable manner. This is often the case when the application is being migrated from pre-printed forms. However be aware that you may need access to change or override the printer files in order to produce line data, rather than the SCS data that the iSeries normally generates. This is discussed in greater detail in Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33, and 2.10, “Putting the application into production” on page 82.

Table 2 outlines the advantages and disadvantages of Infoprint Designer.

Table 2. Advantages and disadvantages of Infoprint Designer

Advantages	Disadvantages
WYSIWYG overlay, image and document design	Data must be in a fixed or predictable position in order to re-format
Powerful conditional processing	Some knowledge of the principles of page and form definitions may be required
Efficient - works with native AFP resources such as fonts and page segments	Printer file must be changed or overridden to produce line data in order to use page and form definitions
Entirely automatic when in production use	
Not required at run-time	

1.1.3 AFP Printsuite – Page Printer Formatting Aid/400

Page Printer Formatting Aid/400 (PPFA) is a separately charged feature of AFP PrintSuite (5798-AF3). PPFA is a page formatting language that allows you to create objects called *page definitions* and *form definitions*. These two object types can then be used externally to the application program to format the data.

In the past, PPFA was more common as the formatting tool for OS/390 or AIX customers, or ones that had AS/400 or iSeries servers in a mixed environment. With the enhancements being made for V5R1, it is expected that more customers may choose this route for formatting their output. These enhancements include:

- The ability to resolve LINE data to AFPDS with the Convert Line Data (CVTLINDTA) printer file parameter or Create AFP Data (CRTAFPDTA) command
- Support on the iSeries server for Record Format Line data

These enhancements are described in more detail in the OS/400, Infoprint Server, and PPFA chapters of this book.

The Infoprint Designer product provides a WYSIWYG alternative to PPFA coding. It is described in 1.1.2, “Infoprint Designer for iSeries” on page 5.

1.1.3.1 PPFA resources

As mentioned above, PPFA creates two type of AFP resources, a form definition and a page definition.

The *form definition* controls how the logical page is placed on the physical medium – the sheet of paper. Source statements within the form definition specify what drawer the paper is selected from, what overlay is to be used, whether duplexing is used, whether multiple logical pages should be placed on a single physical page, what copies are to be automatically created, and what fields should be suppressed from which copy.

The *page definition* defines how data is placed on a logical page layout. Input print lines are read in, optionally parsed into individual fields, and placed on the page. Similar in structure to DDS, the page definition language of PPFA enables you to place print lines or print fields anywhere on the page while controlling font, orientation, and color characteristics. Data can also be printed as a barcode.

Page definitions can contain conditional logic. This means that the formatting rules can change based on the contents of an input field. A trigger field (for example, company number) may be used to select a whole series of formatting commands. Figure 3 illustrates how page and form definitions work.

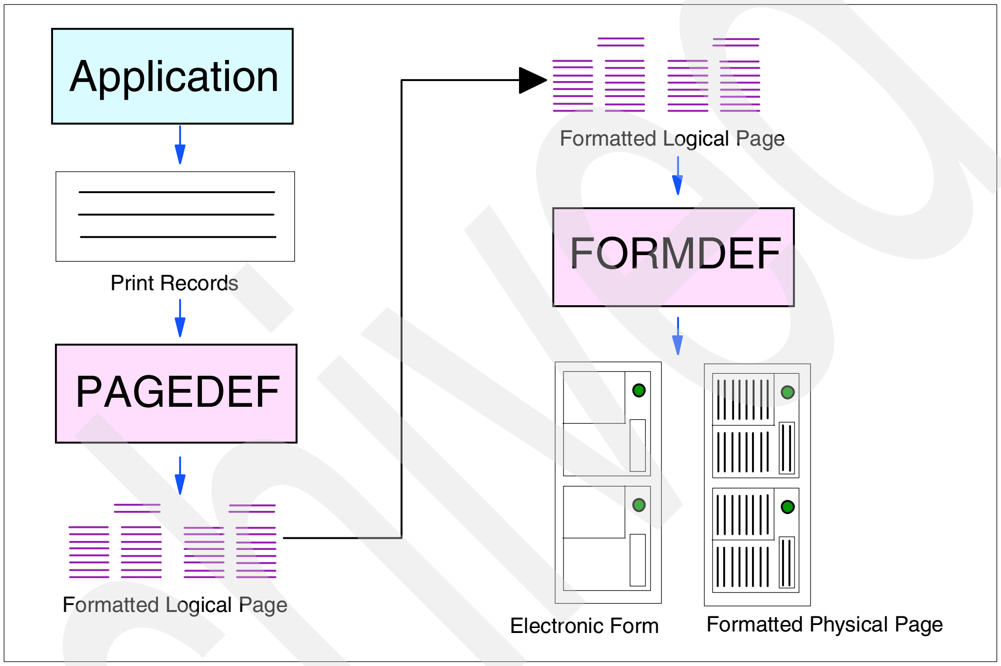


Figure 3. Concept of page and form definitions

1.1.3.2 Using PPFA

To create form definition and page definition using PPFA, you first enter the source statements into a source physical file on the iSeries. You then use the PPFA command Convert PPFA Source (CVTPPFASRC) to generate physical files that contain the AFP instructions. The form and page definition objects are then built (compiled) using the OS/400 commands Create Form Definition (CRTFORMDF) and Create Page Definition (CRTPAGDFN).

The printer file refers to these two object types with the FORMDF and PAGDFN parameters. A form definition can be used with spooled files that are SCS, AFPDS, AFPDSLINE, or LINE. Page definitions can only be used with spooled files that are generated as LINE data.

The application that generates the data does not need to be changed to use these PPFA objects, with the exception of a one-time change to the printer file to generate *LINE data and reference the new resources. The formatting rules that are defined in the form and page definition are processed by PSF/400 at print time. Note that this means that the formatted job (with overlays, fonts, and other resources) is created only at print time. Therefore, if you require a fully-composed AFP file to be created (for example, for viewing, archiving, or using with Host Print

Transform), you must use the CVTLINDTA command as mentioned previously (see 1.1.2.2, “Using the objects created by Infoprint Designer” on page 5).

1.1.3.3 When to use PPFA

If you do not have access to the application programs, you may want to consider using PPFA to create form definitions and page definitions to format your data. Be aware that you may need access to change or override the printer files in order to produce line data, rather than the SCS data that the iSeries normally generates. The considerations are the same as for the layout tool in Infoprint Designer. This is discussed in greater detail in Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33.

You may want to use PPFA if you are also doing AFP processing on IBM @server zSeries, pSeries, and xSeries servers. As of Version 5 Release 1 of OS/400, PPFA/400 on the iSeries supports all the same functions as these other platforms.

If you are developing new applications, the Record Format Line data support, which is available in PPFA, provides a new way to deal with different types and amounts of data on the same page. Each record includes a format identifier that determines how the data is to be mapped. It supports different types of lines such as page headers, body, page trailers, and groups. Graphical elements, such as lines, boxes, circles and ellipses, can be placed directly on the page. These new functions are ideal for printing complex statements that have multiple types of data such as may be found on a financial statement or telephone bill.

This new function is discussed in greater detail in Chapter 12, “Using form definitions and page definitions” on page 255.

Table 3 outlines the advantages and disadvantages of using Page Printer Formatting Aid/400.

Table 3. Advantages and disadvantages of PPFA/400

Advantages	Disadvantages
Powerful page formatting tool; finishing and record-level formatting supported	No GUI design
No changes to the original application	Requires knowledge of the principles of page and form definitions
Resources may be used on other IBM platforms	Requires a programmer’s approach to source coding, if not an actual programmer
No run-time compiler needed; resources “plug into” OS/400 printer file	Printer file must be changed or overridden to produce line data in order to use page and form definitions
Entirely automatic when in production use	

1.1.4 AFP Printsuite – Advanced Print Utility (APU)

Advanced Print Utility is a separately charged package of AFP PrintSuite (5798-AF3). It enables you to use SCS files as input and then to transform that input to “full-page” electronic output, with pages that include electronic forms, image, barcodes, lines, boxes, and text in a variety of fonts.

APU is designed to make it very easy to migrate applications that print on multi-part impact printers to a laser print environment. You do not need to modify any existing application programs or spooled files.

1.1.4.1 How APU works

During the design stage, you display an existing spooled file on any 5250 style terminal and set up the formatting rules in what is known as a *print definition*. You can change the fonts, move, remap, or suppress fields, or you could have them print as barcodes. Constant text, overlays, page segments, and boxes can be added to the page. You can produce multiple copies of the spooled file using different formatting rules. For example, you could have a “Terms and Conditions” page printed on the back of the customer copy of an invoice, but not on the file copy. You may choose to suppress the amount fields on the packing copy and print barcodes of the part number instead.

You can also choose to apply different layouts to different pages in your document based on the content of the page. For example, you may want to use a special layout or overlay for documents that have “PAGE 1” printing at the top and a different layout for all other pages.

When you are ready to go into production, you set up the APU output queue monitor to watch for spooled files that match certain criteria, such as file name, user data, or form type. When there is a match, APU takes the input SCS spooled file, applies the rules that you set in the print definition, and builds a new spooled file using the AFP data stream.

Figure 4 shows the possible results of using APU.

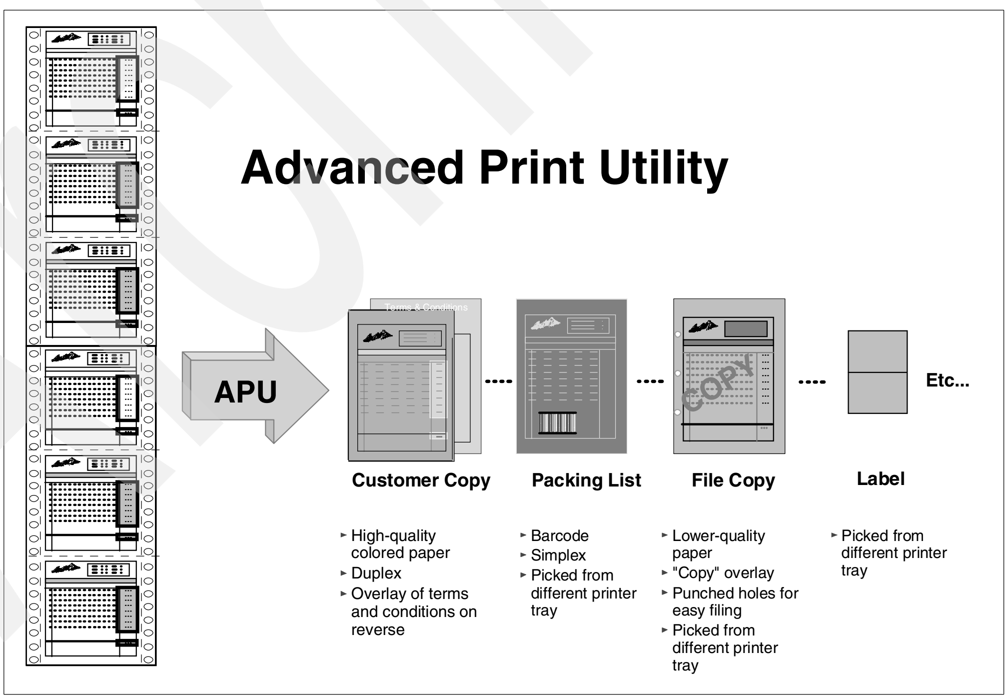


Figure 4. Results possible when using Advanced Print Utility

1.1.4.2 When to use APU

APU is a good choice when your application programs generate SCS and you have absolutely no access to any of the source, not even to the point where you would be able to change the printer file type to DEVTYPE(*LINE), as needed by Infoprint Designer or PPFA.

It provides a “green screen” approach that is familiar to many iSeries customers and may be considered to be a mid point between true WYSIWYG programs such as Infoprint Designer and the programming needed for DDS or PPFA.

APU has more flexibility than PPFA and Infoprint Designer when it comes to how multiple copies of the same page can be formatted. With PPFA and Infoprint Designer, copies of the same page can have a different overlay or paper source, but the data mapping is essentially the same. The only data mapping differences would be the ability to turn on or off suppression for selected fields for different copies. With APU, each copy of a given page can have completely different field mapping. If this is a requirement, APU would be the product of choice.

The APU monitor also provides additional functions not available with any of the other formatting applications. It is possible to use the monitor to route files to different destinations, even if they are not to be reformatted. The monitor also provides the ability to process one spooled file multiple times, either to the same printer or different printers. For example, you may have a program that generates one copy of a spooled file, and you use APU to send one copy to the office printer for mailing to the customer, and the second copy, with suppression of the prices, to the warehouse printer.

Table 4 outlines the advantages and disadvantages of using APU.

Table 4. Advantages and disadvantages of Advanced Print Utility

Advantages	Disadvantages
Easy to use	Design interface is 5250 “green-screen”
No changes to the original application	Data must be in a fixed or predictable position in order to re-format
Entirely automatic when in production use	No support for graphics or finishing
Add overlays, images, barcodes and different fonts to your output	
Dynamically select different page layouts and/or overlays based on your data	
Send completed documents to different printers and/or at different locations	
Replicate multipart stationery	

1.1.5 AFP PrintSuite – AFP Toolbox

AFP Toolbox is a separately charged package of the AFP PrintSuite (5798-AF3). AFP Toolbox provides a high-level, application-programming interface that enables programs written in the ILE C, ILE COBOL, and ILE RPG programming languages to produce an AFP data stream directly. The MO:DCA-P, or AFPDS documents created with AFP Toolbox, are platform-independent and can be printed or viewed by all the major IBM platforms, that is, iSeries, zSeries, pSeries, xSeries, and their predecessors AS/400, S/390, RISC/6000, and Intel platforms. AFP Toolbox output can be written to a file or returned to the application for further processing.

The iSeries version of AFP Toolbox consists of RPG, COBOL, and C libraries containing the procedure function calls to be included in the application program. The document is built up using these function calls in a hierarchical fashion; a document contains pages, and pages are made up of various objects.

The text processing components are much more advanced than the other products available from IBM for the iSeries. AFP Toolbox supports use of justified typographic fonts, variable sized, dynamically-built boxes, and indexing tags for use in archive/retrieval applications, or for viewing. One of the key features is the way AFP Toolbox can handle text in the form of paragraphs. Paragraphs of text can be defined and dynamically formatted. The text can flow and be aligned based on the area of the page allotted for it. Figure 5 shows some of the advanced formatting options including left and right-justification of typographic fonts, columnar alignment of typographic fonts, a dynamically-sized box and variable page segments, and a “page x of y” page count.

A skilled application programmer who understands the document structure is needed to create applications with the AFP Toolbox.

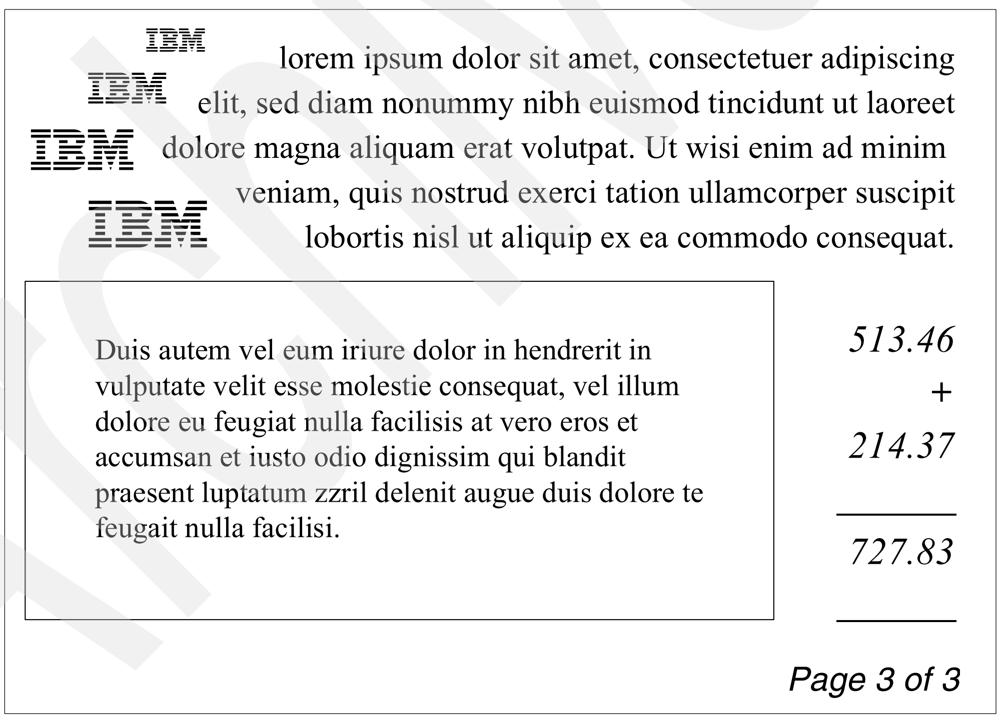


Figure 5. Some of the formatting options with AFP Toolbox

1.1.5.1 When to use AFP Toolbox

AFP Toolbox is the solution when the customer has a requirement for very complex, highly formatted output. Examples might be complex statements such as one might find in the financial industry where the document content is highly variable. The ability of AFP Toolbox to deal with paragraphs of data makes it ideal for complex letter creation.

AFP Toolbox has also been used by customers who want to convert applications that had printer-specific sequences embedded in the output. The AFP Toolbox allowed them to move to a printer independent and platform independent output format, while still maintaining much of the original program logic.

The advantages and disadvantages of using AFP Toolbox are highlighted in Table 5.

Table 5. Advantages and disadvantages of AFP Toolbox

Advantages	Disadvantages
Very powerful document formatting tool	High level of programming skill required
Only solution for justified typographic fonts	Access to application source required
May be incorporated into existing application logic	
Low-cost software licence	

1.1.6 AFP Utilities for iSeries – Print Format Utility

Print Format Utility is a component of AFP Utilities for iSeries (5722-AF1) for V5R1. It is included with the AFP Utilities for iSeries (AFPU) product. It enables you to build AFP reports or documents directly from iSeries database files without needing any application programs or any programming (Figure 6).

Print Format Utility (PFU) is like an AFP version of Query/400, producing high-quality output that features overlays, image, barcoding, and more. Just as you would use Query/400 to select database records and produce a report, you can use Print Format Utility to select database records and create advanced, full-page AFP output. With comprehensive barcode support built in, PFU is especially adept at quick barcoded applications.

The design editor of PFU is very similar to the visual Overlay Utility editor. (This is described in 1.2.2, “AFP Utilities for iSeries” on page 16). You can include the same kinds of electronic form elements in your report design. Essentially, you design the layout for one label and then tell PFU how many times to print it across and down the page. Some elements, such as page segments may be selected based on information in the database.

Print Format Utility also allows you to select fields for page breaks, and you can print summary information such as totals or counts on each page, or at the end of each group.

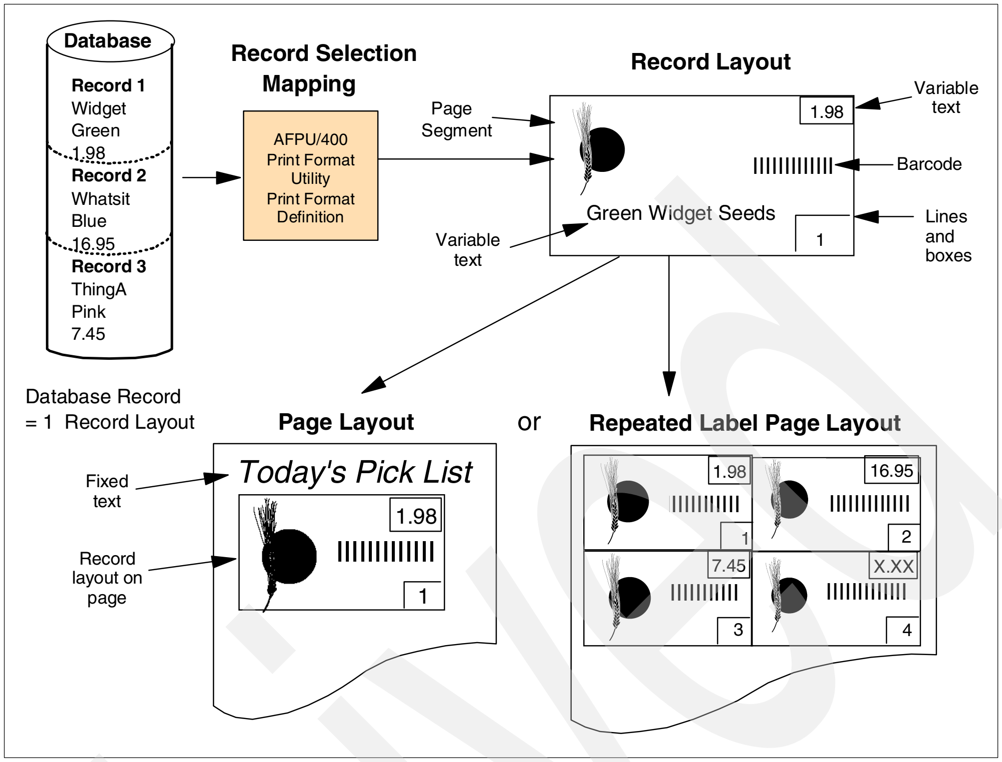


Figure 6. Use of Print Format Utility to extract and format records from a database file

1.1.6.1 When to use Print Format Utility

PFU is an easy tool to use for simple reports. It provides more formatting than Query and allows you to add AFP resources. Customers needing to print labels will find a quick way to produce them without any programming. Table 6 outlines the advantages and disadvantages of using PFU.

Table 6. Advantages and disadvantages of Print Format Utility

Advantages	Disadvantages
Easy to use; simple query-like use	Only acts on a single database file (programming resource needed to create a single logical file)
No programming required	AFP Utilities/400 required at run-time
PFU print definitions may be incorporated into CL programs	

1.2 Building or obtaining external resources

Most output consists of variable data that changes for every page such as individual customer invoices or statements. This data is merged with fixed elements that are constant. These fixed elements or resources do not change on a day-to-day basis. This section describes the IBM AFP software offerings for the creation of these fixed resources, namely:

- Overlays or electronic forms
- Page segments, such as logos or signatures
- Fonts

Strictly speaking, form and page definitions are also considered to be external resources. However, it is more logical to consider them in the section on formatting your variable data (1.1, “Formatting your application data” on page 1).

1.2.1 Infoprint Designer for iSeries

Infoprint Designer (5733-ID1 for V4R5 or V5R1) has three components, of which two are considered to be primarily resource-creation tools. The third component, Infoprint Designer Layout Editor, creates formatting resources, for example, form and page definitions, which are described in 1.1.2, “Infoprint Designer for iSeries” on page 5.

1.2.1.1 Infoprint Designer Overlay Editor

This part of the Infoprint Designer product creates overlays, in full WYSIWYG mode, on a Windows PC. It actually uses native AFP objects, such as fonts and page segments, as well as common PC elements such as TIFF, BMP, and JPEG files. The AFP resources are uploaded and created on the iSeries server with a “one-click” operation. Advanced graphics features, such as rounded boxes, dashed or dotted lines, shading, diagonal objects, and circles, may be created quickly and easily.

1.2.1.2 Infoprint Image Editor

The Infoprint Image Editor can be used both to “clean-up” scanned image files at the individual picture element (pel) level and to create AFP page segments from such files.

1.2.1.3 When to use Infoprint Designer

Infoprint Designer is a one-product solution to the problem of matching electronic forms (overlays) with variable data. The same interface is used both for designing the overlay and for integrating the variable data with that overlay.

Use Infoprint Designer when you want development staff (especially non-programmers) to quickly generate print applications using a Windows GUI interface. The advantages and disadvantages of using Infoprint Designer are outlined in Table 7.

Table 7. Advantages and disadvantages of Infoprint Designer

Advantages	Disadvantages
Windows GUI overlay and image design, full WYSIWYG	Only one active user per iSeries
Easy download of sample iSeries spooled file data	Printer file must be changed or overridden to produce line data in order to use page and form definitions
One-step upload and creation of AFP resources	
Overlay and image elements are native AFP objects; will match iSeries AFP objects	
Existing overlays or pre-printed forms may be used as templates for new overlays	

1.2.2 AFP Utilities for iSeries

In 1.1.6, “AFP Utilities for iSeries – Print Format Utility” on page 13, we explain AFP Utilities for iSeries (5722-AF1) in relation to the Print Format Utility application. Now we consider AFPU as a resource-generation tool.

1.2.2.1 Overlay creation

AFPU includes an Overlay utility. This permits the near-WYSIWYG creation of electronic forms, or overlays, directly on the iSeries server.

At first glance, the on-screen display looks nothing like the final printed form (see the example in Figure 7). However since V4R2, you can use an extremely useful feature. Assuming you have Client Access/400 or Client Access Express, you can type *VIEW in the Control field of the Design Overlay screen. This invokes the AFP Viewer function within Client Access/400, and the overlay is displayed with full fidelity, except for graphics and barcodes.

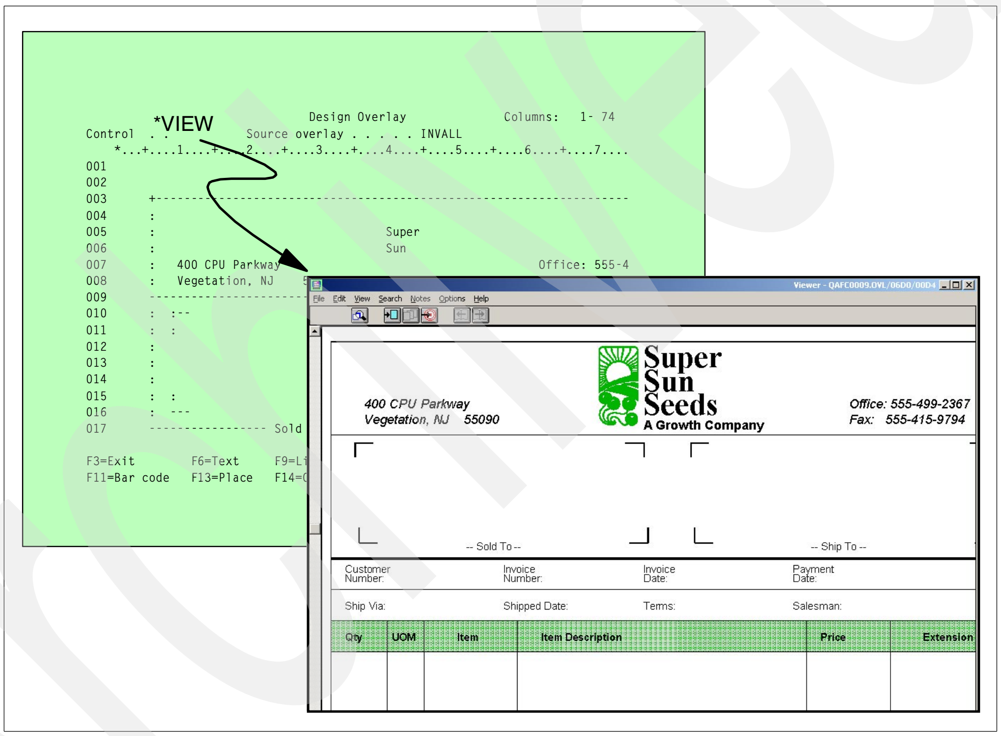


Figure 7. Using the WYSIWYG AFP Viewer with AFP Utilities for iSeries

You can use the on-screen iSeries GUI to design the overlay and then refine the position of overlay elements down to 1/100” (or centimeter). This is particularly useful when later mapping data using a product such as Advanced Print Utility; you can correlate the position of overlay elements with those of variable data.

Perhaps the major advantage of the Overlay utility is its ease of use. With simple step-by-step instructions for drawing boxes, lines and text, you can be reasonably proficient within an hour.

1.2.2.2 Page segment creation

AFPU also includes a utility to create images such as company logos and signatures. These are known as *page segments* in the AFP architecture. Creation of page segments is found within the Resource Management Utility within AFPU.

The principle is that you obtain your image (for example, from a scanned source) and create an IBM Image Object Content Architecture (IOCA) version of the source file. Then AFPU “wraps” it and produces an AFP page segment. The source file may be in an iSeries physical file, or more likely, exist as a PC file in the iSeries Integrated File System, placed there using iSeries NetServer or a similar method.

How do you create an IOCA image in the first place? Windows graphics packages, such as HiJaak Pro, RasterMaster, and others, can convert standard image file formats such as BMP, GIF, or JPEG into IOCA (with a file suffix of *.ICA).

Once the IOCA file exists, AFPU can resize or rotate the page segment it produces. There are also various mapping options such as a scale to fit option that is useful for keeping a logo or signature in proportion while fitting it into a particular size area.

1.2.2.3 When to use AFP Utilities for iSeries

If you are creating multiple similar overlays that may be updated fairly frequently (for example, twice a year), AFPU is very convenient. The source is stored safely on the iSeries server and can be accessed from any 5250 session. Extremely precise overlays may be created that are very efficient in terms of file size and therefore printing performance. You can see the advantages and disadvantages if AFP Utilities for iSeries listed in Table 8.

Table 8. Advantages and disadvantages of AFP Utilities for iSeries

Advantages	Disadvantages
Easy to use, quick learning curve	Only a near-WYSIWYG view in design mode on iSeries “green-screen”
Call to AFP Viewer provides WYSIWYG view of overlay	Expensive if AFPU used just for overlays, unless the license is on a smaller iSeries server
Produces efficient overlays	

1.2.3 AFP Font Collection for Workstations and OS/400

The AFP Font Collection for Workstations and OS/400 (5648-B45) Version 2.1.1 is an inexpensive collection of AFP and compatible Type 1 and CID keyed outline fonts, compatible across several platforms and covering approximately 46 languages. As of OS/400 V4R5, the single-byte character set (SBCS) fonts are included free of charge with any new PSF/400 order.

A free tool is available on the IBM Printing Systems Web site to help you use your AFP fonts. You can find it at: `http://www.printers.ibm.com/R5PSC.NSF/Web/rdfont01`

The RDFONT01 package includes two tools:

- **LOADFNTC**: Select the languages and the object types you need. Then this program builds them into libraries on your iSeries server.

- **PRTFNTC**: Selectively print fonts from the AFP Font Collection or printer resident fonts.

1.2.3.1 AFP Type Transformer and Utilities for Windows

This is an optional, priced feature of the AFP Font Collection. It allows you to take PC outline fonts (such as Windows TrueType and Adobe Type 1) and convert them to AFP fonts, either in raster or outline format, or both. A Windows NT 4.0 platform is required to run the Type Transformer.

The Type Transformer feature is also required if you want to create raster versions of the double-byte character set (DBCS) fonts, since the latter are supplied in outline format only.

1.2.3.2 When to use the AFP Font Collection

This product is especially useful for customers who:

- Need to have consistent font appearance on different printers
- Have different printers with different resolutions
- Have operations in more than one country or region
- Want to maintain the fidelity of printed output to media such as fax and viewing
- Need support for the Euro character
- Need to convert PC fonts to AFP format (using Type Transformer)

1.2.4 AFP Fonts/400

This section refers to the products 5769-FN1 (DBCS) and 5769-FNT (SBCS). The available fonts are listed in *AS/400 Printing V*, SG21-2160, in the Fonts chapter.

1.2.4.1 When to use AFP Fonts/400

You should note that these font products supply only 240-pel raster fonts and for selected font families such as Sonoran Serif. Therefore, order them only if you have a very specific need for the particular font families.

1.2.5 AFP Driver for Windows

The AFP Print Driver for Windows is supplied with products such as Client Access, Content Manager OnDemand, and AFP Workbench. It is also available on an “as-is” basis from the World Wide Web at:

`http://www.printers.ibm.com/R5PSC.NSF/Web/afpdr`

The driver can be used to create:

- AFP versions of your PC documents for printing on IPDS printers
- AFP overlays
- AFP page segments

See the example in Figure 8. The resources may be created as full-page image, giving true fidelity at the expense of a large file size, or as text objects. The latter method maps AFP fonts to PC fonts and requires that the AFP Font Collection be installed on the iSeries server. Using this method, the file size is much smaller, however, and the print quality may be better. On some IPDS printers, there may be a performance delay while downloading complex AFP resources produced using this tool. This is usually negligible or not the case with Advanced Function Common Control Unit (AFCCU) printers such as the Infoprint 4000 or Infoprint 62.

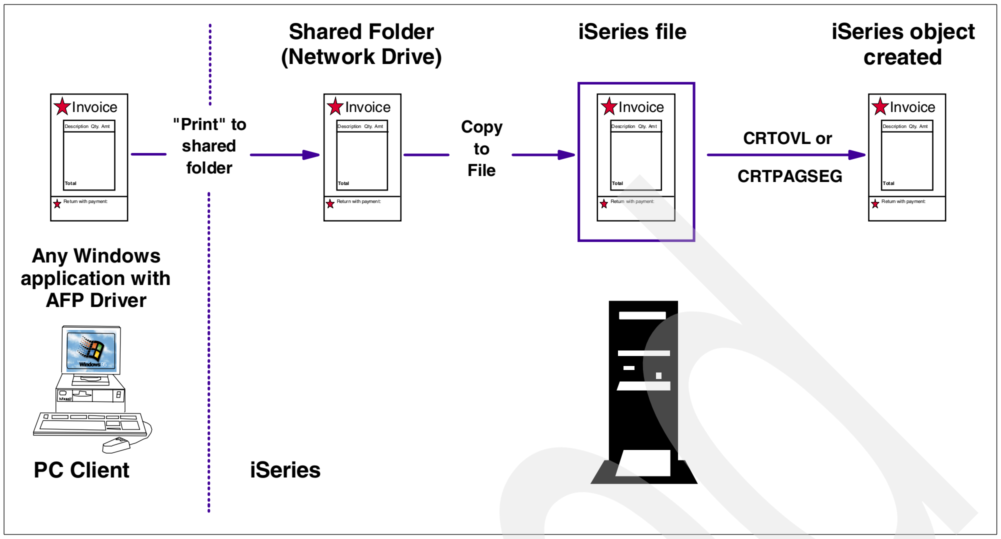


Figure 8. Creating AFP resources using the IBM AFP Driver for Windows

1.2.5.1 When to use the AFP Driver

For page segments, the AFP Driver is useful for taking good quality copies of your company logo or a signature and creating an AFP page segment. However, there is no ability to “clean-up” the image.

For overlays, the AFP Driver allows you to use your favorite graphics design tool to create an overlay and then upload the results to the iSeries server. You need to perform testing to ensure that the overlays printing performance is satisfactory. You must also carefully manage the storage and backup of potentially valuable source overlay material since only the final overlay objects are normally stored on the iSeries.

The AFP Driver process is most useful when using PC functions such as spell-checking and automatic text alignment. A Terms and Conditions overlay is a typical example of using the AFP Driver.

Remember that you can use the AFP Driver to create small, specific AFP resources, for example a page segment of a signature or an overlay that gives a “watermark” effect, in conjunction with a more efficient overlay tool such as Infoprint Designer or AFP Utilities/400. Use the appropriate tool for the appropriate task.

Table 9 outlines the advantages and disadvantages of using the AFP Driver for Windows.

Table 9. Advantages and disadvantages of the AFP Driver for Windows

Advantages	Disadvantages
May be used with any Windows application	Requires setup and management of the various steps in the process such as file transfer
Permits use of advanced design elements such as curved lines and boxes	Backup and storage of the source Windows documents is a separate process to be managed

Table 9. Advantages and disadvantages of the AFP Driver for Windows

Advantages	Disadvantages
Allows use of PC word processing functions such as spell-checking and text alignment	May be difficult to correlate overlay element positions with that of the corresponding variable data
No-charge product	Produces relatively inefficient AFP overlays; complex overlays may print slowly on smaller printers

1.2.5.2 Uploading resources created by the AFP Driver

Figure 8 shows using a mapped network drive with iSeries NetServer, as a means of transferring the AFP resources. Another method is to use File Transfer Protocol (FTP). This is described in the "AFP Driver" chapter in *AS/400 Printing V*, SG24-2160.

A new method is to use the AFP Manager feature within Operations Navigator. This is described in Chapter 9, "AFP Manager" on page 213.

1.3 Driving the printer

In today's world, a customer may have a wide variety of printer types, each supporting a different data stream or language. They may have small laser printers for desktop publishing, workgroup printers for convenience printing, and high speed production printers for company output. At the same time, different applications or software packages generate the output in different ways. To add to the confusion, there may be restrictions on the type of data supported on a given printer depending on the attachment method.

In the past, users were restricted to printing their output only on the specific printer type for which the application was designed. IBM now offers a number of different solutions that allow almost any type of document to be printed on almost any printer.

This section describes the different offerings and tools available from IBM to transform the data from one data stream to another or send it to the printer. Essentially we describe printer drivers for the iSeries.

1.3.1 OS/400 for SCS and IPDS twinaxial printers

The default data steam type generated by iSeries is SNA Character Stream (SCS). This EBCDIC-based data stream has a minimum number of formatting controls in its basic implementation.

iSeries can also produce the Intelligent Printer Data Stream (IPDS). This is a *bi-directional* data stream that provides a two-way conversation between the printer and the iSeries. In the twinaxial environment, without PSF/400, IPDS support includes graphics, printer resident fonts, images, and barcodes. External resources, such as overlays, page segments, or host fonts, are not supported in this environment.

The spooling subsystem included in OS/400 supports sending SCS and a subset of IPDS to the appropriate twinaxial attached printer. SCS spooled files may also be sent to an IPDS printer. Some IPDS spooled files may be sent to SCS printers, but there will likely be some loss of function in the process. This is described in “Redirecting Output” in *iSeries Printer Device Programming*, SC41-5713.

1.3.1.1 When to use OS/400 print support

You use OS/400 print support whenever you connect twinaxial printers to the iSeries, whether they are SCS printer or IPDS printers configured as AFP(*NO). If the latter are configured AFP(*YES), you will use PSF/400 as described in the following section.

1.3.2 Print Services Facility/400

Print Services Facility/400 is an optional, chargeable feature of OS/400, 5722-SS1. It is the driving force behind the Advanced Function Presentation (AFP) architecture on the iSeries. The application data stream of AFP is the AFP data stream (AFPDS). This data stream is independent of the target printer or the system it is being generated on. PSF/400 takes the AFPDS and converts it to the printer specific, or printer-dependent, data stream called Intelligent Printer Data Stream (IPDS). AFPDS is sometimes referred to as *Mixed Object Document Content Architecture (MODCA)*. There are some very slight differences architecturally between the two, but they are functionally the same when it comes to printing on the iSeries.

In this environment, the full range of printer capability is supported, including external resources. PSF/400 carries on a two-way conversation with the device, making sure every page is printed accurately (for example, with fidelity), and with the correct external resources. If there are problems, PSF/400 ensures recovery at the page level.

PSF/400 also accepts spooled files that have been generated in SCS, IPDS, LINE, or AFPDSLINE data stream formats; each time it automatically converts the data stream to IPDS.

What is PSF/400?

You can think of PSF/400 as being the software equivalent of the IPDS hardware function on the printer. In the bi-directional communication, the PSF/400 software talks to the IPDS hardware in the printer.

1.3.2.1 When you must use PSF/400

PSF/400 has a licensing requirement if you are using IPDS printers configured as AFP(*YES) on your iSeries. Since all LAN-attached IPDS printers only function as AFP(*YES), you also need PSF/400 for any LAN-attached IPDS printers.

1.3.2.2 When you should use PSF/400

Print Services Facility/400 provides the highest degree of integration, function, and management for printing on an iSeries. We recommend PSF/400 for any business-critical application where output fidelity and error handling down to the page level is essential.

1.3.2.3 What PSF/400 does

Because PSF/400 is so well integrated into OS/400, it may be difficult to appreciate what it is capable of doing. The following functions are all performed by PSF/400, if required:

- Interrogation of printers to determine their characteristics
- Automatic download and management of resources at the printer
- Enables printing of AFP resources:
  - Barcodes
  - Outline and raster fonts
  - Overlays
  - Page segments
- Tracks jobs to completion; issues detailed error messages if problems occur
- Provides support for page range printing
- Allows printer sharing between other hosts including Windows
- Automatically transforms data streams as required
- Enables advanced finishing options where supported by the printer
- Automatically recovers sessions if a print session is interrupted
- Enables a resource library list for the printer
- Rotates of printed image to cater for edge-sensitive paper (for example, punched)
- Permanently captures host fonts at the printer
- Uses custom font-substitution tables

1.3.2.4 Licensing PSF/400

PSF/400 is an optional component of the Operating System/400 (5722-SS1). The price is based on the speed of the single fastest printer attached to the iSeries. For V5R1, the price breaks are for 1 to 45 impression per minute (ipm), 1 to 100 ipm, or AnySpeed. There is no licensing limit to the number of printers that may be used with PSF/400.

PSF/400 is also included in the ValuPak for AS/400 for V5R1 (5722-VP1). ValuPak consists of the following products:

- Option 36 of OS/400 - PSF/400 1-45 IPM Printer Support
- 5722-PT1 Performance Tools for iSeries - Option 1 Manager feature
- 5722-QU1 Query for iSeries
- 5722-ST1 DB2 Query Manager and SQL Development Kit for iSeries
- 5722-XW1 Client Access Family

Do not confuse ValuPak for AS/400 with the ValuPak for AS/400 Printing. This latter product (5769-PPK) was actually an offering, bundling the APU and PPFA/400 components of AFP PrintSuite for OS/400, AFP Utilities/400, and the AFP Font Collection. ValuPak for AS/400 Printing was withdrawn from marketing at OS/400 V5R1 but is still available for ordering with OS/400 V4R5.

1.3.3 Host Print Transform

Host Print Transform is included with OS/400. It provides a simple and easy way to convert spooled files that are generated in SCS or AFPDS to ASCII. SCS files may be converted to one of several ASCII printer data streams. AFPDS works best with the PPDS or HP-PCL ASCII data streams. AFP resources (such as fonts, overlays, page segments) referenced in AFPDS spooled files are converted into an ASCII printer data stream and passed to the ASCII printer.

A wide variety of preset definitions are available to support many printers, from IBM and many other vendors. Beyond that, you may modify the translation through the use of workstation customization objects, although this is not a trivial task.

There is extensive documentation on Host Print Transform in earlier AS/400 printing redbooks, for example:

- Chapter 7, "Host Print Transform" in *IBM AS/400 Printing III*, GG24-4028
- Chapter 6, "Advanced Host Print Transform Customization" in *IBM AS/400 Printing IV*, GG24-4389
- Chapter 6, "Host Print Transform" in *IBM AS/400 Printing V*, SG24-2160

Host Print Transform can be invoked when you send a spooled file to a printer using a variety of methods, such as an ASCII device description, a remote output queue, and a 5250 printer session. In any of these methods, the default for HPT is TRANSFORM(*YES). You need to select the appropriate Manufacturer Type and Model for the customization, because the iSeries has no way of communicating directly with the printer to establish its characteristics in the way that PSF/400 does.

1.3.3.1 HPT limitations

Host Print Transform has the following limitations or restrictions:

- AFP graphics (GOCA objects) are not supported
- Graphics (GDF) files are not supported
- GDDM fonts are not supported
- CHRSIZ DDS keyword is not supported
- Edge-to-edge printing is supported only by removing all no-print border values
- The FIDELITY printer file keyword is ignored
- AFP to ASCII transform may need to be run in raster mode for better print fidelity, at the risk of poorer performance
- Impact printers are not supported
- Computer output reduction (COR) and multiple-up printing is not supported
- Color barcodes are not supported
- Only 240 and 300-pel images may be transformed

1.3.3.2 Image Print Transform

It has been possible to transform image or PostScript data streams since OS/400 V4R2. The resulting data streams are either AFPDS or ASCII. Specifically the supported input data streams are TIFF, GIF, BMP, and PostScript Level 1; the supported output data streams are AFPDS, PCL and again PostScript Level 1.

The source of the image or PostScript data stream would typically be a Windows PC, but could also be a Network Station, or a file placed in the iSeries Integrated File System (IFS) by some other process.

Examples of where this might be used include the ability to print a customer PostScript application on a high-speed iSeries IPDS printer. This is transformed by the iSeries into AFPDS and then passed to PSF/400 for conversion into IPDS and printing as normal. The difference now is that you have page-level status and error recovery for your PostScript job.

Another example might be TIFF images from an archiving system: these could be printed on a variety of iSeries printers, the OS/400 operating system determining the appropriate transform in each case.

Image Print Transform is described in detail in Chapter 7 in *AS/400 Printing V*, SG24-2160. As Section 7.8, "Troubleshooting" in the same book implies, it is an early invocation of Image Print Transforms within the operating system. For example, only PostScript Level 1 is supported, and there is very limited control over both the positioning of converted images and the speed of conversion. For production use of image or data streams, that is, involving business-critical printing, you must purchase the Infoprint Server for iSeries program product.

1.3.4 Infoprint Server for iSeries

Infoprint Server (5722-IP1) contains many additional data stream transforms, several of which can be used to drive printers. The product is described in detail in several chapters of this book, but the following sections provide an overview.

1.3.4.1 PDF transform

Your print applications that create AFP spooled files for printing on mission-critical IPDS system printers can also generate industry-standard Adobe Portable Document Format (PDF) files. These can be stored in the iSeries Integrated File System (IFS) for later processing, printed directly on a PDF-capable printer (such as the Infoprint 21 or 70), or automatically sent as an e-mail.

The PDF produced by the transform is actual text (not an image), so the resulting PDF file is efficient, in terms of file size, and enables use of standard Adobe Acrobat Reader functions such as text search, replace, and copy/paste.

Since the transform to PDF takes place on the IPDS spooled file that Print Services Facility/400 (PSF/400) creates, the initial spooled file can be anything that can normally be printed on an IPDS printer, such as SCS, IPDS, AFPDS, Line data (externally formatted with page definitions and form definition), mixed data (line data mixed with AFP), or output from OfficeVision (OV/400).

Further to this, an AFP spooled file may be segmented into multiple PDF files. This is likely to be of use with e-mail, for example one customer statement per e-mail from a statement run of many hundreds or thousands of customers.

If you want to break the spooled file into multiple PDFs or multiple e-mails, you need to have special AFPDS structured fields inserted into your data stream. This can be done in one of two ways:

- Use the DDS keywords STRPAGGRP and ENDPAGGRP in your application and generate an AFPDS spooled file directly.
- Use the indexing feature of the CRTAFPDTA command. This requires line data as input. For more information, see Chapter 5, “Infoprint Server for iSeries: CRTAFPDTA” on page 149.

1.3.4.2 Creating AFP Data (from line data)

The CRTAFPDTA command included with Infoprint Server allows you to convert a line data spooled file (or one containing mixed line and AFP data) to an AFP file. This is a fully-resolved AFP file. For example, all the externally-referenced AFP resources are built into the AFP file. Optionally, the file may be indexed.

The significance of this is that the fully-resolved AFP spooled file may now be:

- Viewed using the AFP Viewer within Client Access or the AFP Workbench
- Sent to another system for printing, storing, or viewing

- Processed by Host Print Transform, converted to ASCII, and sent to a non-IPDS printer
- Stored in a document archival system, such as iSeries Content Manager OnDemand

All this can be done without needing to send or manage the separate AFP resources.

The indices may be used to break up the file into multiple PDF files, which may be e-mailed to different destinations.

1.3.4.3 Converting PCL, PostScript, or PDF to AFPDS

Infoprint Server can convert several ASCII data streams to AFPDS, for example PCL, PDF, and PostScript (levels 1 to 3). Using OS/400 NetServer (described in "NetServer" on page 27), for example, we can print directly to an iSeries-attached IPDS system printer using an ASCII printer driver. Previously you would have needed the AFP Driver for Windows to generate AFP from the Windows application. More likely, you might have an application generating PostScript or PDF. You can place this output on the OS/400 spool and have PSF/400 takeover the printing. This is the same scenario as portrayed in 1.3.3.2, "Image Print Transform" on page 23, but with enhanced versions of the transforms (for example, PDF and PostScript levels 2 and 3).

1.3.4.4 Converting PC image formats to AFPDS or PostScript

These transforms run on a Windows PC and transform the images into AFP overlays or page segments, or into PostScript Level 2. The supported images are TIFF, GIF, and JPEG.

1.3.5 Convert Line Data (CVTLINDTA)

This is a new parameter available in the Create, Change, or Override Printer File (CRTPRTF, CHGPRTF, or OVRPRTF) commands with Version 5 Release 1. It takes printer files that accept line data as input, applies a Page Definition to it, and generates a spooled file that is in fully-resolved AFPDS format.

By "fully-resolved", we mean that the instructions for placing the data on the page have been embedded with the text. The page definition is no longer needed. External AFP resources, such as fonts, overlays, page segments, remain external to the file and must be managed separately if moving the files from one system to another. If you want to include the external resources with the spooled file, you need to use the CRTAFPDTA command that part of the Infoprint Server product. See 1.3.4.2, "Creating AFP Data (from line data)" on page 24, for general information, or 3.2.1, "CVTLINDTA" on page 96, for details.

This parameter is part of OS/400. It was provided to compensate for some of the limitations of spooled files generated as DEVTYPE(*LINE) data:

- Cannot use the IBM AFP Viewer with these spooled files
- Cannot send these spooled files to a printer configured to use Host Print Transform

The resolved AFPDS output generated when using CVTLINDTA overcomes these limitations.

1.3.6 SAP R/3 AFP PrintSuite feature

SAP R/3 has become a popular choice for medium to large companies that require an integrated software product to provide basic business automation. The R/3 client/server application provides businesses with a series of integrated modules that span the major functions of manufacturing, finance, sales, distribution, and human resources.

SAP R/3 normally generates output in two formats:

- **ABAP**: This is a classical computer listing that contains fixed characters, line positions, and fixed fonts, such as Courier. R/3 uses this format whenever you request a report. Some R/3 applications use report printing for outputting forms.
- **SAPscript***:* This is the formatting program for word processing. SAPscript processing creates a data stream in Output Text Format (OTF). SAPscript is a tag-oriented language that contains records with the print options, fixed text, and variable data placed on the page by a particular printer. SAPscript is used in most of the commercial applications that comprise R/3, such as invoices, bills, reminders, and paychecks. Usually, SAPscript uses typographic fonts such as Times Roman or Helvetica.

The SAP R/3 product includes device drivers for the iSeries to convert the output to SCS, PCL, or PostScript. The SAP R/3 feature of AFP Printsuite (5798-AF3) extends this print support to enable users to access the full range of IPDS printers. This provides them with the intelligent error recovery not found on ASCII printers and, therefore, the ability to perform business-critical printing.

This added support is provided through the Convert Print Data (CVTPRTDTA) command. CVTPRTDTA provides a direct transform of SAP R/3 print data into AFPDS or LINE data. It may be run from the command line or automatically called from within an R/3 spool process.

The CVTPRTDTA function allows you to add form definitions (with overlays) images, fonts, and barcodes to the output. You may choose to add a page definition to further format the ABAP report output. A number of page definitions is provided with the package, or you may choose to make your own using PPFA/400 or Infoprint Designer.

PSF/400 and the AFP Font Collection are required to print the output produced by the SAP R/3 AFP PrintSuite feature on an IPDS printer.

For more information, see: `http://www.printers.ibm.com/r5psc.nsf/web/fssuite3`

1.4 Managing the printed output

This section examines some of the software available for managing and controlling your printed (and non-printed) output.

1.4.1 OS/400 spool support

The spooling support in OS/400 is taken rather for granted. This is especially true when you compare the options available for manipulating a spooled file in Windows. OS/400 users can change, hold, save, redirect, and delete their spooled files, subject to security status. In most cases, users may select a range of pages to be printed. They can view the spooled files in text mode on a "green screen" or with full fidelity using the AFP Viewer within Client Access. Much of the above applies also to PC-sourced files, printed via the iSeries. Again compare this with the support offered by a Windows print server.

NetServer

The NetServer function was introduced as part of OS/400 V4R2. It provides Windows Network Neighborhood support for print and file serving on the iSeries server. This means that you can access iSeries printers and files in the Integrated File System from the Windows desktop. A product such as Client Access, described below, is not required, although it does offer enhancements to the file/print operations.

For more information, see: `http://www.as400.ibm.com/netserver/index.htm`

1.4.2 PSF/400 spool support

OS/400 twinax devices provide some feedback when a problem occurs, such as a ribbon check, open cover, or paper out. IPDS printers, whether attached over twinax or the LAN, SNA, or TCP/IP, provide an even higher degree of feedback and error recovery. However, they require (in most cases) the complementary software on the iSeries server called Print Services Facility/400 (PSF/400), which is an optional charged feature of OS/400.

Do not look too hard for a PSF/400 front-end menu. The real secret of PSF/400 is that it is *system-managed* printing. Most of the parameters to change printer and printing behavior are found in the user-created PSF configuration objects. These are optionally referenced (for example, invoked) in a printer device description.

PSF/400 typically delivers detailed status or error messages to the print writer job logs (of which there are two for an AFP=*YES printer) or the system operator message queue.

For a reminder of many of the functions that PSF/400 performs, refer to 1.3.2.3, "What PSF/400 does" on page 21. See also the Web site at:
`http://www.printers.ibm.com/r5psc.nsf/web/4psf400`

1.4.3 Client Access/400

Client Access/400 is a Windows product for connecting your PC to one or more iSeries servers. It provides such functions as 5250 terminal emulation, file transfer, and database access. The full-function version of the product is known as iSeries Client Access for Windows 95/NT (5763-XD1), but the strategic new version of the product is iSeries Client Access Express for Windows (5769-XE1). Both clients are available with the iSeries Client Access Family for Windows product (5769-XW1).

1.4.3.1 Operations Navigator

Operations Navigator is part of Client Access, whether the original full version or the later Express client. Among the many possibilities, it offers a GUI view of your printing operations. The most notable feature is the integration of the AFP Viewer product. Simply double-click a spooled file from a list, and the AFP Viewer is invoked. AFP resources are fetched automatically from the iSeries and cached locally on the PC for easy viewing.

Using Operations Navigator to implement online viewing

This is actually a "free" way to implement online viewing, rather than printing in an enterprise. Let's consider an example. Daily management reports are printed overnight and delivered to a user's desk in the morning, perhaps several hundred pages of information, printed on an impact printer, using "music-rule" stationery. Sometimes the printed output sits on the desk all day, until discarded – unused – at night.

Alternatively the same job *could* be delivered to the user's output queue, rather than directly to the printer. The user would double-click the spooled file, using Operations Navigator, obtain the information they need, and perhaps copy and paste it into a spreadsheet or word processing document. The user could also select one or more pages to be printed locally, on the nearby Windows inkjet, or laser printer, even if the original spooled file was in AFP format (the Viewer converts the data into PCL or PostScript for local printing). Finally, the user could, if desired, drag and drop the spooled file onto the traditional impact printer queue and print it "on demand" (as required). The difference here is that you give users the choice to be more productive, use less paper, and therefore, keep costs down.

1.4.3.2 AFP Manager

AFP Manager is an install option with Operations Navigator under Client Access Express that is new with Version 5 Release 1. The component provides three main functions:

- Easy import of AFP resources created on the PC to the iSeries
- Creation and management of PSF configuration objects
- Creation and management of font substitution tables

AFP Manager is described in detail in Chapter 9, "AFP Manager" on page 213.

For more information about Client Access, see:

`http://www-1.ibm.com/servers/eserver/iseries/clientaccess`

1.4.4 AFP Viewer

This very useful utility has been described several times already in this chapter. The following list shows which products ship the AFP Viewer:

- AFP Workbench for Windows
- iSeries Client Access for Windows 95/NT
- iSeries Client Access Express for Windows
- Content Manager for iSeries

The list is actually more extensive when you consider that products, such as AFP Utilities/400 and Facsimile Support/400, can invoke the AFP Viewer if it is already installed on the client PC.

For most iSeries customers, the AFP Viewer provided with one of the Client Access products provides all the functionality they need. The AFP Workbench product is intended for cross-platform access to AFP documents, but may also be useful if Client Access is not in use at a customer site. The AFP Viewer shipped with various components of Content Manager is usually invoked while using those particular products.

The AFP Viewer can view:

- AFP documents
- AFP overlays
- AFP page segments
- PC file formats such as TIFF and PCX

Indexed AFP documents are particularly effective when used with the AFP Viewer, in terms of speed and ease of navigation. Documents may be viewed in “n-up” or rotated views and printed locally with full fidelity on a local Windows printer. The option of displaying image data in a different color to text is useful when creating efficient AFP resources such as overlays and page segments.

1.4.4.1 AFP Viewer plug-in

Another means of using the AFP Viewer is to obtain the AFP Viewer Web browser “plug-in”. You can download this from:

`http://www.printers.ibm.com/R5PSC.NSF/Web/afp_viewer_plug-in`

This is free of charge but supplied “as is”. It auto-detects AFP documents and adds the appropriate navigation icons to the browser toolbar. Full details are supplied in the IBM AFP Viewer Technical Reference, which is available at the Web site above and downloaded along with the plug-in. The plug-in works with Internet Explorer and Netscape Navigator, from version 3.01 and higher (but not with Internet Explorer 6.0).

1.4.5 Infoprint Manager for AIX and for Windows NT/2000

The IBM Infoprint Manager product family can potentially deliver a great deal of additional function to the iSeries server, for example:

- Enterprise-wide printer management
- Intelligent document routing and scheduling
- Multi-host printer usage
- Job notification and accounting for all printers in the enterprise

The main concern for an iSeries system administrator is how to dispatch print output to one of these external print servers. This topic is covered in detail in Appendix H, “AS/400 to AIX printing” of *AS/400 Printing V*, SG24-2160. Much of what is described can also be applied to the Windows NT or 2000 environment.

For more information, see:

`http://www.printers.ibm.com/R5PSC.NSF/Web/ipmgrfamilyhome`

1.5 Complementary IBM offerings

By complementary offerings, we mean applications that complement, use, or at the very least, are compatible with IBM printing products, in particular those that use AFP printing. Read this section if you are wondering how to further develop your AFP print applications into other forms of output presentation, such as fax, viewing, archiving, e-mail, and Web-enabling.

1.5.1 Facsimile Support/400

Facsimile Support/400 V4R4 (5798-TBY) and Facsimile Support for iSeries V5R1 (5798-FAX) are an iSeries-based inbound and outbound fax product. Links with the AFP print model include:

- Using iSeries IPDS and other printers
- Using AFP overlays and page segments with header sheets
- Spooled files may be faxed instead of or as well as printed
- The Client Access AFP Viewer can be used to view inbound faxes

For more information, see: `http://www.ibm.com/servers/eserver/iseries/fax400`

1.5.2 Content Manager OnDemand for iSeries

This product has been known by a variety of names. It began life as Report/Data Archive and Retrieval System (R/DARS). It then became EDMSuite OnDemand. The current full name is Content Manager OnDemand for iSeries (5722-RD1). For now, we refer to it simply as “OnDemand”.

OnDemand is a high-volume archive and retrieval product. Such products are often known by other names, such as Computer Output to Laser Disk (COLD). But in fact, the storage media used by OnDemand could be iSeries disk, CD-ROM, tape, or any other accessible iSeries media. Similarly, another industry term is Enterprise Report Management (ERM), but with OnDemand, many objects may be stored, such as spooled files (including AFP spooled files), e-mail, images, and OS/400 objects. These may be indexed (if not already AFP-indexed) and stored in the OnDemand repository. They may then later be searched for, retrieved, and possibly viewed (using the AFP Viewer), and therefore re-enter the output presentation cycle of print, fax, e-mail, and so on.

1.5.2.1 Content Manager Common Server

Although OnDemand for iSeries is specifically architected for the iSeries server, some customers may also be interested in using a more common or generic archive tool. The Common Server product from the AIX and Windows NT platform has been ported to the iSeries platform. The resulting product is iSeries Content Manager Common Server, included with the native OnDemand for iSeries product (5769-RD1). The Common Server provides the tools for complete and secure capture, storage, retrieval, and viewing of all your documents. It incorporates the Spool File Archive (SFA) and AnyStore capabilities available in the parent OnDemand for iSeries product.

Some of the new, unique features supported by Common Server are:

- A larger number (up to 32) of more flexible, longer index keys
- Improved searching tools
- Optional support for indexing PDF documents, in addition to existing support for line data, SCS, and AFPDS documents
- Unique views by user; viewing can be done across folders
- Full text search, if desired
- Higher compression ratios for archived data

For more information, see:
`http://www.ibm.com/software/data/cm/cmgr/about_cm400.html`

1.5.3 Content Manager for iSeries

This product has also been known by several different product names. Separate products such as ImagePlus, VisualInfo, and Workfolder Application Facility/400 were brought together under the EDMSuite banner. Today they are functions within the Content Manager for iSeries 5.1 product (5722-VI1). This is a *content and object management* product that is used for storing all the miscellaneous pieces of information a company holds, from images to word-processed documents to multimedia.

While OnDemand/400 concentrates more on storing company-generated documentation, Content Manager is more likely to be used for document imaging and workflow, implying inbound documents. This is an over-simplification of the roles of the two products but suffices in this instance. It is important to note that if the two products are in use in an organization, they can share a common repository of data, accessed by a common front-end (probably one of the products' clients).

For more information, see Chapter 16, "Using Content Manager OnDemand for iSeries" on page 311, and the IBM Content Manager OnDemand product family site at: `http://www.ibm.com/software/data/ondemand`

1.5.4 Backup Recovery and Media Services

Backup Recovery and Media Services (BRMS) for iSeries (5722-BR1) provides automated data backup and retrieval for the iSeries server. From a printing point of view, of interest is the ability to save spooled files, output queues, and any external AFP resources associated with AFP spooled files.

For more information, see:
`http://www-1.ibm.com/servers/eserver/iseries/service/brms.htm`

1.6 Complementary non-IBM offerings

Some customer's print formatting requirements may lie outside the scope of the various products described above. This section describes a few of those types of products.

1.6.1 Office Vision/400 replacements

With the proliferation of easy-to-use word processing packages available on the workstation, fewer customer use the 5250-based Office Vision/400 product as an editor for straight document creation. However, many customers used the powerful ability of OV/400 to easily merge their iSeries database records to create customized documents. In some cases, mission-critical applications have been built up over the years around these functions. As of Version 5 Release 1, you can no longer run OV/400 on your iSeries. The redbook *How to Replace OfficeVision/400 in Your Applications: Looking at Domino for AS/400 and AS/400 Alternatives*, SG24-5406, addresses this topic.

Lotus Domino provides a possible replacement for the simple document creation, mail, and calendaring functions of OV/400. Other products from Lotus and other vendors can be used to create new word processing documents. However, most other word processing packages do not provide an integrated way of merging iSeries database information into text documents.

Three vendors are described in the OV/400 replacement redbook:

- Inventive Designers' DTM for AS/400
- Aia's Intelligent Text Processing/Client Server (ITP/CS)
- Triangle's Data/Text Merge for Notes/Domino

This list is not exhaustive. Other IBM Business Partners and independent software vendors provide solutions to replace OfficeVision/400 services or APIs in an iSeries environment.

For additional information, see the summary in Section 3.3 of *How to Replace OfficeVision/400 in Your Applications: Looking at Domino for AS/400 and AS/400 Alternatives*, SG24-5406, or Appendices J, K, and L for details on each product.

A useful Web site for considering OV/400 migration is:
`http://www.as400.ibm.com/developer/ov400/index.html`

Chapter 2. Advanced use of IBM Infoprint Designer for iSeries

IBM Infoprint Designer for iSeries provides a fully-graphical document composition interface to the iSeries and AS/400 printing system. It supports the requirements of today's complex documents and reports, producing fully electronic documents combining data, text, electronic forms, graphics, image, barcoding, and typographic fonts. Infoprint Designer can be used for the design of new output applications or the re-engineering of existing applications.

Infoprint Designer consists of three components:

- **Overlay Editor**: Designs overlays (electronic forms) to be used in the print application
- **Image Editor**: Designs images to be used in the print application
- **Layout Editor:** Puts all the design components together into the final electronic document

With Infoprint Designer, you can build complex electronic print applications easily. It manages the upload of the resources you design to the iSeries server. Then, once the printer file is enabled for use with Infoprint Designer resources, the original application is automatically transformed with your new design and layout.

2.1 Getting started with Infoprint Designer

The manual *IBM Infoprint Designer for iSeries: Getting Started*, G544-5773, is the primary reference for use with this product. It covers such topics as product prerequisites, installation, and the basic tasks within each of the three components. It also describes the formatting required to produce more complex applications, such as an "either/or" page layout, for example, whether to print the final (totals) page of an invoice or to print the standard full-page item listing page within the invoice. The manual then describes how to implement the new design and layout with the original print application.

This chapter covers more advanced uses of Infoprint Designer. You should read and thoroughly understand the *Getting Started* manual before using the information in this chapter.

2.2 Setting up the PC libraries

You may need to tell Infoprint Designer the name and location of the "libraries" where Infoprint Designer can find the AFP resources you use or create. These are actually Windows directories on your PC. Infoprint Designer already has the following default locations for:

- AFP overlays
- AFP font resources (coded fonts, character sets, code pages)
- Images (AFP page segments, PC TIFF files)

The paths for the above resources are stored in a file called the *library profile*, with an extension of LBP. Infoprint Designer creates the following two library profiles in the \DESIGNER\USERISIS directory during the installation process:

- **DEFAULT.LBP**: This is the default library profile for use with Infoprint Designer. The fonts directory path specifies AFP outline fonts.

- **RASTER.LBP**: This alternative library profile specifies a fonts directory containing AFP raster (bitmap) fonts.

AFP outline fonts are resolution-independent scalable fonts that provide an almost unlimited range of point sizes. However, your AFP printer must be capable of receiving such fonts. If your printer does not accept them, or you are using an ASCII printer with Host Print Transform, normal font substitution will occur (typically substitution of printer-resident for host-resident fonts).

Both library profiles specify default paths as shown in Table 10. This table also shows the default PC file suffix used to denote a resource type.

To check your library profile, select **Defaults-> Libraries** from the main menu in Infoprint Designer. You cannot change it from here; instead use **Edit-> Change library**. However, avoid doing this in the middle of a project. The preferred way to switch to a new library profile is to select **File-> New project** and browse for a different library profile. See Figure 9 on page 36.

Table 10. Infoprint Designer default AFP resource locations

Default Infoprint Designer location C:\DESIGNER	Type of AFP resources stored	Default PC file suffix of AFP resource
\OGL	Overlay source files	OGL
\OVL240	240-pel overlay objects	OVL
\OVL300	300-pel overlay objects	OVL
\OVL600	600-pel overlay objects	OVL
\FONTS\CODEPAGE	Font code pages	none
\FONTS\OUTLINE	Outline fonts	OLN
\FONT\RASTER	240-pel and 300-pel raster fonts	240, 300
\PSEG240	240-pel page segments	240
\PSEG300	300-pel page segments	300
\PSEG600	600-pel page segments	600
\TIFF	PC tagged image format (TIFF) files	TIF
\PPFA	Page and form definition source files	PFA

The location for page and form definition source is not defined in the library profile, but defaults to \DESIGNER\PPFA.

It is a good idea to define your own libraries for storing self-created resources such as overlays and page segments. Infoprint Designer uses common suffixes such as 240, 300, and 600 to distinguish between different resolutions of fonts. These values are also used for convenience with page segments, but it is the directory location that Infoprint Designer needs (PSEG240, PSEG300, and PSEG600).

Other AFP applications, such as the AFP Viewer, use the suffixes outlined in Table 11.

Table 11. Common AFP resource suffixes

AFP resource	Suffix
Overlay	OLY
Page segment	PSG
Form definition	FDE
Page definition	PDE

You should be aware of the resolution of the resources you will use. The following AFP resources are resolution-dependent (for example, will be either 240, 300, or 600-pel):

- Overlay objects
- Page segments
- Raster font character sets
- Raster coded fonts

Typically they are 300 dpi. All other AFP resources are resolution-independent.

2.3 Starting a project

Infoprint Designer uses the concept of a *project*. There are two types of projects, overlay projects and layout projects, as explained in the following sections.

2.3.1 Overlay projects

Use overlay projects if you are using Infoprint Designer to create overlays that will be used with other formatting solutions, such as DDS or Advanced Print Utility. You may also use overlay projects to work on overlays that are used in many different layout projects, or just to have a clean slate while the initial design work is being done. The project file, with the suffix *.prj, for an overlay project contains the name of the overlays and a pointer to the default library profile document.

Figure 9 shows an example of the panel you see when you start a new overlay definition (that is, an overlay project). The default name of the project is UNTITL (for “untitled”). Do not save it under this name; you should change it *immediately* to a meaningful name of your own choice. When you change the project name, it also automatically changes the name of the overlay file. You may add additional existing overlays to your project when you start it by clicking the **Append** button.

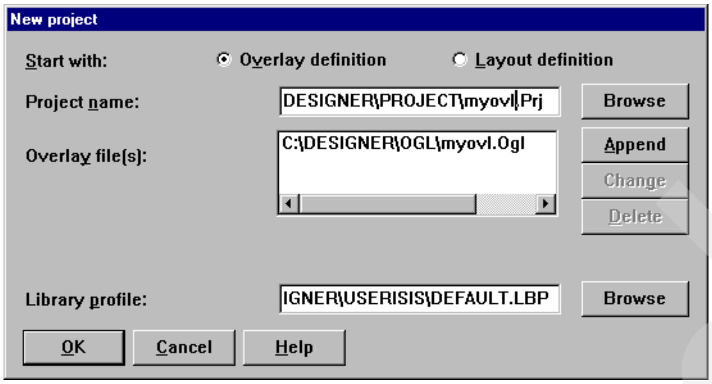


Figure 9. Starting an overlay project

2.3.2 Layout project

When you start mapping application data onto your overlay, you need to switch to using the Layout Editor (see 2.7, "Layout Editor" on page 44). Your work here is saved into a layout project. You can also start directly with a layout project. As before, you are not permitted to save the file under the default name of UNTITL, so change the name of the project to a meaningful name of your choice. When you change the name of the project, it automatically changes the name of the layout file. This name becomes the name of the AFP page and form definition objects that are created. (See 2.8, "Introduction to form and page definitions" on page 51.)

At this time, you may also choose to associate a line data file from either the AS/400 or your PC to your project. This is the sample file of data you work with and is described in 2.7.1, "Selecting sample data from the iSeries server" on page 44.

Figure 10 shows an example of the New project options for a layout project.

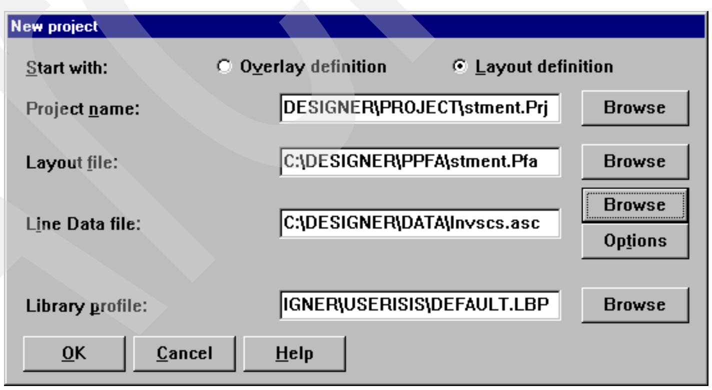


Figure 10. Starting a layout project

The project file for a layout project contains references to the sample spooled file used to create the project along with the name of the layout source file and a pointer to the library profile.

Typically, an Infoprint Designer project corresponds to your application, for example Invoices, Purchase Orders, Statements, and so on. Depending on the design of your application, however, a project might encompass two or more applications. For example, you might include the ability to determine whether an invoice is actually a credit note and invoke the appropriate overlay and page layout.

2.4 Designing and creating an overlay

By default, Infoprint Designer begins with a screen labeled "View/Edit". This is your workspace. The reason for the screen title is that if you are editing your page layout, you can also view the overlay "behind" the data and vice-versa. See Figure 11.

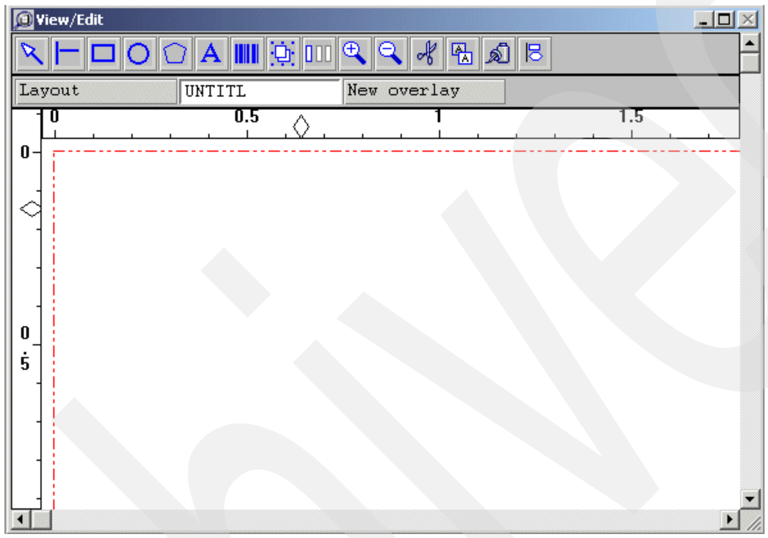


Figure 11. Initial Infoprint Designer View/Edit display

Notice that the selected (middle) tab is called "UNTITL", an untitled overlay. We also have the opportunity to start work on another overlay by clicking the *New overlay* tab, or begin work on the document layout (for example, mapped data and overlays) by clicking the *Layout* tab.

2.4.1 Saving your work

Infoprint Designer auto-saves your project file by default every two minutes. You can adjust this by selecting **Edit-> Preferences-> General**. You are also prompted to save the overlay and layout source separately if you attempt to close or quit the current project. The prompt asks if you will save an "ogl" file. OGL stands for Overlay Generation Language, an IBM mainframe program product used for creating overlays. Note that you *cannot* upload and use the OGL source file independently from Infoprint Designer; the program license is for use only as an iSeries product. However the actual AFP resource objects you create – overlays, page segments, etc. – may be transferred to and used with other AFP systems.

It is necessary only to save the Project file (this saves the layout and overlay definitions), but you may find it useful to save the individual component files separately for use with another project. For example, your Statement project file may contain two overlay files (such as Stmnt.ogl and Terms.ogl). The Terms and Conditions overlay (Terms.ogl) might be used with another application, for example, another project. You might choose to store it under a different name or in a different location.

The *Save* and *Save As* panels are similar to the panels shown in Figure 9 on page 36 and Figure 10 on page 36.

To save your work, click **File-> Save project as...** Then highlight the Overlay line and click **Save as**. Type in a new overlay file name. Then navigate to a preferred path if necessary and click **Save**. If the path and file name are already correct, clicking Save simply updates the overlay file with your latest changes to the overlay.

2.4.2 File names

Even though Windows and Infoprint Designer permit long file names for project names, we advise that you save the various files using a 8-character maximum. This is enforced for the save of overlay source files in any case. Layout files are truncated to eight characters on upload to the iSeries server (for example, NEWSTATEMENTS.PFA is truncated to NEWSTATE).

If you are producing AFP resources for use with other non-iSeries servers, you might restrict filenames to six characters. The reason for this is because on some IBM platforms, AFP source code is compiled to produce resources beginning with specific prefixes, namely P1 for a page definition, F1 for a form definition, O1 for an overlay, and S1 for a page segment. For example, a layout file NEWSTA would produce page and form definition objects called P1NEWSTA and F1NEWSTA.

Unless you know that you will use your overlay and page segment resources with a product such as PPFA/400 (which does use the prefixed naming conventions), or have previously adopted that naming convention, we recommend that you *do not use* the prefixed naming conventions. To do this, click **Edit-> Preferences-> AS/400** and ensure that the four Prefix check boxes are deselected as shown in Figure 12. This is the default setting, which means your resources will not be prefixed with “O1”, “F1”, and so on.

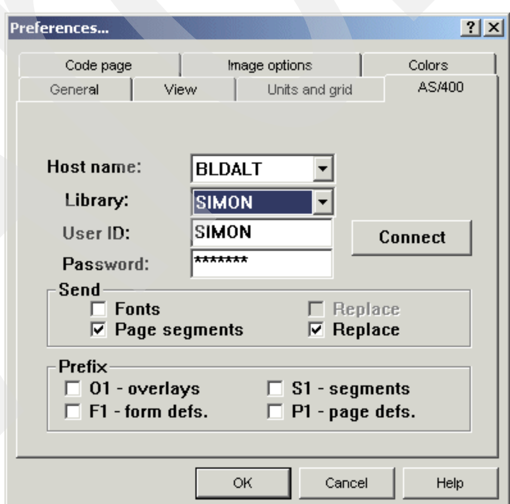


Figure 12. Recommended configuration for resource names and send options

A last point on filenames is that a Project file (*.PRJ) *can* have a long file name, such as “Project to construct new statements application.pfa”. This name is retained in Windows, although only the first 43 characters are displayed in the title bar of the Infoprint Designer window and you lose the *.prj suffix and the selected host printer resolution (for example 300 ppi). To keep this information on display, restrict your project names to 29 characters in length, or use the same name as that of the Layout and Overlay definitions.

2.4.3 Importing existing electronic overlays

If you have existing AFP overlays, you can convert these for use directly with Infoprint Designer. The procedure first depends on which method you use to get the overlay file to your PC.

2.4.3.1 Using file transfer

You need to use the CVTOVLPFM command provided with AFP Utilities/400 (5769-AF1 or 5722-AF1) to convert the overlay object into a physical file member then transfer the file to the PC, using a suitable method such as FTP or Client Access/400 file transfer, in binary in both cases. (You cannot use the original AFP Utilities/400 source file.)

2.4.3.2 Using a mapped network drive

With this method, use the OS/400 CL command CPYTOPCD to copy the overlay object to a suitable folder on the iSeries, for example:

```
CPYTOPCD FROMFILE(SIMON/IPD) TOFLR(DESIGNER) FROMMBR(INVOICE)
TODOC(INVOICE.OLY) REPLACE(*NO) TRNTBL(*NONE)
```

Now create a share to this folder using Windows (go to **My Computer-> Tools-> Map Network Drive**, enter a drive letter and the path to the folder, for example \\MYAS400\DESIGNER). You can then copy the overlay object to your PC or access it directly from the network drive.

2.4.3.3 Using AFP Manager

AFP Manager is an addition to the Operations Navigator interface. It is described in Chapter 9, “AFP Manager” on page 213.

2.4.3.4 Using a Java program

Once you have the AFP overlay file on your PC, select **File-> Get overlay** from within Infoprint Designer. The suffix is significant only for filtering the display of available files. See Figure 13. Before allowing you to work with the overlay, Infoprint Designer prompts you to save it as an *.OGL source file. Infoprint Designer then imports the overlay into your current overlay editor screen, where you may work with it as normal.

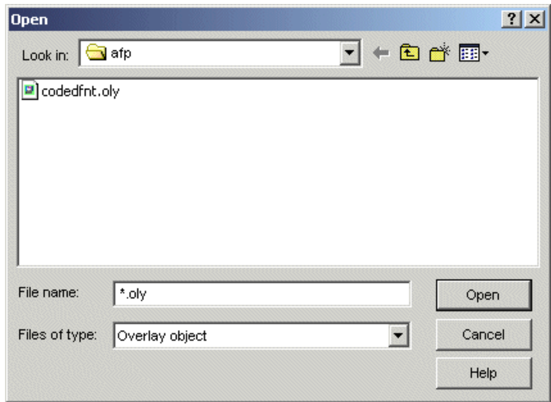


Figure 13. Importing an AFP overlay object

An alternative method is shown in the Java program in Appendix C, “Extracting AFP resource contents” on page 331.

2.4.3.5 Limitations

This method works best with simple overlays of lines and boxes. Text that was originally separate items may now be in a large group of words. This is unavoidable because of the way the text is stored in the AFP overlay. In addition, although overlays created with a font character set and code pages may be imported, these fonts are mapped to coded fonts with code page 500, which may not provide the results you want. In the worst case, you may have to delete or recreate just the text blocks if changes are needed. Also note that page segments referenced in the overlay need to be brought down separately and stored in the appropriate page segment libraries identified in the Designer’s library profile.

If you previously created an overlay using Infoprint Designer, you have the overlay source saved as a *.OGL file (typically in the OGL directory of your Infoprint Designer installation directory). You can also open this for use with a new design by selecting **File-> Get overlay** as before. Figure 14 shows the results of opening an existing overlay called INVOIC.

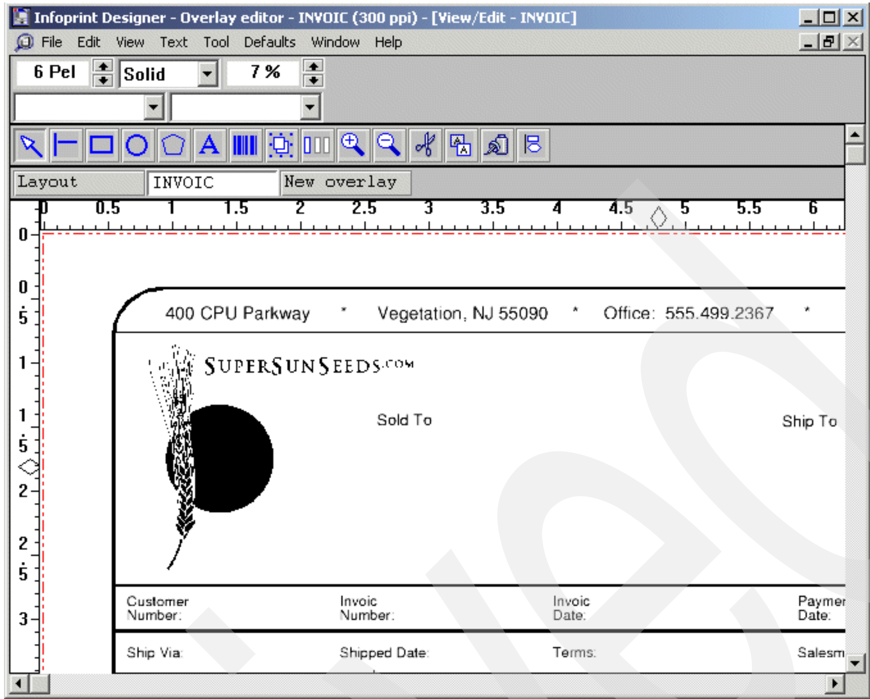


Figure 14. Opening an existing Infoprint Designer overlay

You can delete an overlay or overlays from your current project by using **Edit-> Overlay list** to select the appropriate overlay and click **Delete**. Note that you can also swap out the overlay for another one, by using the Change button in the same menu dialog.

2.4.4 Converting pre-printed forms

Another approach to the above method is to make a good-quality scan of a printed copy of your existing AFP overlay, or even use an existing pre-printed form. Set up your scanning program with the following options:

- TIFF format
- Bi-level (black and white)
- Resolution to match that of your target printer (probably 300 dpi, but possibly also 240 or 600 dpi)
- Scan the image at actual size (1:1)

Infoprint Designer refers to a scanned image of this type as a *Template*. To import it, select **File-> Get template**. Browse for the TIFF image on your PC and select it. The scanned image appears as a non-editable “background” image. You can now trace over the existing elements (boxes, lines or text) to create your new overlay and add new elements. The template will not print or appear with your new overlay. When you have finished using the template, you can select **File-> Get template** and then click the **Delete Template** button.

You can also use overlay source files and overlay objects by selecting *Get template*. The overlay is presented as a grayed-out image as before. The effect is the same as if you loaded (*Get overlay*) an overlay into the project. Since the template overlay will not be saved with the project, this approach may be easier if your project contains many overlays and you need another for reference or for tracing.

2.5 Miscellaneous hints and tips

Most of these tips apply to the Image and Layout Editors, but you are perhaps more likely to use them while constructing overlays.

2.5.1 Using the grid

You can create very accurate overlays with the assistance of the grid feature. Select **Edit-> Preferences-> Units and Grid**. Then set up the grid to your preferences. Notice that this only covers the grid dimensions and characteristics. Use **View-> Grid** to enable the display of the grid (or Ctrl-G) and **View-Snap to Grid** to enable grid snapping. The latter term refers to the automatic alignment of your overlay or page elements to the grid. For example, if you draw a box beginning at 0.48” across the page and grid snapping is on with a grid at 0.5” increments, the box will left-align at 0.5”. We recommend you keep grid snapping on and only occasionally switch it off for fine adjustment of individual elements.

2.5.2 Using the right mouse button

Use the right mouse button frequently. For example, with no elements selected, selecting the right mouse button in the View/Edit window brings up a menu that allows you to control zoom in/out, the ability to fit your view to the page width or height, and the Refresh facility. The latter refreshes your display, which is sometimes necessary when you have just drawn a more complex series of elements. How often you need to use Refresh depends on the performance of your PC and its graphics adapter.

If you do have an element selected, using the right mouse button brings up all the parameters that may be edited for that element. The equivalent menu bar function is to use **Edit->Object properties**. The keyboard shortcut is Ctrl-E.

2.5.3 Using the Shift key

Using the Shift key permits multiple elements to be selected. For example, select several text fields in a document. Select **Edit-> Object Properties**. A small box with the options COLOR, TEXT PAR, and ALIGN appear. The selected change will be applied to all three elements.

2.5.4 Using the Ctrl key

Using the Ctrl key while using the left mouse button often invokes additional functions. For example, for the “tear-off” function, select an element, hold down the Ctrl key, and drag a copy of that element elsewhere on the page.

In Layout mode, the Ctrl key is useful for adjusting the line spacing of all fields in a repeated printline. For example, select any but the first field, hold down the Ctrl key and click and drag the fields to a new line spacing interval.

2.6 Using the Image Editor

Most users of Infoprint Designer require image data in their application. Examples of this could be a corporate logo, a signature, or an industry membership sign. Generally these exist somewhere in the company in one of the common PC formats such as BMP, GIF, JPEG, or TIFF. If they do not, they may be scanned in and converted to one of these formats.

Like most image-editing tools, Image Editor allows you to touch-up or enhance the quality of the scanned or imported image. However the Image Editor goes on to convert these PC file formats to AFP format, allowing you to work natively with the AFP object at the PC graphic design level. Specifically, it creates an AFP page segment, which will be uploaded to the iSeries when the project is uploaded. At any time, the image may be imported into the AFP overlay.

2.6.1 Creating separate AFP page segments

Your requirements for an overlay may be fairly simple, for example the company logo plus the address and contact details of its offices. In this case, select **File-> Get image** to incorporate the company logo as part of the AFP overlay. Anytime you print that overlay, the same logo prints with it.

On the other hand, there may be times when you want to use a page segment independently of any overlays being used. Examples of this are:

- Different signatures used on a check depending on the value of the check
- Using one constant overlay shell where you change the company logo based on some condition in the application
- Variable positioning of the page segment using other tools such as DDS

Tip

Even if a page segment is not going to appear on an overlay in your application, you may choose to create a dummy overlay project and add the page segment or segments to it. That way it will be easy to upload the page segments to the iSeries. It will also help with testing.

2.6.2 Using Image Editor

Image Editor is a relatively straightforward graphical design tool, with features such as online context-sensitive help and floating “bubble help” for the icons. The best way to learn its features is simply to start using it. However, there are some important points that you should note.

2.6.2.1 Print resolution of resources

You should set the resolution of the resulting AFP image to match that of your target printer. Typically this is 300 picture elements per inch (pel), also referred to as 300 dots per inch (dpi)). Note that even though your target printer prints at 600 pels, as far as PSF is concerned, the resolution is likely to be either 240 or 300 pels. For example, printers such as the IBM Infoprint 20 usually print at 600 pels, but the resources sent to them will be either 240 or 300 pels. Such printers as the IBM Infoprint 70 have the ability to receive 600 pel resources directly, but for portability, 300 pel should be sufficient.

2.6.2.2 Print orientation of resources

AFP images are known as *page segments*; they are individual segments of a logical page, and may contain their own characteristics, different from that of the rest of the page. The most important of these is the orientation of the page segment. If created at zero degrees rotation, it always prints at zero degrees orientation, even if the rest of the document and overlay are rotated. Therefore, it is common to produce at least two versions of each page segment, one at 0 degrees, the other at 90 degrees, together with a suitable naming convention such as LOGO and LOGO90.

2.6.2.3 Creating page segments

If your image exists in a color form, try to obtain a black and white version before converting it to a page segment. If this is not possible, *dither* the logo first. Select **Image-> Dither** and then select **Color B/W** or **Greyscale**. The best choice may require some experimentation. Remember to use **Edit-> Undo** (or Ctrl-Z) if you don't see satisfactory results the first time.

Having converted it, you need to select **File-> Save as**. Save it as an AFP image (page segment) with a suffix of *.PSG. You are then prompted for the AFP file parameters. There are many possible combinations, but a typical selection would be like the example shown in Figure 15.

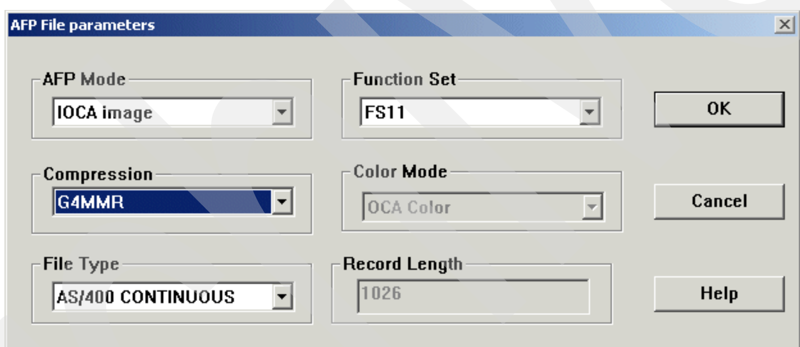


Figure 15. Typical combination of save parameters for an AFP page segment

2.7 Layout Editor

Layout Editor is the tool within Infoprint Designer that you use to design the complete electronic form, adding overlays and images to data from your application.

Infoprint Designer can work with either unformatted or formatted data. Unformatted data is sometimes known as a “flat file” or *line data*. Formatted data may have some or all of the data placement already determined, for example a customer name and address at the top left and an invoice total at the bottom right. In either case, we can use Infoprint Designer to redesign the page layout without changing the original application.

2.7.1 Selecting sample data from the iSeries server

The first task is to gather sample data from typical pages of the selected application. Infoprint Designer is tightly integrated with the iSeries server, which is why the first step in an Infoprint Designer session is to sign on to an iSeries session.

As soon as you click the **Layout** menu button, you are prompted for the source of your data. To begin with, it’s likely that this will be the iSeries server. By selecting the **AS/400** radio button, you are provided with the list of output queues to which you have access on the iSeries server. Then speed up the search if possible by typing an output queue name that contains your test spooled files. See Figure 16.

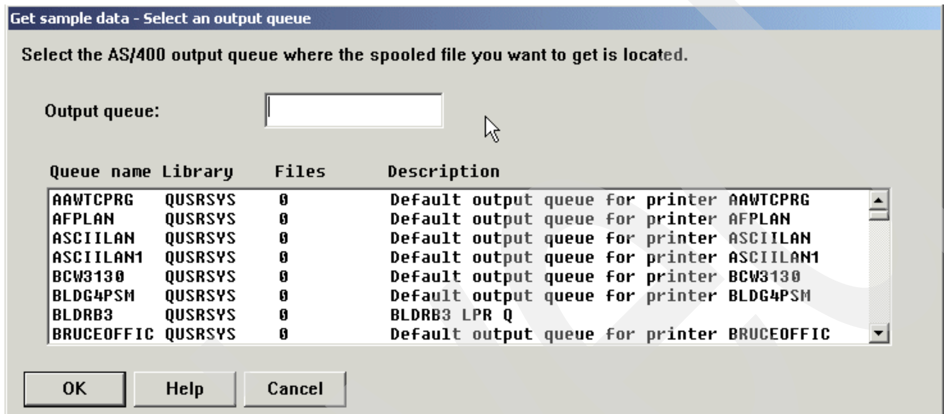


Figure 16. Selection of output queue containing sample data

You could scroll through the list of output queues but assuming you know the name of the queue, enter it and click **OK**, as in Figure 17.

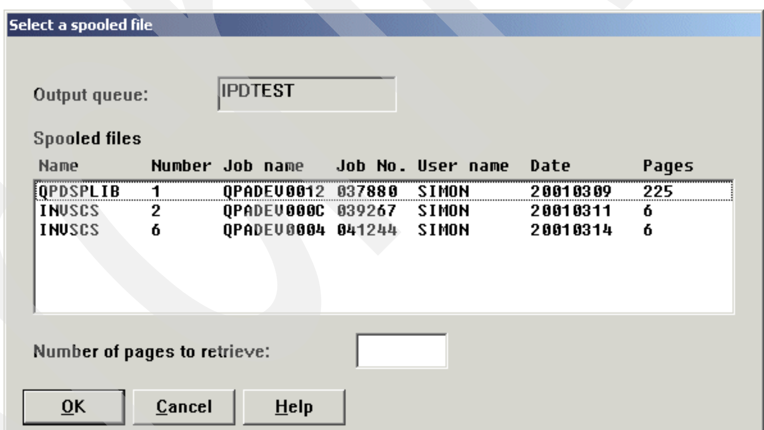


Figure 17. Selection of a sample spooled file

The next dialog you see is the “Select line data file”. This is actually prompting you to save the spooled file data locally on your PC. Give it a suitable name and suffix (for example ASC – Ascii) and save the file. Your View/Edit window remains blank, apart perhaps from any overlays you have in the background.

Now click **Window-> Data** and you see a new window entitled the Data Window. This contains your *unmapped* sample spooled file data, blue in color. Now click **Window-> Tile** to access the Data Window and the View/Edit window displayed together. From here you can drag and drop selected areas of text into the View/Edit window, as described in the *Getting Started* manual. If an overlay is already open, you will notice it is in the background, as shown in Figure 18.

When you have the windows arranged to your liking, click **Windows-> Arrangement-> Save**.

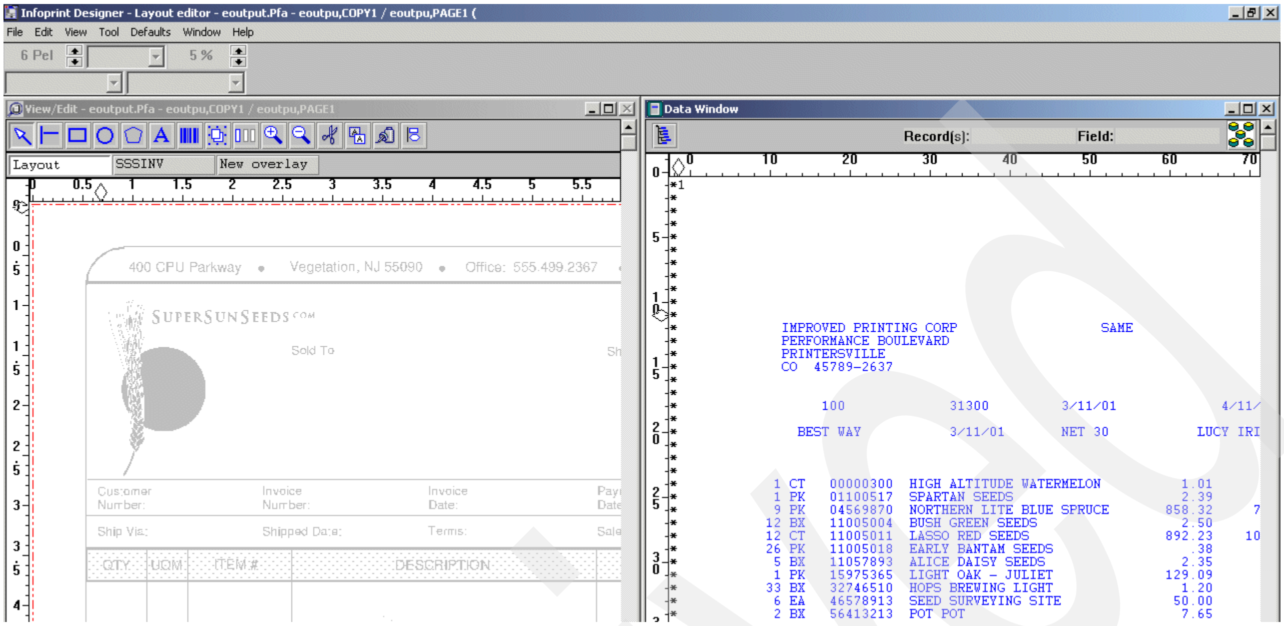


Figure 18. iSeries sample data about to start data mapping with Infoprint Designer

2.7.2 Selecting sample data from the PC

When using Infoprint Designer, you always connect to the iSeries server. However, you may later find that the sample spooled file you used for your designs was deleted from the system, perhaps through regular spooled file housekeeping or simply because it was printed! The solution is to use the local copy stored on your PC, which Infoprint Designer refers to as the “line data file”.

To access this file, select **File-> Get sample data** and select **From PC**. By default, Infoprint Designer looks in the \DESIGNER\DATA directory. You can select a file of type *.ASC (meaning ASCII data), but you can select any text file you have available. You are then prompted further as shown in Figure 19.

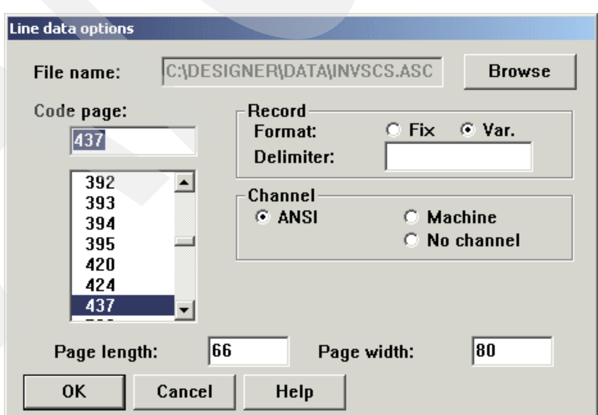


Figure 19. Line data options when importing a sample spooled file from the PC

The default values in this window are taken from the original downloaded spooled file attributes. Therefore, you should leave all or most values as they are, but consider the following notes, which may be of help.

2.7.2.1 Code page

The code page should match the code page Windows is using on your PC. Typically this is 437 (MS-DOS Latin 1 US) or 850 (MS-DOS Latin 1Multilingual). If you are not sure which code page your PC is using, type `chcp` at a DOS prompt to display the active code page. Use care when referring to Windows documentation. The terms “code page” and “character set” mean the same thing. This is not the case with the IBM AFP architecture.

2.7.2.2 Record format

Use the *variable record format* with PC spooled files previously derived from the iSeries (even though the iSeries is a fixed record format system). Also use the variable record format with PC spooled files sourced from IBM mainframe systems such as OS/390. If you are extracting data from a spreadsheet file, you might select a fixed record with a delimiter such as tabs or commas.

2.7.2.3 Channel type

In this context, “channel type” refers to line spacing and page skip controls that may or may not be in the PC file. A more common term is *carriage control* or, in iSeries, First Character Forms Control (FCFC).

With *ANSI* encoding, a line space occurs *before* the printing the line. Common ANSI FCFC characters include a space (write, then enter one line space), + (no space), 0 (two spaces), - (three spaces), and 1 (write after skipping to a new page). The latter case is also known as a “Channel 1”.

Machine encoding means that the data is printed and then any line or page skip controls are performed, for example, *after* the data line.

“*No channel*” means that no spacing or skip controls are present in the file. If this is the case, you have a true “flat file” of data and the file is unlikely to be a spooled file but perhaps imported from another source such as a spreadsheet or data file.

ANSI line spacing control is the most common channel type on the iSeries platform. If your spooled file originated on the iSeries, this is the most likely choice for your PC file.

2.7.2.4 Page length and width

Leave these at their default setting (for example, as inherited from the iSeries spooled file). The page length is especially important because this dictates how many records (lines) will be available for mapping on each page of output.

2.7.3 Mapping data to the layout

Once you successfully load the sample data into your project, you can start mapping individual fields or groups of fields. You do this by highlighting the area you want to select from within the Data Window, using the left mouse button, and then dragging it using the right mouse button.

It is a good idea to have the documentation for the application you are working on available. The data you see on the sample page you are looking at may not represent the full length of the fields. For example, in the Invoice example in Infoprint Designer, the original programmer's DDS code (not included) showed that the field NAME is placed at position 12 on the line and is 25 characters long. You can verify that you have the correct position and length of field by watching the information displayed at the top of the Data Window as you are highlighting the area. If you don't have access to this information, you must ensure you test your new print application with as wide a variety of sample data as possible.

When you first bring in the data, the first logical page is represented in the Data Window in blue. Subsequent pages are in red. All the field mappings that you perform should be selected from a blue page. If you need to bring up a different page of the data, click the *Next* button (just below the menu items) to page through the available data.

Once you place the field on the View/Edit window, you can make some further modifications to it. Right-click the field to see the Field Properties window. From the General tab, you can verify the positions from the original input record and field length against your application documentation. Click any of the other tabs to change the characteristics of the field, such as font or orientation. Figure 20 shows an example of mapping a field and modifying its properties. Note the entries at the top bar of the Data Window showing that we selected Record 12, Field 12-36.

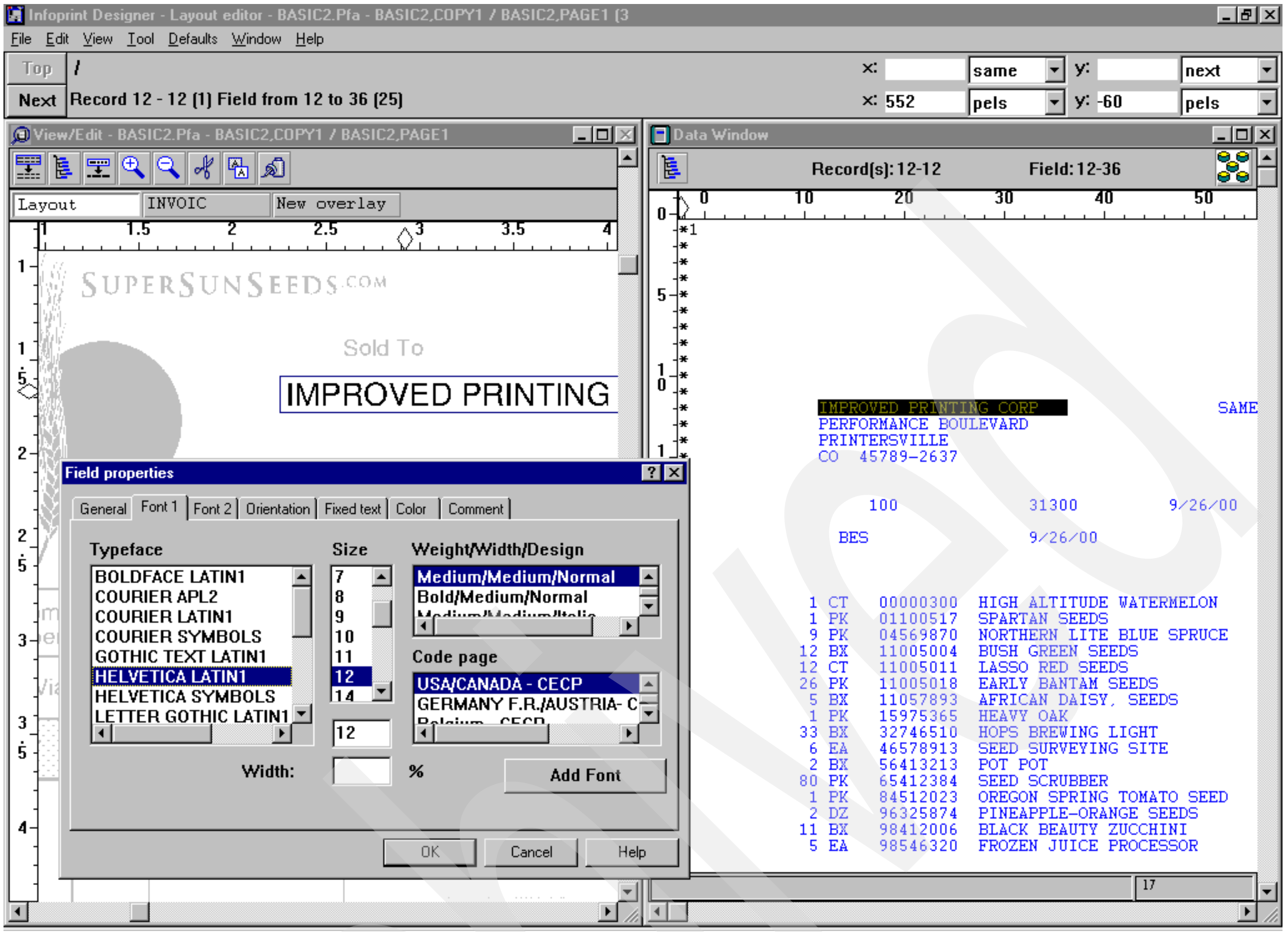


Figure 20. Mapping and modifying a field

If a column of fields is to have similar characteristics, you can select the entire area and then map them in one step. Figure 21 shows the results of highlighting and mapping a column. The fields remain logically grouped together in your design, so you can move them or change other characteristics in mass.

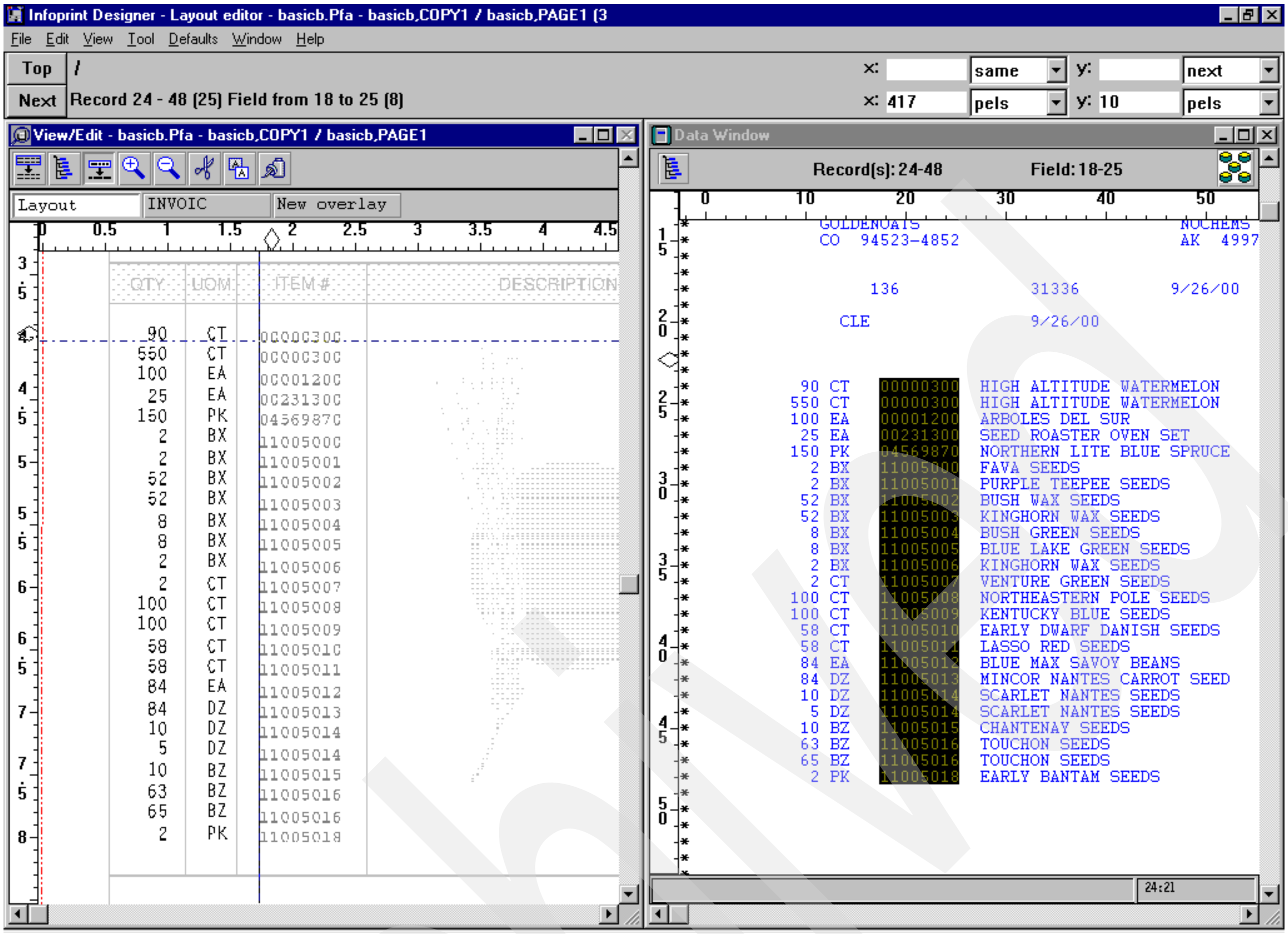


Figure 21. Mapping a column of fields

If you want to make group changes, click just the first element of the group and then either drag it to a new position, or right-click to change other characteristics. If you want to change the interline spacing of a group of fields, click any field in the group, other than the first, and drag it up or down until the column is arranged to your liking.

You may notice that the information bar at the top of the screen shows slightly different information, depending on which field you select. If you select the top field, you see the absolute position at which it is mapped. If you select other fields, you see the relative positions.

If you have multiple columns of data in a section, you can adjust the line spacing of all the columns at one time. Click one of the fields in one of the columns and hold down the Ctrl key while you adjust the line spacing. All the other columns that use the same range of lines adjust accordingly. This action synchronizes the interline spacing. You have to manually ensure that the first entries in the group are aligned.

Note

It is possible to ungroup a set of fields, using the Ungroup Printline icon from the toolbar. Use this with caution. Once you have done so, you cannot easily regroup the fields.

Continue mapping all the required fields from the Data Window to the View/Edit window. You may find it easier to do the mapping on a gross level first, just dragging and dropping the required fields over to their approximate destination View/Edit window. Then close the Data Window and enlarge the View/Edit window to fine tune the positions and make other changes to the field properties.

As a final test, you may want to open both the View/Edit and the Data Window and page through all the available data pages to make sure you have captured everything you need.

2.8 Introduction to form and page definitions

Underneath the covers, Infoprint Designer builds two very important AFP resource objects called a *form definition* and a *page definition*.

A form definition is concerned with physical entities such as simplex/duplex, print quality, paper source drawer, and the position of the logical page on the physical piece of paper. A page definition defines the appearance of the logical page of data, for example the position of the text, the fonts, any overlays, the page orientation, and so on.

Note that these two formatting controls are *application-independent*. With the addition of the easy-to-use Infoprint Designer GUI, document presentation changes need no longer involve the iSeries application programmer.

For basic use of Infoprint Designer, it is unnecessary to take note of the above information. However for more advanced use, it is very useful to have an understanding of how page and form definitions interact to build the final electronic document. We look at several areas that make closer reference to the concept.

2.8.1 Form definition components

A form definition specifies how the printer controls the processing of the physical sheets of paper. Within a form definition, you specify functions such as duplex, overlays, copies, suppression, jogging (offset stacking), paper source, constant pages, multi-up, and finishing.

The form definition has two important subdivisions: copy groups and subgroups.

2.8.1.1 Copy group

A *copy group* defines the current physical page within a form definition. Note that it defines the physical page and could therefore include the formatting instructions for two sides of a piece of paper. A form definition may contain only one copy group. If a different copy group is invoked, it follows that printing must start on a new physical page. An alias for copy group is *medium map*.

Common purposes for a copy group include the ability to pick from a different drawer on the printer. Another example is a switch between simplex and duplex. A final example is the printing of an overlay of standard terms and conditions on the reverse side of client copies of a purchase order. Switching between copy groups in the form definition is controlled by conditional processing in the page definition.

2.8.1.2 Subgroup

A subgroup is a subset of a copy group. Up to 127 subgroups can be constructed within any one copy group. The easiest way to remember the function of the subgroup is to think of multipart stationery (top copy, pink copy, blue copy, etc.), as commonly used on impact printers. To replicate this electronically, we can use multiple subgroups to repeat each printed page as required. The difference is that each copy may be altered subtly if required.

For a duplexed sheet of paper, there is a subgroup for *each* side.

2.8.1.3 Form definition logic tree

Infoprint Designer uses a logic tree structure to represent the form definition being used in the layout. To display or work on the form definition tree, select **Layout mode** from the View/Edit window, and then select **Window-> Form definition** from the main menu. From the Form Definitions window, you can easily add copy groups, subgroups, or overlays to your design, or change their properties. Figure 22 shows an example of a very basic layout.

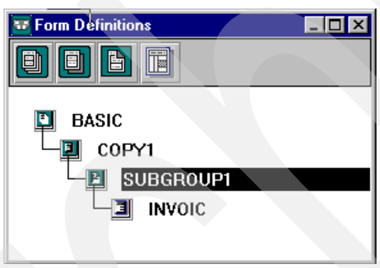


Figure 22. Basic form definition tree

In this example, we have a form definition named BASIC. Within it, we have one copy group COPY1. Having only the one copy group means that all pages of data will print using the same media. Within the copy group, we have one subgroup, SUBGROUP1. With only one subgroup defined, the resulting output will only generate one copy of each page. The selection of the overlays to be printed on every page is defined within the subgroup. In this case, we are using INVOIC, the Super Sun Seeds Invoice.

2.8.2 Page definition components

The page definition specifies how you want data positioned on the logical page. The primary purpose is to map each line of input data to a position on the formatted page. You may change the fonts of a field, rotation, interline spacing; turn it into a barcode; or indicate that you want to suppress printing. Page segments and overlays may be defined within a page definition. There are controls in the page definition to control multiple-up printing. Finally, the conditional processing logic to switch between different page formats or copy groups is defined in the page definition.

The page definition must contain at least one *page format*.

2.8.2.1 Page format

Another name for the page format is a *data map*. This is a less common term but the name may help you to remember that the data map or page format *maps data* onto a page. Different page formats result in different page layouts. For example, you may want to lay out the data of a detail page of a report differently from the cover page or the final summary.

Typically, different page formats are selected by “trigger fields” in the application data. The trigger fields could be anything you choose, as long as they are in a predictable position in the original application data. Testing these fields is done in the page definition itself. Switching between different page formats is another example of conditional processing.

2.8.2.2 Subpage

All page formats generated with Infoprint Designer contain at least one subpage. It is within the subpage that the actual data mapping takes place. Multiple subpages within a single page format allow you to place multiple logical pages on one physical page.

Subpages are also used in controlling the timing of when a switch takes place in conditional processing. This may be seen in the condition definition window. See Figure 34 on page 63 and Figure 35 on page 64, as well as 2.9.3.3, “Advanced conditional processing” on page 65.

2.8.2.3 Printline

Printline is the instruction within a subpage that is essentially used to define what to do with each line of data in the input file. It is possible to group similar information, such as the detail lines in the body of a report, using the repeat function. An entire line may be formatted using printline, or it may be split into individual fields. Page segments or overlays may be associated with a printline.

Conditional processing instructions are associated with specific printlines. Printlines may be selected based on a *channel code* that may be in position 1 of the data. The most common instance of channel code is a “1” being used to indicate a new page. Subsequent printline commands are processed in sequence. For more information on channel codes, see 2.7.2.3, “Channel type” on page 47.

2.8.2.4 Fields

Individual strings from the input data may be formatted independently of the rest of the line. For each field, you can define its font, location, suppression, orientation, or whether to print it as a barcode.

2.8.2.5 Page definition tree

Infoprint Designer uses a tree structure to illustrate the page definition being used in your layout. An example of a basic layout is shown in Figure 23.

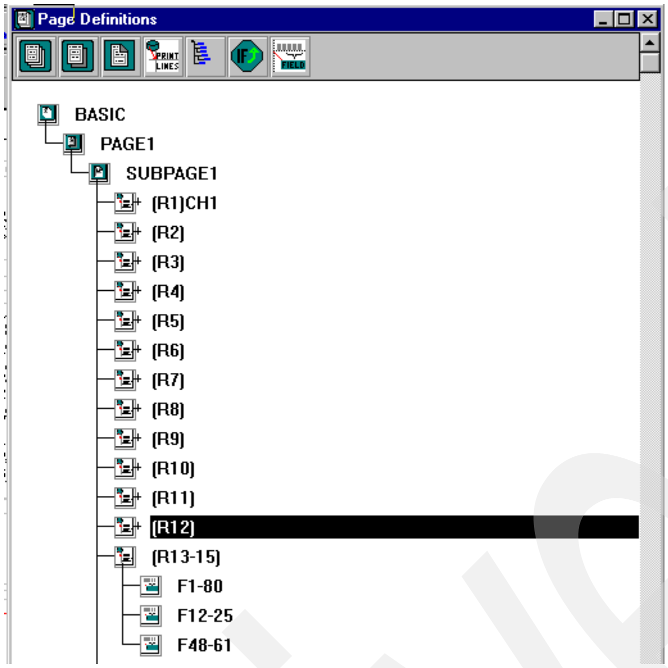


Figure 23. Basic page definition

In this example, we have a page definition named BASIC. Within it, we have one page format, PAGE1. Having only the one page format means that all pages of the input file will be mapped the same way, using the same fonts, etc. There is one subpage, SUBPAGE1, which means that we will print one logical page per physical page.

The first printline instruction, [R1], is triggered when channel 1 is encountered in the data, as indicated by CH1. The next eleven data records are processed individually in sequence. (The field mappings have been collapsed, as indicated by the “+”.) Records 13 to 15 are all to be handled using the same rules, as indicated by [R13-15]. In this case, a field consisting of characters 1 to 80 [F1-80] was mapped by default by Infoprint Designer to be suppressed. Then, as part of the act of creating the design, a field [F12-25] consisting of characters 12 to 25 was placed on the page in a specific location with a specific font, as were characters 48 to 61 [F48-61].

2.8.3 Infoprint Designer and suppression

When you start a new Infoprint Designer session, a default layout is created, even before you do any field mapping. The form definition that is created contains one copy group, which in turn contains one subgroup. It you started the project with an overlay definition, it will have the default overlay mapped into the subgroup. The subgroup will have suppression set with the name DFLT as shown in Figure 24. This means that any fields defined in the page definition with the Suppression subcommand set to DFLT will not print.

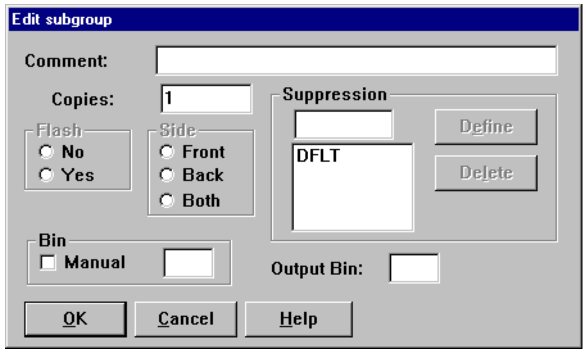


Figure 24. Subgroup with DFLT suppression

To see the details of the subgroup, or any other item in a form definition or page definition logic tree, right-click the item.

The page definition has one page map and one subpage. As soon as you bring in the sample data, Infoprint Designer creates one printline for each line in the sample data page. Each printline has one field defined corresponding to the length of the data. Each field has suppression set to DFLT, as shown in Figure 25. To see the suppression settings on a field, right-click in that field in the page definition logic tree and then click the **Suppress** button.

Figure 25. Field with DFLT suppression

Essentially, Infoprint Designer has created a layout that maps the entire page, but suppresses it all at the same time. If you are only doing single page mapping, this may not concern you, but if you are using some of the more advanced techniques, such as conditional processing, copy groups, or subgroubs, you need to understand what is happening.

First, here's some background. The basic layout technique of page definition is to describe how each line is to be handled in sequence. For example, do something with line one, do something else with line 2 and repeat it 5 times, and then do something yet again with line 7. If you didn't want to do anything with line 8 of the input file, you have to explicitly inhibit it, usually using suppression. If you do not, the next instruction that you intended for line 9 is used against line 8.

When you use Infoprint Designer to map fields, you only need to drag and map the fields you want to use. If you want to print the address that starts on line 12 of the input data, you only need to drag that line across. Infoprint Designer has already taken care of lines 1 through 11 by suppressing them for you.

If you create a new subgroup, make sure you specify the DFLT suppression on it. Otherwise, you will end up with two mappings of your data field – the ones that you intended plus the ones that are supposed to be suppressed. Figure 26 shows an example of what you may see if you forget to do that.

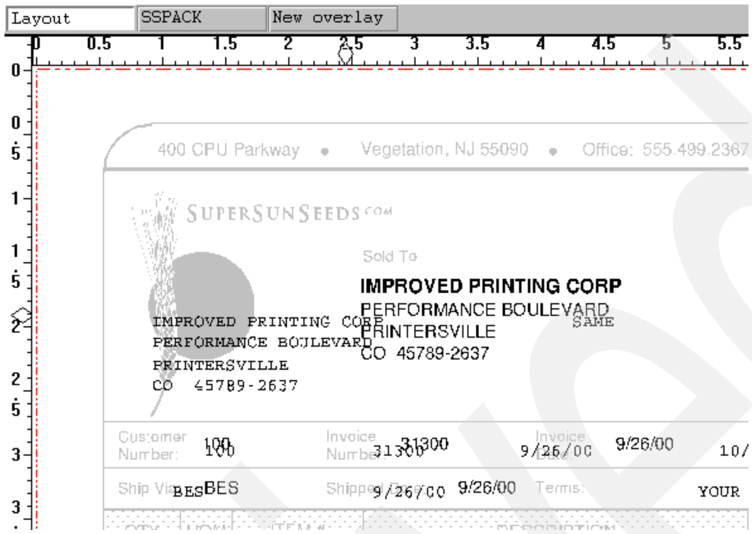


Figure 26. Example of double mapping

A way perhaps to avoid this is to take copies of a subgroup, rather than create a new one. The fast way to do this is to display the form definition window (**Window-> Form definition**) and select the subgroup. Hold down the Ctrl key and drag the subgroup to just below its original position. Release the mouse button and a second subgroup is created. This method is used frequently in the following examples.

2.9 Introduction to advanced examples

The following sections describe step-by-step instructions on using advanced form and page definition features to enhance your applications. Sample projects have been included with your copy of Infoprint Designer for iSeries. These examples are based on some of those samples. You may want to copy the samples and try to make the same changes as described in the following sections. For a complete list of the available samples and the features they demonstrate, see Appendix B, “Infoprint Designer for iSeries projects” on page 329.

2.9.1 Using subgroups

When migrating an application from multipart impact printing to laser, one of the first things you may want to do is use Infoprint Designer to create multiple copies of the pages in your spooled file. You can do this easily by defining multiple subgroups within one copy group. The copies do not have to be exactly the same. Each subgroup could have different overlays associated with it. Paper may be selected from different bins, allowing you to select different-colored paper for example. Finally, you may have a different set of suppressions set up within the different subgroups. For example, it is common practice not to print the prices on a packing slip. Alternatively there may be information on the internal copy (such as accounting information) that you would rather not have printed on the client copy.

Note

When designing multiple copies of a page using subgroups, be aware that the data mapping of each copy will be the same. The only difference is that it is possible to *suppress* different combinations of fields on the different copies.

2.9.1.1 Packing slip example

This example assumes that the design and mapping for an invoice page has already been completed. This corresponds to the sample project *basic.prj*. Now we want to modify the design to add a second copy of page using a different overlay, different paper source, and suppression on the price fields. The sample project *twocpy.prj* reflects these changes. The steps you need to take to accomplish this are:

1. Open the original invoice project on which you want to base the new project.
2. It is may be a good idea to select **File-> Save project as** immediately so that your new work does not affect the original project. On the Save project as dialog, change the name of the project file and the layout file. Click **OK**. The form definition and page definition take on the new name.
3. Open the Form Definitions window. You see the existing logic tree with one copy group, one subgroup, and the Invoice overlay. We want to add a second subgroup to this definition. Copy the original subgroup.
   a. Hold down the Ctrl key and click the icon for the subgroup. A slightly larger icon appears with a red cross through it.
   b. Holding down the mouse button, move the new icon down so it is just below the original icon. The red cross should disappear.
   c. Release the mouse button. Infoprint Designer creates a new subgroup on the logic tree, exactly the same as the original one, and just below it.

   If you wanted a brand new subgroup that wasn't to be a copy of the existing one, you could drag down the Subgroup icon from the toolbar. Remember to add the DFLT suppression any time you create a new subgroup.
4. You can now modify the new subgroup. If you copied the original subgroup, the new one will have the Invoice overlay associated with it.
   d. Click the icon for the Invoice overlay within the new subgroup and press the Delete key.
   e. Add in the Packing Slip overlay (assuming that you created it earlier). Click the overlay icon on the toolbar, and drag it down to the new subgroup.
5. You are now presented with an Open dialog box. Select the desired overlay (**PACK** in this case).
6. Change some of the other characteristics of the new subgroup.
   a. Right-click the icon for the subgroup. The Edit subgroup dialog displays, as shown in Figure 27.
   b. We want to create a suppression tag so that the price fields do not print. Place the cursor in the entry field under the Suppression heading and type the name of the tag that will be used on the price fields. In this example, we will use PACK.

c. Click the **Define** button and the tag moves to the larger box below. The DFLT suppression should be listed there already. If not, add it in.

d. Now you can add a meaningful comment, and change the paper source to **Bin 2.**

e. Click **OK** when the changes are complete.

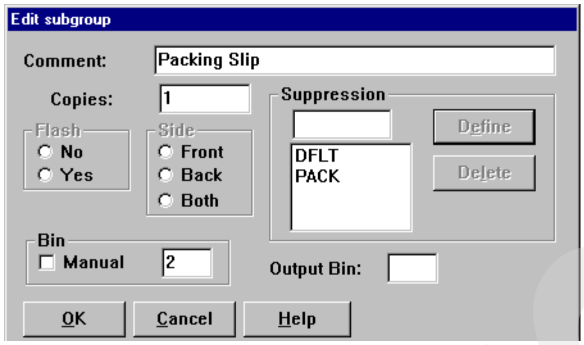


Figure 27. Edit subgroup

7. Modify the price fields in the page definition so that the PACK suppression prevents them from printing on the packing slip.

   a. Activate the View/Edit window. Select the first field in the price column and right-click it to bring up the Field properties dialog as shown in Figure 28.

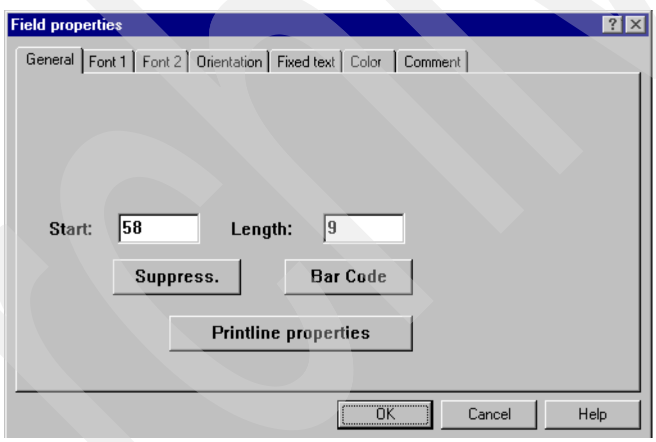


Figure 28. Field properties

   b. Click the **Suppress** button to display the Field Suppression dialog as shown in Figure 29.

   c. You see a list of suppression tags that have been defined in your subgroup. Click **PACK** to activate it for this column of fields.

Figure 29. Field suppression

d. Repeat this process for any other columns or individual fields you want to suppress.

8. Before you complete your project, it is a good idea to check that you have set up the suppression correctly.

   a. Open up both the View/Edit and the Form definition windows.

   b. Activate the new Packing Slip copy group by clicking it on the Form definition window.

   c. Click a blank part of the View/Edit window. The data is reloaded, and you should see the Packing Slip overlay with the description fields, but not the prices. Your screen should look something like the example in Figure 30.

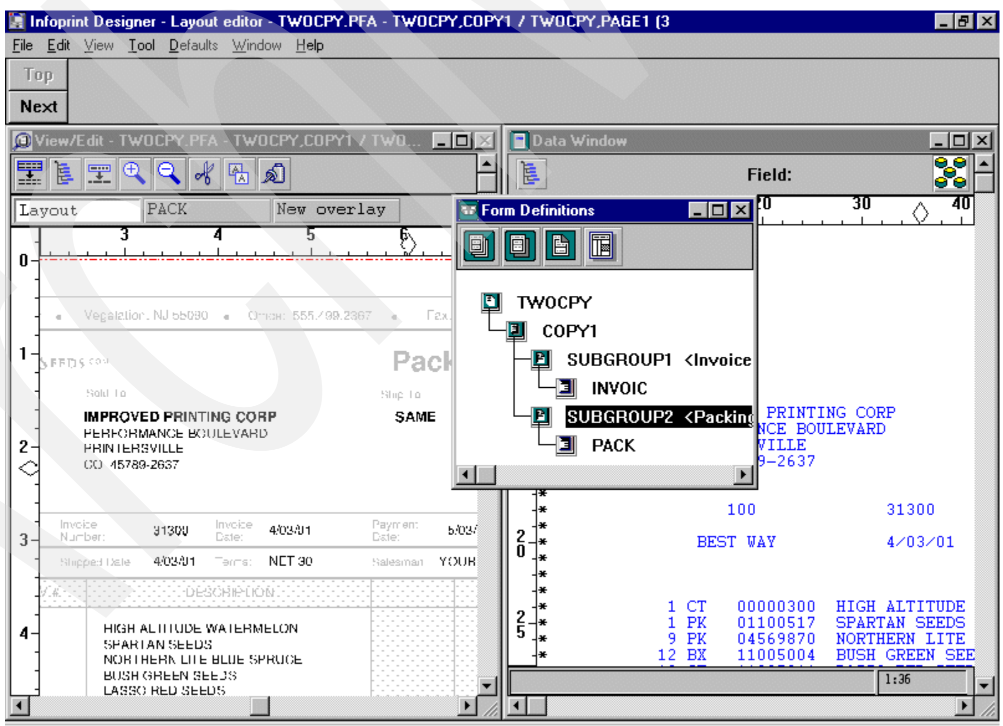


Figure 30. Verifying subgroup

9. Once you are satisfied that your design is complete, save your work and upload it to the iSeries. Remember, that in order for bin selection to be controlled by the new form definition, you must specify `DRAWER(*FORMDF)` on the printer file. This is covered in 2.10.4, “Duplex and drawer selection” on page 83.

2.9.2 Adding a constant back to a subgroup

It is common practice to print a "Terms and Conditions" page on the back of a document such as a company invoice or statement. With subgroup processing, we can do this using a function called *constant back*. This is a useful feature of form definition processing. Without this, the application would need to track what pages were to be printed in simplex and what pages in duplex, and insert a blank page at the appropriate points.

We need to set up the copy group for duplex printing and tell it we want to use a constant back. Then we define a pair of subgroups for each copy of our document, one for the front and one for the back, each with the appropriate overlay defined.

2.9.2.1 Example of constant back

Following the previous example, from 2.9.1.1, "Packing slip example" on page 57, assume we already have a project created that prints two copies of every page, an Invoice and a Packing slip. The sample project *twocpy.prj* has these characteristics. The copies are slightly different, as they draw from different bins. The use different overlays and suppression of prices is in place on the packing slip. Assume we want to add a Terms and Conditions overlay, TERMS, to the invoice copy only. This overlay already exists. We leave the back of the packing slip blank. The sample project *back2.prj* reflects these changes.

1. Start by making the necessary modifications to the copy group, COPY1. Activate the form definition window, and right-click the copy group. Click the **Duplex** tab. You see the Copy group properties window as shown in Figure 31.

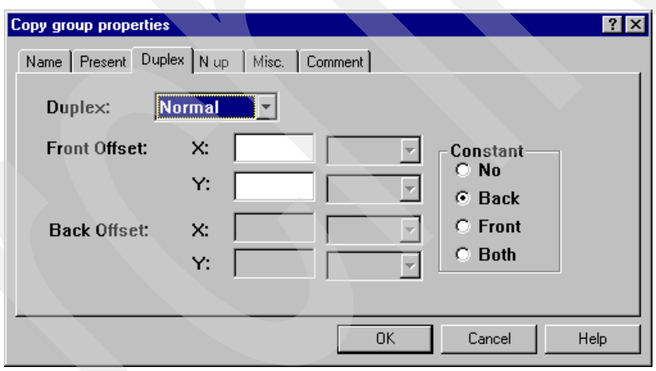


Figure 31. Constant back settings in copy group

   Click the **Duplex** drop-down list and change it to **Normal**. Then click the **Back** radio button in the **Constant** box. Click **OK**.

2. Now you need to set up the subgroups. Start by dragging the subgroup icon from the tool bar to just below the original invoice subgroup. Repeat and place a new subgroup below the packing slip.

3. You should have four subgroups now. Right-click the **Invoice** subgroup. For the Side option, select the **Front** radio button. The window should look like the example in Figure 32. Click **OK**.

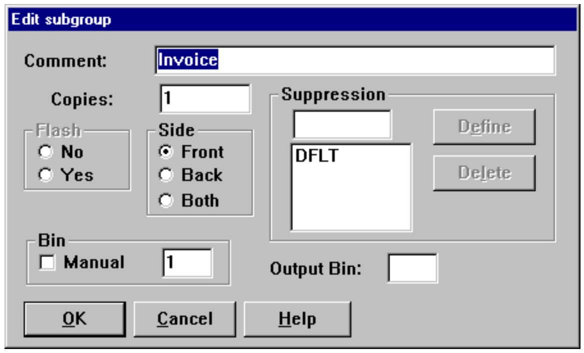


Figure 32. Subgroup specifying front side only

4. Repeat this step for the packing slip subgroup, making it active for the front side only as well.
5. Right-click each of the two new subgroups, and make them active on the Back side only. Add descriptive comments.
6. Add the TERMS overlay to the subgroup that is on the back of the invoice. To do this, drag the **Overlay** icon from the toolbar to the terms and conditions subgroup. Select the **TERMS** overlay from the Open dialog box.
7. When you are done making all these modifications, your Form Definition window should look like the example in Figure 33. Note the value of comments and a good naming convention.

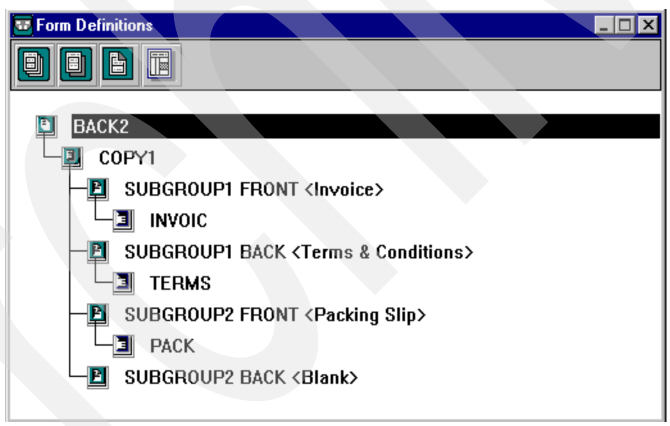


Figure 33. Form definition with constant back

8. Save your completed work and upload it to the iSeries. Remember, that in order for bin and duplex selection to be controlled by the new form definition, you must specify `DRAWER(*FORMDF)` and `DUPLEX(*FORMDF)` on the printer file (covered in 2.10.4, “Duplex and drawer selection” on page 83).

2.9.3 Using conditional processing

By using conditional processing, electronic documents start to change from being direct replacements for pre-printed forms to highly-variable, customized print applications that react to the data within them. Individual pages within the document may be automatically tailored to specific functions (invoice or credit note) or individual customers (valued or delinquent) or specific pages within a set, such as the final page of an invoice versus the other pages.

2.9.3.1 Concepts of conditional processing

Conditional processing may be defined as "use of print data to control the appearance of the final print output". In the page definition, you define a field and a comparison test to do on it. You can compare the field against a constant, or check to see if it changed from the previous instance. Based on the results of the test, you can tell Infoprint Designer to use a different copy group or a different page format.

Conditional processing may also be used to force a new piece of paper in a duplex environment or force a new side in a multi-up job.

It is also possible to control *when* the change is to take place. Should Infoprint Designer finish the line or subpage it is on before making the change, or restart the line or subpage with the new layout?

In the example in 2.9.3.2, "Example of simple conditional processing" on page 62, we use a fairly simple example of conditional processing. For a detailed explanation of how to use all of the features of conditional processing, refer to *IBM Page Printer Formatting Aid: User's Guide,* S544-5284.

2.9.3.2 Example of simple conditional processing

In this section, we look at the Super Sun Seeds invoice as an example of conditional processing. On the last page of each invoice, a "Thank You" message is printed at the end of the item list. We want to bring more attention to that message by changing the font and moving it down the page to the blank space at the bottom.

We can identify the last page because the content of the "Total due" box is different. On the last page, it contains the amount that is due. On earlier pages of a multi-page invoice, the word "Continued" prints in that box. We take advantage of this to set up our conditional processing trigger.

Assume that we already have a working design for the invoice, using one page format. The sample project *twocpy.prj* was the starting point for this example. The results are reflected in the project called *cond.prj*. Here are the steps to create the new version of the document using conditional processing:

1. Make a copy of the original project and save it under a different name.
2. It may be easiest to work on this project if you open both the View/Edit window and the Page definition window. In the Page definition window, you should see one page format with one subpage. It may be easier at this step if you collapse the subpage tree if it is expanded. You can do this by double-clicking the icon.
3. You need two different page formats for this application. Holding down the Ctrl key, right-click the existing page format and drag it down until you can place a new copy below it in the logic tree. You should now have two identical page formats in your logic tree. Right-click the first and change its name to **TOTAL**. Change the name of the other one to **CONTD** for Continued.
4. Let's assume you do not want to make any changes at this time to the continuation pages. Click the **TOTAL** page format to activate it, and then click the **View/Edit** window to refresh the data. You can verify that the correct page format (or copy group) is selected at any time because it is displayed in the title bar of the View/Edit window.

5. Make the necessary changes to the field mappings. In the original page definition, you may have had all the description fields in the detailed lines mapped as one repeating group, of 25 lines. In the new page format, we only have 16 detail lines. An easy way to change this is to right-click the first field in the description column to bring up the field properties. Then click the Printline properties box. Change the repeat value to 16, and click **OK** twice.
6. Now, you are ready to remap the “Thank you message”.
   a. If it isn’t already displayed, open the Data window.
   b. If the page being displayed does not have the message, click the **Next** button in the control bar until a page that does have it is displayed.
   c. Highlight the message information, drag it to the View/Edit window, and format it as you want.
7. Add the condition to the page definition. Let us add it first to the TOTAL page format. Click the **TOTAL** page format to make sure it is active, and then click the **View/Edit** window to refresh the data.
8. In the View/Edit window, click the field in the Amount Due box. A thin blue box appears around the field. Then click the **Condition** icon on the tool bar. The Conditional processing dialog appears. It is a good idea to give the condition a meaningful name so it will be easy to reference from other parts of your page definition. We call it **TOTALBOX**.
9. If the total amount prints in the box, it can start in position 68 if the highest amount possible prints. However, if the word “Continued” prints, it starts in position 70. We need to change the Start of the comparison field: entry to match this. The length to check will be 9.
10. Insert the first part of the condition. Click the **Insert** button to display the Condition definition dialog (Figure 34).

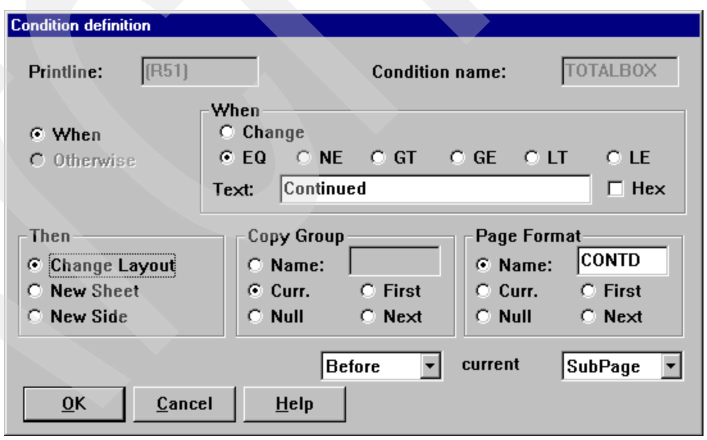


Figure 34. Condition definition: When condition

Follow these steps”

a. To get to this point, make sure the **When** radio button is selected, on the left.
b. In the When box, to the right, select the **EQ** radio button.
c. This brings up a small window in which you can type the word `Continued`.
d. In the Then box, select the **Change Layout** radio button.

e. We do not change Copy Group, so that can be left with **Curr.** (current) selected.

f. Under Page Format, click the **Name** radio button and enter the name of the page format, CONTD.

g. Click **OK** when all selections are complete to return to the Conditional processing dialog.

We just told Infoprint Designer that when the field in print line 51, starting in position 70, for a length of 9, equals the string “Continued”, change the layout so that it uses the page format called CONTD.

11.Click the **Insert** button again. In this example, there are only two choices: a “Continued” page or a “Total” page. Once the first choice is defined, you can use the Otherwise function. Click the **Otherwise** radio button, and then make the selections so that the **TOTAL** page format will be used. Your display should look like the example in Figure 35.

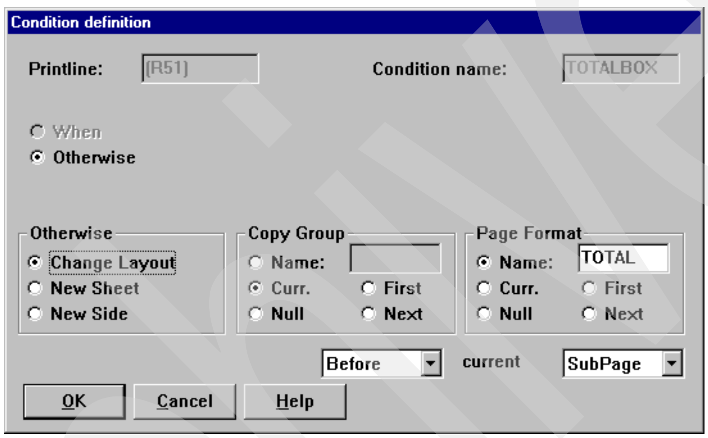


Figure 35. Condition definition: Otherwise condition

When your selections are complete, click **OK**. The Conditional Processing dialog should now look like the example in Figure 36.

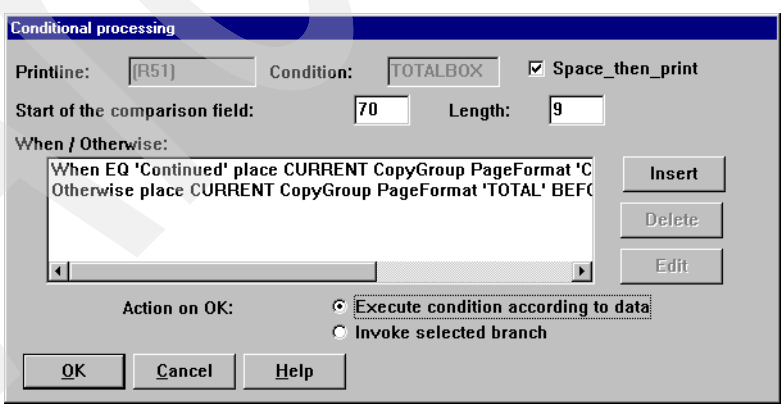


Figure 36. Completed condition

12.Make a note of the Printline value. In this case it is **R[51]**. You need to know that to set up a similar condition in the CONTD page format. (If you don’t set up a similar condition in the CONTD page format, once you reach the condition that takes you from the TOTAL format into the CONTD format, you have no way of knowing when it is time to leave the CONTD format and go back to a TOTAL page). Click **OK**.

13. Go to the Page definition window, and activate the **CONTD** page format.

14. Rather than redoing the logic for the condition, you can use the existing condition as a reference. You are going to associate the copy of the first condition with the same record. You need to have that record displayed in the page definition window. An easy way to do that, is to click in the Amount Due box in the View/Edit window. Make sure you still have the CONTD format active. You should see R[51] in the page format window.

15. Click the **Cond. Reference** icon in the toolbar and drag it down to the **R[51]** line. When you release the mouse, the Select Condition dialog (Figure 37) displays. You see a list of existing conditions defined in the page definition. Click the condition that was created earlier, **TOTALBOX**, and click **OK**.

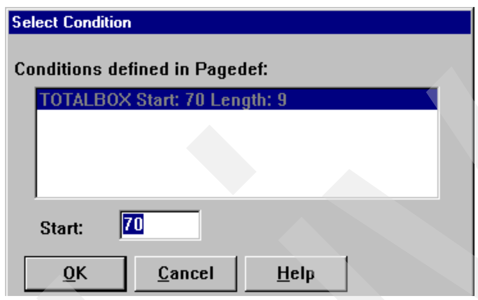


Figure 37. Select Condition

16. You see a tag showing the reference to the TOTALBOX condition associated with Record [51].

17. You have now completed your conditional processing definition. You can check your logic by opening the View/Edit window and paging through the data using the Top and Next buttons on the control bar. The layout of the data should change as the content of the total box field changes.

18. When everything is completed to your satisfaction, save your work and upload it to the iSeries.

2.9.3.3 Advanced conditional processing

These samples just touch on the power of conditional processing available in the architecture of the page definition. For more information on advanced conditional processing topics, we recommend you refer to the *IBM Page Printer Formatting Aid: User's Guide,* S544-5284.

In that manual, you will find information on such situations as:

- Using conditional processing to change copy groups
- Action on a condition at different times:
  - Before the current line is formatted
  - Before the current logical page is formatted
  - After the current line is formatted
  - After the current logical page is formatted
- Using conditions to start a new side or new sheet

- Using conditions triggered by a change in a value, such as change in invoice number
- Multi-up processing

Differences between Infoprint Designer and PPFA

At the time this redbook was written, there are a number of differences between functions available in the architecture of page and form definitions, and the functions available using the Infoprint Designer interface. It is possible that further PPFA functionality could be made available through the Infoprint Designer interface. However this is not guaranteed.

Some of the PPFA functions that are *not* supported by Infoprint Designer are:

- Record Formatting in the page definition
- Table Reference Characters (TRCs)
- AFP structured fields within the line data file
- PRINTDATA YES/NO subcommand in the PRINTLINE command
- Subcommands pertaining to finishing features such as stapling or z-folding available on some printers.

2.9.4 Constant text in the layout

There may be times when you want to add a small item of constant text to your layout but do not need or want to build a whole new overlay. One example of this is to add the project name in small print somewhere on the page. This is also a very good practice to adopt, similar to the reference number or code that external printing companies often print on pre-printed stationery.

Constant text is defined as a field on a given printline and consequently will be specific to the page format that the printline occurs in. If your project includes multiple page formats, you have to add the constant text to all formats to which it applies.

2.9.4.1 Adding constant text in the View/Edit window

To add constant text in the View/Edit window, follow these steps:

1. Activate the View/Edit window
2. Click to select an existing field that is going to print on the page. Click **Tool-> Append field** from the main menu.
3. You see the Field Properties dialog. Click the **Fixed text** tab. Click the **Fixed text** box and enter the text string for the field.
4. Click the **Font 1** tab and make the desired selections. Select **OK**.
5. This takes you back to the View/Edit window with the positioning cross hairs displayed. Use your left mouse button to place the constant text where you wish. It now prints any time the page format is activated.

2.9.5 Creating a landscape project

In the previous examples, data was presented in what is commonly known as *portrait orientation* (where the narrow edge of the paper is at the top). Infoprint Designer may also be used to create projects that are rotated to a *landscape orientation*, namely the data is oriented so that the long edge of the paper appears to be at the top.

There are some unique parameters that must be set correctly for the output to be generated with the correct orientation. This is true for any landscape project.

This section describes how to generate an Infoprint Designer project to simulate a function that is widely used on the iSeries for printing, namely Computer Output Reduction (COR). COR is normally implemented without Infoprint Designer by setting the printer file parameter for Page Rotation (PAGRTT) to *COR or *AUTO.

This is the default used in iSeries printer files, such as QSYSPRT, QPJOBLOG, or QPSUPRTF. With PAGRTT(*COR) or PAGRTT(*AUTO), output that is normally destined for impact printers with forms that are 14 inches wide and 11 inches long is automatically reduced to print on cut sheet printers on 8.5 by 11 inch paper. For a complete description of the rules of COR, see *iSeries Printer Device Programming*, SC41-5713.

The advantage of using COR is that a programmer does not need to make any changes to print simple reports that were 132 or 198 characters wide by 66 lines long (based on 10 or 15 characters per inch and 6 lines per inch).

The disadvantage is that there is no flexibility. The font is predetermined and may even be different on different types of devices. Customers who previously changed their printer files to fit on legal (8.5 by 14 inch paper) cannot take advantage of COR if they must further reduce the output to 8.5 by 11.

With that said, it is possible to simulate the function of COR using Infoprint Designer while at the same time add other enhancements that are otherwise not available, such as:

- Adding an overlay
- Changing fonts for different parts of the report
- Adding conditional processing to format different pages differently

> **Note**
>
> This project is not included in the initial shipment of Infoprint Designer. It can be obtained by ordering the most current PTF for the product.

2.9.5.1 Understanding page origin

When designing a landscape application, the layout of the output is a function of how it relates to the logical page origin. In most cases, the origin of the paper is the top left corner when viewing the sheet in a portrait orientation. When you print a landscape job, the origin remains at the same spot on the page. However, from the data point of view, it becomes the bottom left corner.

There are additional considerations when printing to continuous form printers or when using the *n-up* function available on some IPDS printer models. For more information on this topic, see *IBM Page Printer Formatting Aid: User's Guide,* S544-5284.

2.9.5.2 Creating a landscape overlay

The page size and orientation for an overlay are controlled by the overlay settings (Figure 38). While you are using the Overlay editor, select **File-> Overlay Setup**. Then, select the **Size** tab. The panel shows the defaults, such as Horizontal (8.5 inches) and Vertical (11 inches) for Letter paper, with a 0 degree rotation. Do *not* change the Horizontal and Vertical values. Change the Orientation radio button to 90. This causes the Horizontal and Vertical values to switch automatically.

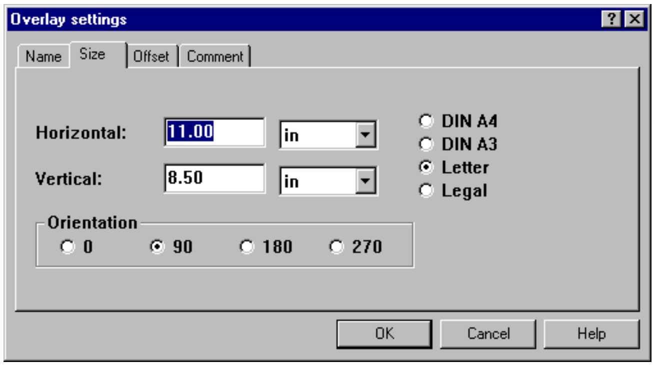


Figure 38. Overlay settings for landscape

Figure 38 shows the panel that appears after selecting the desired rotation. When you close this dialog and return to the View/Edit window, the overlay frame (thin red line) reflects the new paper dimensions.

Using images on a landscape overlay

When you place a page segment on an overlay, its relative orientation, with respect to the origin of the page, remains constant (even if you define the overlay as being at 90 degrees). If you have a page segment that is normally used for portrait projects, you must make a new, rotated version of it for a landscape project. You can do this without leaving your Infoprint Designer project by following these steps:

1. While using the overlay editor, select **File-> Get image**. Select the original (portrait) image and click **Open**.
2. The image may be placed off the top of the page boundary, but you should be able to see the blue outline box in the top left corner of the View/Edit window. Use the mouse to drag the blue box to an area where it is more visible. The image appears to be rotated. Figure 39 shows an example of a portrait image when it is initially placed on an overlay that is set up with Orientation set to 90.

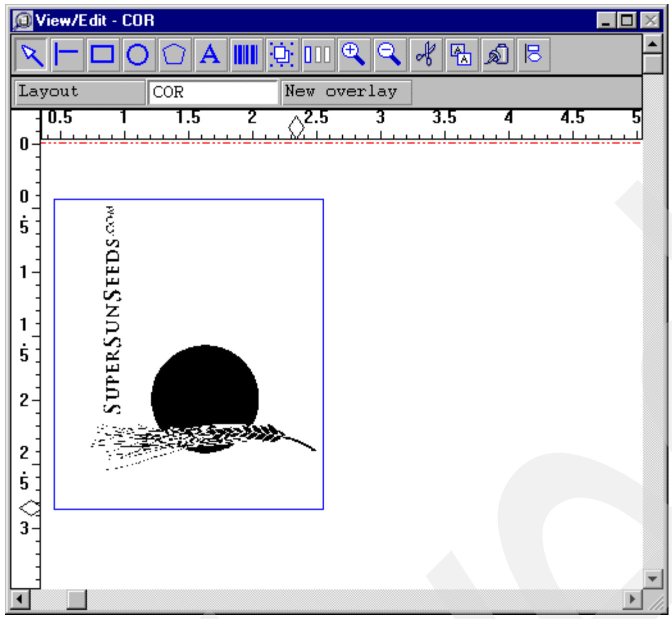


Figure 39. Portrait image on a landscape overlay

3. To rotate the image to 90 degrees to match the overlay, right-click the blue box surrounding the image. The Page segment properties panel appears. Select the **Rotate** tab and for Rotate, click the **90** radio button.
4. You are prompted to save the new page segment in your library path. A useful convention is to add “90” to the original name (in this case `ss2top90`). Figure 40 shows an example of the panel.

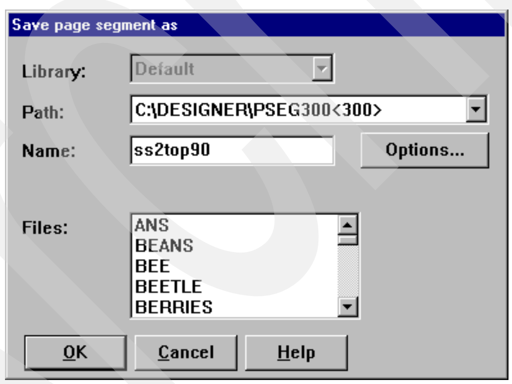


Figure 40. Saving the rotated page segment

> **Note**
>
> If you are using Infoprint Image Editor to create your images, you have the opportunity to create the rotated versions by selecting **Image-> Rotate** from the menus list or by selecting the **Rotate the image** icon.

Overlay specifics for COR project

When COR is used, the interline spacing is reduced to 70 percent of the original specification. Output that normally prints at 6 lines per inch (or .1667 inches per line) is reduced to 8.571 lines per inch (or .1167 inches per line). Each of the bars on this overlay accommodates three lines of print (or .35 inches). The outside box should be at least 10 inches wide and 7.7 inches high to accommodate output that has 132 characters by 66 lines. Add some extra white space to improve readability.

When dealing with an application like this that requires very precise measurement, it is a good idea to use the positioning section of the toolbar to accurately place each element on the page. Figure 41 shows an example of the toolbar for the first box on the overlay.

Figure 41. Positioning section of the toolbar

Figure 42 shows an example of a completed overlay that is used in the COR simulation project.

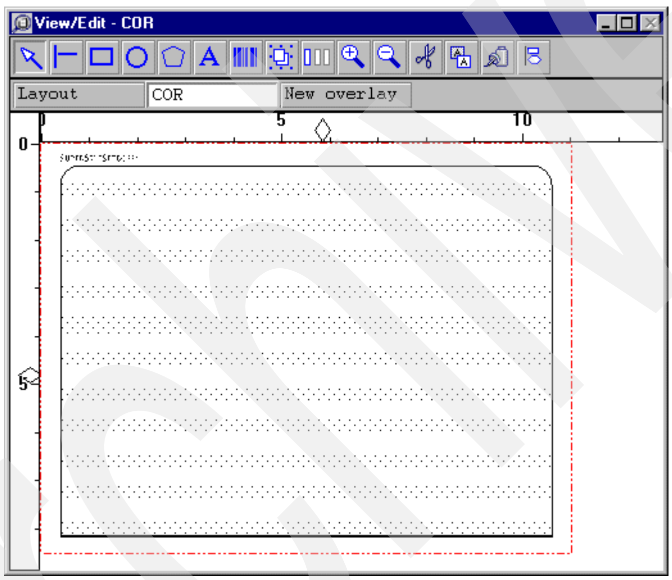


Figure 42. Sample overlay for COR simulation

2.9.5.3 Creating a layout for landscape printing

As with the overlay, the data that you map on the page using the layout editor is positioned and oriented based on the origin of the page. If you start your project by creating the landscape overlay, and then switch to Layout, the overlay appears sideways. This is rectified by rotating the layout. Figure 43 illustrates the relationship between the origin of the page and the placement of data for portrait and landscape orientation.

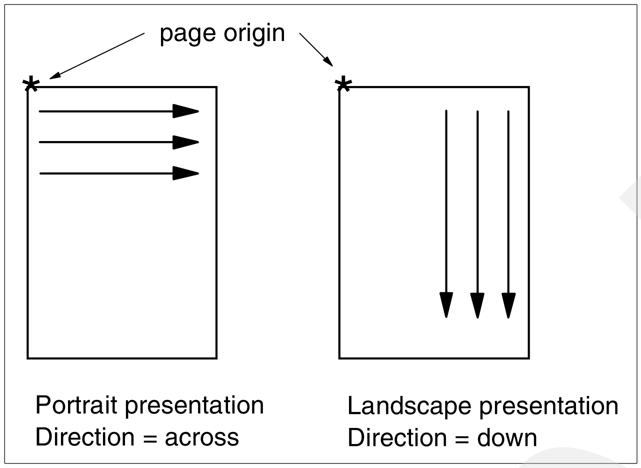


Figure 43. Portrait and landscape presentation

To set up your layout for landscape printing, select **Edit-> Page properties ...**. In the case of a letter size application, you see an *X* value of 8.5 inches and a *Y* value of 11.0 inches. Do *not* change these. For Direction, select the **Down** radio button. The X and Y values switch for you. Figure 44 shows the panel you see when you do this.

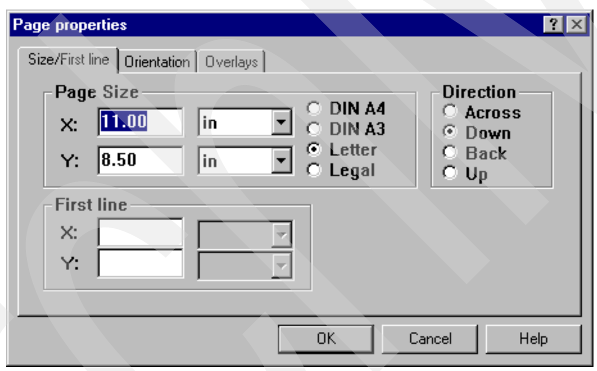


Figure 44. Page properties for landscape layout

You do not have to select the Orientation tab from this window unless you are printing on a continuous forms printer. *IBM Page Printer Formatting Aid: User's Guide,* S544-5284, describes the relationship between the values for *Direction* and *Orientation*.

When you close the Page properties dialog, the page size in the View/Edit window should reflect these changes, and the overlay should be presented in the correct orientation.

2.9.5.4 Layout specifics for COR project

There are a number of things you must be aware of when creating a layout for a landscape project (in particular, one that simulates the COR function):

- If the original application is set to print at 10 characters per inch (cpi), COR converts that to a font that prints at 13.3 cpi. When selecting a font in Infoprint Designer, use a fixed pitch font (such as Courier or Gothic Text) and select **9** points for Size. Use the information in Table 12 to determine the correct point size depending on the original spooled file cpi setting.

Table 12. Font size selection for COR

Original cpi	COR cpi	point size
10	13.3	9
12	15	8
15	20	6

- The interline spacing used by COR is 70% of the standard spacing. This works out to 0.1167... inches (or 8.5714... lines per inch). To avoid any chance of rounding errors, you may choose to use the *pels* units of measure. Infoprint Designer uses 240 pels per inch. This number is not affected by the selection of Host Printer Resolution in the General tab in the Preferences dialog. At 6 lines per inch, standard output measures 40 pels. With COR, an interline spacing of 28 pels is used.

Note

It is important to set the Infoprint Designer Preferences for **Vertical unit** and **Line spacing unit** prior bringing in the sample data and working on the Layout portion of the project. Select **Edit-> Preferences** and click the **Units and Grids** tab. Figure 45 illustrates the dialog box. If you do not do this, you may find “out of page boundary” errors when uploading your project to the iSeries. See 2.9.5.7, “Boundary errors” on page 75.

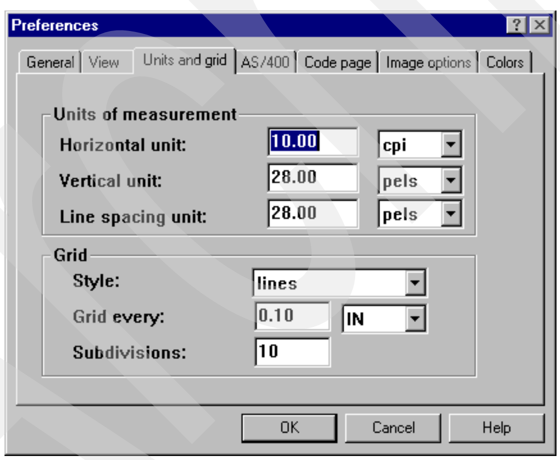


Figure 45. Setting preferences for units and grids

- After you map the data to the View/Edit window, click any line of data but the first, and verify the value for *s:* in the positioning section of the toolbar. An example of this is shown in Figure 48. If the incorrect value is displayed, it probably means that the Preferences were not set correctly. It is best to fix the preferences, refresh the sample data, and redo the mapping.
- When mapping the data, ensure that you select all of the records and the entire length of the records at one time. This ensures that any font changes you make to the first field affect the entire page and that the selected line spacing is used throughout.

If your data does not allow for this, you can fix the layout after performing the initial mapping. Left-click the first line of data and then right-click to bring up the Field properties window. Click the **Printline properties** button. On the Printline properties window, change the value for Repeat: to reflect the number of lines you want to map (for example, 66).

For example, the sample COR project was created using an iSeries job log that had 60 lines of data per page. By modifying the project to support up to 66 lines, it generates resources that can be used by a wider variety of iSeries reports.

- If you change the repeat value, you may find that there are extra unused printlines in the page definition. It is a good idea to look at the page definition tree. It should be similar to the page definition shown in Figure 46.

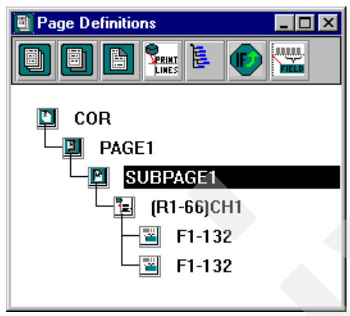


Figure 46. Page definition for sample COR layout

- Note the following points regarding this page definition:
  - If you mapped all of the records as one repeat group, there should be one printline on the subpage indicating the desired number of repeats (in this example (R1-66)). Any extra printlines must be deleted.
  - The CH1 indicates that this printline is activated any time there is a “1” in the first position of the data record. This ensures that a new page is started when required. Make sure you have not modified or deleted this setting in the course of making other changes.
  - There are two fields mapped in the one printline definition.
  - One field (probably the second) was generated when you dragged and dropped the map data. It should reflect the correct line length. If it requires adjusting, right-click the field entry and change the value in the Length: box.
  - The other field is generated automatically by Infoprint Designer and references the DFLT suppression. See 2.8.3, “Infoprint Designer and suppression” on page 54, for more information on the DFLT suppression. To determine which field is which, right-click one of the fields to bring up the Field Properties window. Then, click the **Suppress** button. The Name: field contains the value DFLT for the Suppressed fields; it is blank for the fields you map intentionally. If, and only if, all lines of your original spooled file are to be mapped, you may delete the Suppressed fields.

Figure 47 shows an example of the completed COR project using sample data from a job log. In this view, the third record was selected to show the *s:* value of 28 pels for interline spacing in the toolbar. The entry in the toolbar confirms that records 1 through 66 are mapped together, are triggered by Channel 1, and that a field from 1 to 132 bytes is mapped.

We recommend that you create a similar landscape project before you attempt one that is more complex.

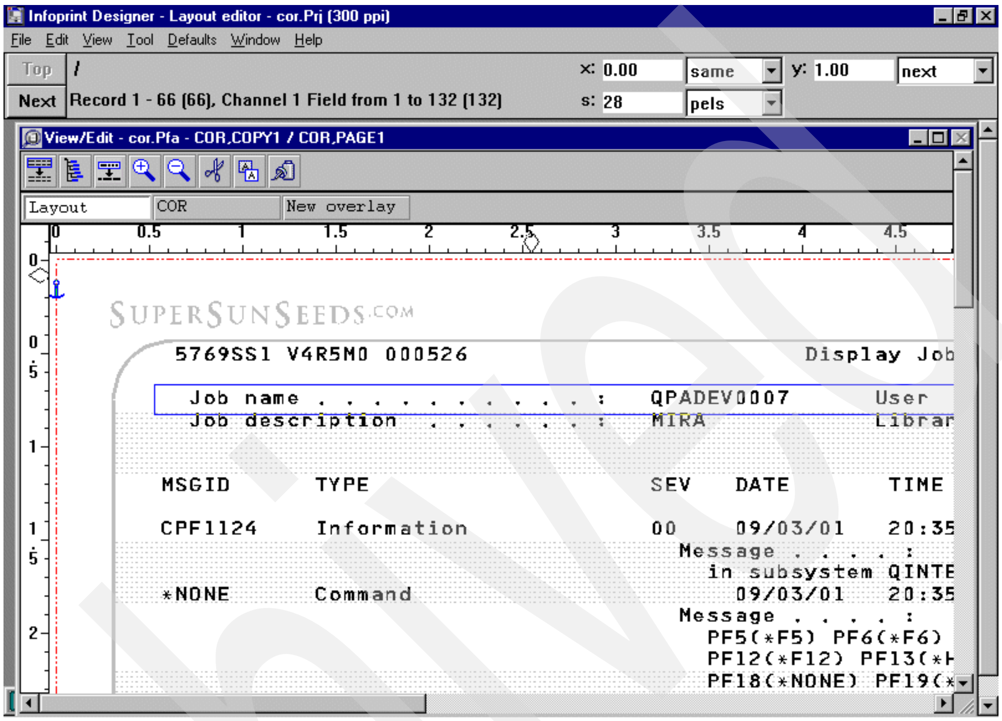


Figure 47. Completed COR project

2.9.5.5 Using the COR project on your iSeries

You can use this project with many iSeries server printer files that default to 132 columns by 66 lines. This example illustrates using it to format a job log listing:

1. Determine the printer file name for your listing. In the case of a job log, the printer file used is QPJOBLOG.
2. Enter a command to override the printer file with the parameters for using the COR project:

   ```
   OVRPRTF QPJOBLOG DEVTYPE(*LINE) FORMDF(COR) PAGDFN(COR) DUPLEX(*FORMDF)
   ```

   If you choose to use this permanently, make a copy of the QPJOBLOG printer file in a user library and use the CHGPRTF command to make the changes.
3. Enter the command to create a printout of your current job's job log:

   ```
   DSPJOBLOG * *PRINT
   ```

   The job log generated should print using the new form definition, page definition, and overlay.

2.9.5.6 Variation of a theme

In the previous example, not much was done beyond simulating a function that the iSeries already performs quite easily. This apparently redundant exercise was performed because, once you understand how to create a relatively simple landscape project, you can use the features of Infoprint Designer to take it to the next level.

Using multiple fonts

As you may have discovered while working on this or other projects, records that are handled in exactly the same manner are grouped as repeats within one printline. If you want to use a different font for the heading section of a report (compared to the body), you must define two separate printlines. Perform this as two separate mapping operations.

If you make a mistake, you can often fix it from the page definition tree by adjusting the repeat values or erasing extra printlines.

Use a high zoom value to align the different sets of data correctly. An alternate approach is to perform some calculations first so you can precisely position the elements.

Conditional processing

Some reports may have different formatting requirements for a heading page at the beginning or a summary page at the end. You may use Infoprint Designer's conditional processing feature to identify those pages and handle them accordingly.

A large report may have logical sections within it for different regional or departmental information. You select the When Change radio button in the condition definition to trigger the selection of a different copy group that uses a colored paper from a different drawer.

2.9.5.7 Boundary errors

One common problem encountered during testing was the occurrence of a message such as:

```
(R1-66)CH1 position exceeds the logical (PAGE1) boundary
```

This message is generated at the time the project is uploaded to the iSeries server. The cause of this is likely incorrect settings for line spacing on the Printline or individual field properties. This is often a result of not setting the Units and Grids tab under Edit-> Preferences correctly, before beginning the layout. However, it is possible to rectify the situation after the fact.

To bypass this error, check the Line Spacing settings in Printline properties and in Field properties. In all cases, they should be set at the correct value for your application (for example, 28 pels). To do this, right-click the **Printline** and select the **Orientation** tab. The Line Spacing setting appears on that window as shown in Figure 48. Change the value if necessary. Repeat these steps to check the Line Spacing value for the two fields.

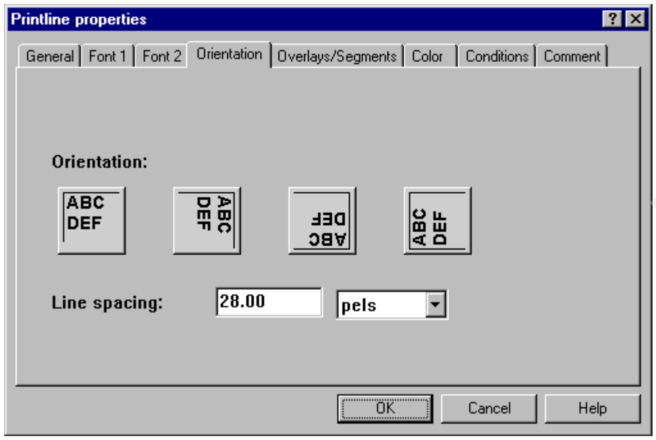


Figure 48. Changing line spacing for a Printline

After you make these changes to the line spacing, minor adjustments may be required to the position of the first line of print.

2.9.6 Multi-up barcode labels

The example in this section shows how Infoprint Designer may be used for a multi-up application, which means that many logical pages from the original application are placed on one physical page. This example also demonstrates the use of barcodes.

> **Note**
>
> This project is not included in the initial shipment of Infoprint Designer. It can be obtained by ordering the most current PTF for the product.

2.9.6.1 Project description

Super Sun Seeds prints barcode labels to place on seed packets. These labels are printed four to a page in a landscape orientation. Each label has a product number printed as a barcode, the product name, quantity, and a short description of one to three lines. The spooled file is generated with the information for each label as if it were to be printed on a separate page. This is referred to as the *logical page*.

2.9.6.2 Overlay creations for SEEDS project

Begin a new project, *SEEDS*, starting with the Overlay definition. Select **File-> Overlay setup-> Size** to change the overlay Orientation to 90.

Start the overlay by drawing lines to separate the quadrants of the page. These help you position the other elements as the project is developed. If you want, you may delete them when you are done. Draw other lines and boxes as needed. Design all the elements for one of the labels and then use the group function to copy the complete label to make the other three labels. If you are using any page segments, ensure you use the rotated versions.

2.9.6.3 Mapping data for the first label

When the overlay is complete, save your overlay and then click the **Layout** button. Select the source for your sample data. In this project, we used the PC file *seeds.dat*. When you are presented with the Line data options window, notice that the page length defaults to 66. Change this to “4”. This corresponds to the maximum number of data records that appears on each logical page. Doing this prevents unnecessary printline generation.

Figure 49 shows the sample data as it appears when it is first brought into the project. For the benefit of those reading hardcopy from a monochrome printer, note that only the data for the first label (HIGH ALTITUDE WATERMELON) is in blue. Subsequent labels, or logical pages, are in displayed in red.

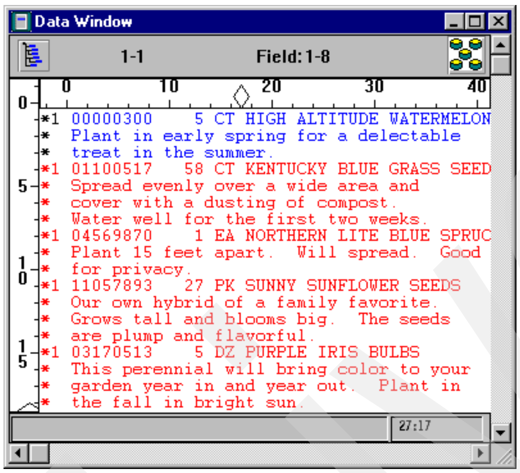


Figure 49. Sample data for label project

Before you map any data onto the page, you must set it up for landscape-oriented printing. Select **Edit-> Page properties** and click the **Down** radio button for Direction.

You are now ready to start mapping the fields. The first page of this file only contains two lines of description. It is better to use the second page (KENTUCKY BLUE GRASS SEEDS), which has the maximum number of description lines on it. Click the **Next** button on the toolbar to activate this page.

Drag the quantity and product description fields onto the first label and set the fonts for each of these fields.

Drag the item number onto the first label and right-click to bring up the Field Properties window. Click the **Barcode** button. The following list describes the settings used in this example:

- On the General tab of the Barcode options window, you must enter a reference name for your barcode. This allows you to reuse these same settings for other fields without redefining each option.
- Select a Barcode type. In this example, *CODE128* is used.
- HRI stands for Human Readable Interpretation. Select this tab to control whether the data is to be printed above or below the barcode lines (this also specifies which font is to be used).
- On the Size tab, a height of .5 inches is selected. The element width is set to 20/1000 of an inch.

Click **OK** on the barcode dialog, and the item number appears in the barcode on the overlay.

Highlight the three lines of the description and drag them over to the label in one operation. Change the font and set the interline spacing so that it fits nicely in the box.

At this point, the first label is complete and it looks something like the example shown in Figure 50. Use the **Top** and **Next** buttons to page through the data to verify that all of the pages fit nicely.

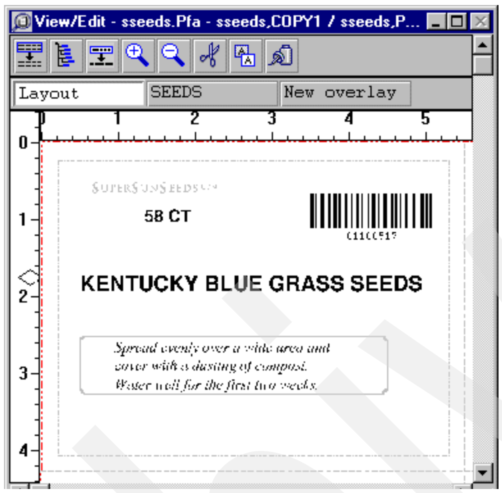


Figure 50. Completed first label

2.9.6.4 Mapping the remaining labels

The next step is to duplicate this mapping for the other three labels. One way to do this is by brute force. Namely, repeat all the dragging, mapping, and setting of fonts and other properties. However, it is easier to make copies of the work performed by working on the Page Definition tree.

Open the Page Definitions window. The page definition for the first label should look something like the example shown in Figure 51.

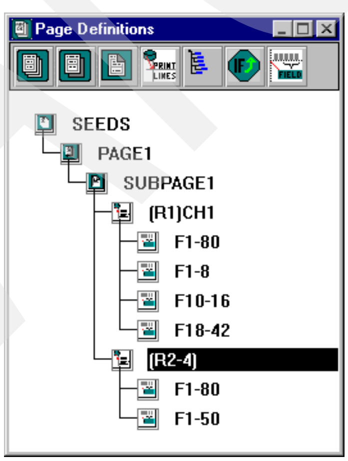


Figure 51. Page definition for first label

There should be two printline definitions. Both have a field of 1-80 for the suppressed fields. The first printline references record one and have a Channel code defined. Three data fields are mapped in it. The second print line repeats from record two through four and maps the description fields.

If extra printlines appear, you must delete them. Click them and then press the Delete key.

Before you begin duplicating the printlines, you may find it easier to work if you collapse the fields within the printlines. Double-click the icon for each printline. The Field definitions disappear and a plus sign appears beside the printline icon.

To copy a given printline, hold down the Ctrl button on the keyboard. Click and drag the icon of the printline you want to copy to the bottom of the tree. When you do this, the fields are expanded. It is a good idea to collapse each printline as you go along.

Copy the printline for (R1)CH1 and place it at the bottom of the tree. The new printline is labeled (R5)CH1. Next, copy the printline (R2-4) to create (R6-8). Repeat this until you have four pairs of printlines (representing a total of 16 records). The resulting page definition tree should look like the example in Figure 52.

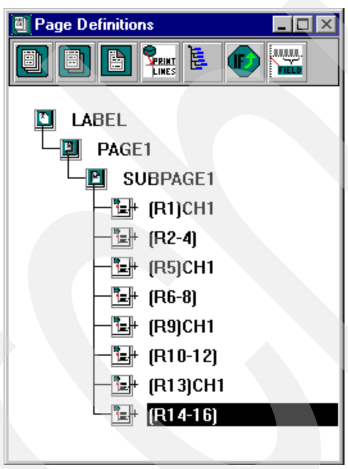


Figure 52. Page definition after copying printlines

While you are making copies of the records in the page definition tree, you may have noticed two things. In the Data window, the records for the first four labels are now blue. The layout in the View/Edit window has become quite congested. Do not be alarmed if it looks like the example in Figure 53.

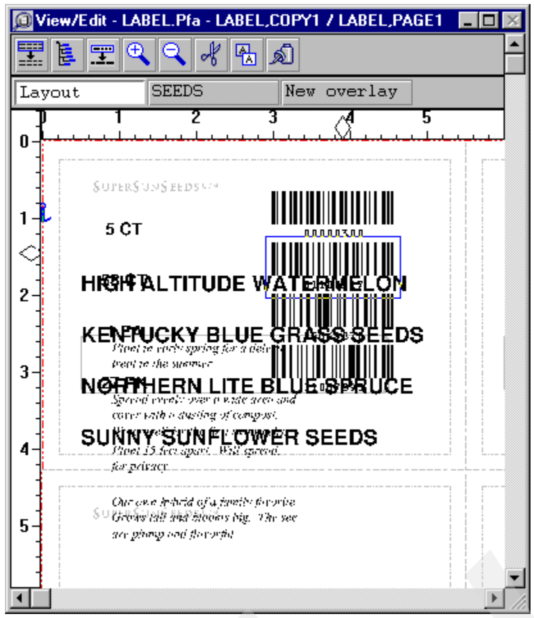


Figure 53. View/Edit window after copying printlines

There are two methods to move the label data onto the corresponding quadrants on the page. One is to use drag and drop, and the other is to enter the locations in the position fields on the toolbar.

Drag and drop method

Click one of the data fields from the first record of the second label. In the example in Figure 53, the barcode for the Kentucky Blue Grass record is selected. Notice a small blue anchor displayed near the left margin of the page. This is the reference point for that printline. Holding down the left mouse button, drag the anchor so that the data for the second label fits in the top right quadrant of the page. Notice that the data for the third and fourth labels is dragged along as well.

Repeat this process for the other two sets of records.

Numerical method

This method involves a little more arithmetic than the drag and drop method, but is more accurate. It involves direct manipulation of the x and y coordinate values for the printlines in the position section of the toolbar. These are the upper pair of numbers that you see when you click on an element.

By default, each printline prints at the same x value as the previous printline, and the y value increases based on the line spacing settings in the Printline properties. The x and y values appear as *same* and *next* respectively. You must determine where the first printline is anchored and then add 5.5 and 4.25 inches (half of 11 by 8.5, for Letter size paper) to the x and y coordinates of the other three printlines.

Select one element from the first record of the first label. To view the absolute position of that record's anchor change, the units in the printline coordinates on the toolbar from *same* and *next* to *inches*. The values for x and y should now appear as 0.00 and 0.17 inches respectively. You can now click on elements of the other records and change their anchor position accordingly. Make sure you click on elements from the first record of each of the labels, *not* the description records. You can use the information provided in Table 13.

Table 13. X and Y coordinates for each label

Label	X coordinate	Y coordinate
1	0	0.17
2	5.5	0.17
3	0	4.42
4	5.5	4.42

Using either method, or a combination of the two, the final outcome should look something like the example in Figure 54. Note the new positions of the anchor for the Kentucky Blue Grass Seeds label.

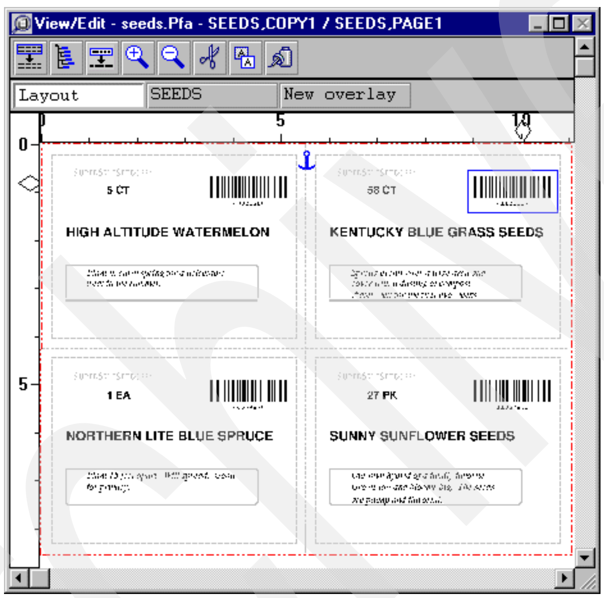


Figure 54. Completed label project

2.9.6.5 Using the SEEDS project on your iSeries

To print the seeds project on the iSeries, follow these steps:

1. Create a physical file on the iSeries for the data:

   ```
   CRTPF mylib/SEEDS RCDLEN(81)
   ```

2. Use FTP or Client Access to copy the data from the PC file that is used for this project into the physical file. The file must be translated from ASCII to EBCDIC during the transfer process. In this example, FTP is used.

   a. Open a DOS window. Select the directory that contains the sample seeds data file:

   ```
   cd Designer\DATA
   ```

   b. Begin the ftp session with your iSeries:

   ```
   C:\Designer\DATA>ftp BLDASL1
   ```

   c. Enter your user ID and password when prompted.

d. Check to see if the destination iSeries library is correct:

```
ftp>pwd
```

e. If you want to use a different target library, you can change it:

```
ftp>cd mylib
```

f. Copy the Seeds.dat file into the iSeries physical file SEEDS, member SEEDS:

```
ftp>put seeds.dat seeds.seeds
```

g. End the ftp session:

```
ftp>quit
```

3. Create a printer file that uses the project:

```
CRTPRTF SEEDSPRTF DEVTYPE(*LINE) FORMDF(SEEDS) PAGDFN(SEEDS) CTLCHAR(*FCFC)
```

4. Copy the data from the physical file to the printer file:

```
CPYF SEEDS SEEDSPRTF
```

This generates samples of the Super Sun Seeds labels.

2.10 Putting the application into production

To utilize the AFP resources you have created and transform the original application, you must override or change the printer file to:

- Generate line data
- Refer to the page definition AFP resource name
- Refer to the form definition AFP resource name
- Use the Drawer selection from the form definition
- Use the Duplex selection from the form definition

You must also ensure that the AFP resources can be located by the print writer (for example, PSF/400).

2.10.1 Changing printer file to *LINE data

The printer file DEVTYPE parameter dictates the format of the data stream the iSeries will produce. The default is *SCS. For printing using an Infoprint Designer-enhanced application this must be changed (or overridden) to *LINE. You can identify the printer file from the spooled file attributes. Enter option 8 (Attributes) next a list of spooled files on your system. You would then use the following command to invoke this change:

```
CHGPRTF printer_file_name DEVTYPE(*LINE)
```

2.10.2 Overriding the printer file to use *LINE data

In Figure 55, the printer file is QSYSPRT in library QSYS. This is an example of where an application-specific printer file has *not* been used. The “Q” prefix tells us this is an IBM-supplied printer file and we do not recommend changing this printer file (otherwise the screen prints and the other output will be affected). Instead you must track down the programmer and the step in their code that *overrides* the QSYSPRT printer file. You can then arrange for the device type parameter (and the page and form definition parameters) to be altered.

```
                        Work with Spooled File Attributes

Job  . . . . . . . . :   QPRTJOB        File . . . . . . . . :   INVSCS
  User . . . . . . . :     SIMON          Number . . . . . . :     1
  Number . . . . . . :     235527

Device type  . . . . . . . . . . . . . :   PRINTER
Printer device type  . . . . . . . . . :   *SCS
Device file  . . . . . . . . . . . . . :   QSYSPRT
  Library  . . . . . . . . . . . . . . :     QSYS
User-specified data  . . . . . . . . . :   SSS.COM
Accounting code  . . . . . . . . . . . :   982806
Program that opened file . . . . . . . :
  Library  . . . . . . . . . . . . . . :
Date file was opened . . . . . . . . . :   03/26/01
Time file was opened . . . . . . . . . :   15:53:41
Date file was last used  . . . . . . . :   03/26/01
System where file created  . . . . . . :   BLDAS45
User who created file  . . . . . . . . :   SIMON
                                                                       More...
Press Enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F13=Change
```

Figure 55. Identifying the printer file that created the sample spooled file

2.10.3 Adding page and form definitions to the printer file

Use either the CHGPRTF or OVRPRTF command as described above to add the page and form definition names, and their library, to the application printer file.

2.10.4 Duplex and drawer selection

PSF/400 uses the value specified in the printer file for Duplex and Drawer selection over any such specifications in the form definition. Change or override the printer file to select DRAWER(*FORMDF) and DUPLEX(*FORMDF) to make sure the design you created in Infoprint Designer is activated correctly.

2.10.5 Library lists

As with all AFP resources, the iSeries must be able to find the resources, for example, in an accessible library list. If you get missing overlays, logos, etc. from your output, the most likely cause is that the library to which you uploaded the resources is not in your library list.

For interactive work (probably testing), you can add any resource libraries to your own library list (EDTLIBL). For batch application, add the libraries to the printer (Device Resource List in the PSF configuration object) or to the job’s library list.

2.10.6 Creating, copying a printer file with *LINE data and placing it higher in a library list

If it is not possible to change the printer file and you do not have access to the system code that runs the application, it may be possible to locate the printer file as above, then take a copy of this (or create a new one from scratch). It must be the same name as the original (because the application will refer to it by name). However if we store it higher up in the library list, with our page and form definitions and *LINE data parameter added, PSF/400 then locates and uses this printer file first.

2.11 Problem quick-fixing

This section offers a quick-fix guide to problems you may experience.

2.11.1 Library profile must exist

Infoprint Designer stores all the library path names in the Default.lbp (or Raster.lbp file; see 2.2, “Setting up the PC libraries” on page 33). It expects to find this text file in c:\designer\userisis. If for some reason this file is moved or missing, symptoms include:

- No fonts, or other AFP resources, available to work with
- Blank name displayed for Library in the Libraries dialog box
- Messages such as the one shown in Figure 56

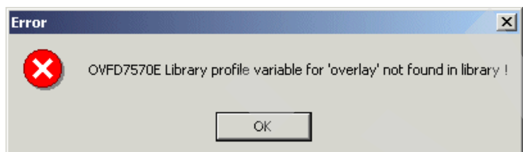


Figure 56. Failure to find default library for overlay resource

To correct this problem, use **Edit-> Change library** and select the correct library profile file.

2.11.2 Missing resources

Figure 57 shows the results of opening an overlay referencing a page segment (separate AFP resource), the location of which is not known to Infoprint Designer. To correct this, either copy the AFP page segment into the correct Infoprint Designer resource library, or add its path information to the library profile. We look at both examples.

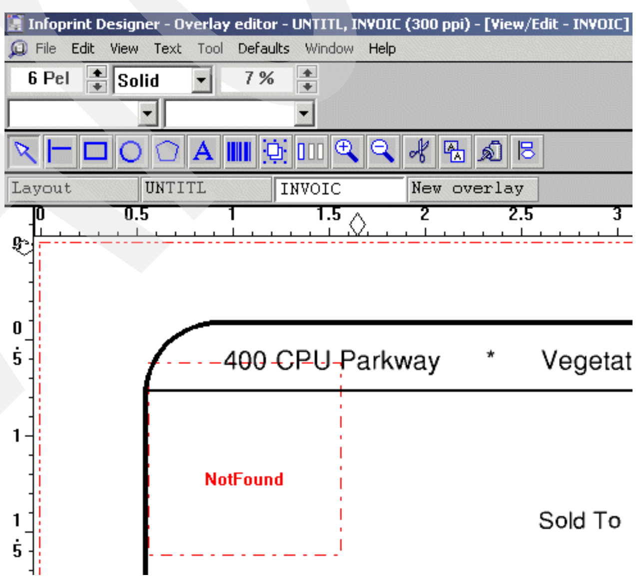


Figure 57. Page segment missing from imported overlay

2.11.2.1 Copying missing resource to Infoprint Designer library

To place a copy of the missing AFP resource into the Infoprint Designer resource library, simply use Windows Explorer to copy/paste the AFP resource into the appropriate PSEG library (PSEG300 in this example). Then right-click the red boxed area that represents the missing resource. This brings up the Page segment parameters dialog, from where you can click **Refresh**. The page segment should then appear.

2.11.2.2 Adding path information to Infoprint Designer

This method leaves the AFP resource where it is (perhaps on a network drive) and tells Infoprint Designer in which library the resource is to be found. To add the directory where the resource is location to the Infoprint Designer library, you are presented with a dialog similar to the one shown in Figure 58. Enter the path of the customer library and the file extension in the Path entry field and click **Append**. Make sure you have the correct resource tab selected, which in this case is **Overlay**.

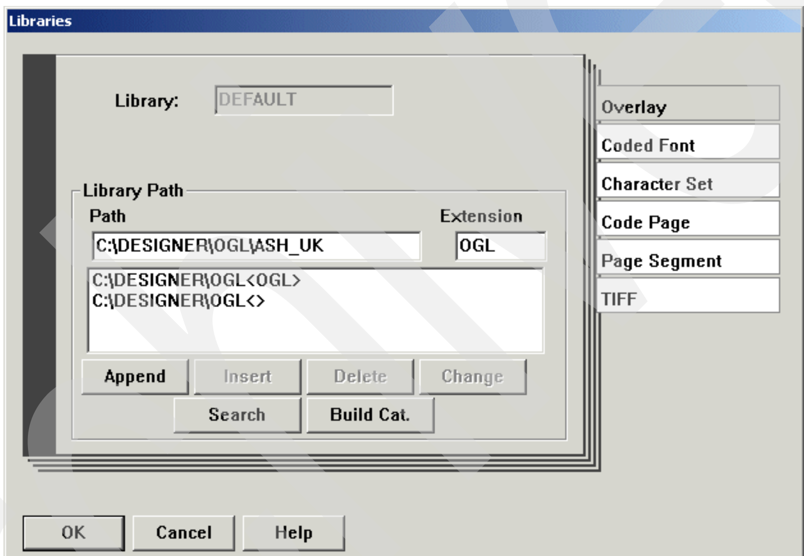


Figure 58. Adding the path and file extension of a resource to the library profile

We recommend you use the latter method, for example, add the path and file extension of the resource to the Infoprint Designer resource library. This has the following advantages:

- Resources are kept in the location of your choice
- Resources may be backed up more easily
- Resources are kept separate from the IBM-supplied programs and code
- Extra/alternative resource suffixes may be defined and added (such as those in Table 11 on page 35).

However, you should be aware that there is a 1024-character limit to the list of path names for any one resource type.

2.11.3 Data shown on constant back overlay

If you are using a Constant Back overlay, as described in 2.9.2, “Adding a constant back to a subgroup” on page 60, you may notice the following problem. As you page through the Data Window, using the Next and Back buttons, you see data displayed along with the back overlay. This is because we are in effect trying to show both the front and reverse sides of the document on the PC display. Unfortunately Infoprint Designer only displays pages using the data in the sample file, so it superimposes the data from the front side on the constant (back) overlay. Rest assured that the data intended for the front sides of the job prints correctly and no data prints on the constant back sides, so the pagination of your job is correct when you send the job to the printer.

To work around this, you can temporarily prevent Infoprint Designer from displaying the back overlay once you verify that they are being included at the correct points. To do this, deselect **Merge windows** in the Window menu option.

2.11.4 License issues and information

Only one copy of Infoprint Designer can be used on an iSeries server at any one time. However you can install multiple copies of the client on multiple PCs. Infoprint Designer can only be used while it has a "live" session with the iSeries server. If another user starts an instance of the client, the software runs in "Demo" mode, where file save and the Upload functions are disabled.

There is a Programming Request for Price Quotation (PRPQ) offering from IBM to support multiple simultaneous users of Infoprint Designer, should you feel this is necessary. (A PRPQ may be an IBM-modified program product offering.) The program number for this PRPQ is 5799-GPW.

Chapter 3. OS/400 V5R1 printing enhancements

This chapter describes the new printing enhancements at OS/400 V5R1. These may be grouped into the following categories:

- Internet Printing Protocol (IPP) Server
- Printer file and DDS enhancements
- Unicode
- Java print support
- Miscellaneous print enhancements

3.1 Internet Printing Protocol (IPP) Server

With V5R1, OS/400 becomes an IPP-enabled print server. IPP is a standard that is likely to replace the default TCP/IP printing protocol of line printer requester/line printer daemon (LPR/LPD). Most iSeries users know the latter through implementation of a remote output queue, usually printing to an ASCII printer on the LAN. LPR/LPD has a number of limitations, including no page range support and little or no printer status or error recovery.

With IPP, the end user has a single interface from which they may submit and manage print jobs over TCP/IP to any iSeries-attached printer. The process is the same for print jobs from the local area network, intranet, or the Internet. This is particularly convenient for users working remotely, for example a traveling sales representative or a home-office worker.

3.1.1 IPP as a standard

The Internet Printing Protocol was developed by the Printer Working Group, (`http://www.pwg.org/`), a consortium of companies, including IBM, organized to promote printing standards.

IPP is transported over HTTP/1.1 and uses port number 631. The application layer in which HTTP operates is at a higher level than the standard LPR/LPD function. IPP is a client-server protocol; an IPP client submits a print request to an IPP Server. The latter is either an IPP-capable printer or a print server with IPP capability. The target IPP printer/print server would usually be defined by an Internet URL, but could also be defined directly by an IP address or via a directory that provides mapping to the URL or IP address.

3.1.1.1 IPP object types

The IPP standard defines two basic object types:

- An IPP printer object
- An IPP job object

These objects can each have *operations* (performed by them and requested of them) and *attributes* (their current and default state). Operations include not only the actual printing but also a variety of other tasks such as getting the printer status, holding a print job, releasing a print job, restarting a print job, cancelling a print job, and getting a print job list. Attributes include printer language, print orientation, paper size, and the number of copies.

3.1.2 IPP clients, servers, printers

Table 14 shows various clients and servers that use IPP technology. Note that some IPP clients also have IPP Server capability.

Table 14. Examples of IPP clients, servers, and printers

IPP clients	IPP Servers	IPP printers
Windows 95	OS/400 V5R1	IBM Infoprint Color 8
Windows 98	OS/390 V2R8	IBM Infoprint 21
Windows 2000		IBM Infoprint 70
OS/390 V2R8		
Infoprint Manager for Windows NT and 2000	Infoprint Manager for Windows NT and 2000	
Infoprint Manager for AIX	Infoprint Manager for AIX	

In this context, an "IPP printer" is one that can receive IPP print jobs without an intermediate IPP Server (in fact the server function is performed within the printer, although it cannot act as a server for other printers). As you shall see, you do not necessarily need an IPP-enabled printer to perform IPP printing. This is where the iSeries IPP Server comes in.

3.1.3 IPP on the iSeries

With OS/400 V5R1, the iSeries acts as an IPP print server. OS/400 is responsible for passing on the print request to an attached printer. This may involve a further print process, for example Host Print Transform (for printing to an ASCII printer) or PSF/400 (for printing to an AFP printer). But from this point on, the print process is "business as normal". It is important to realize that the target printer itself does not have to be IPP-capable, but it has to be attached to the iSeries server. In Figure 59, the iSeries printer could be any already-attached printer, even a twinaxial printer. All that is required on the IPP client is an appropriate printer driver.

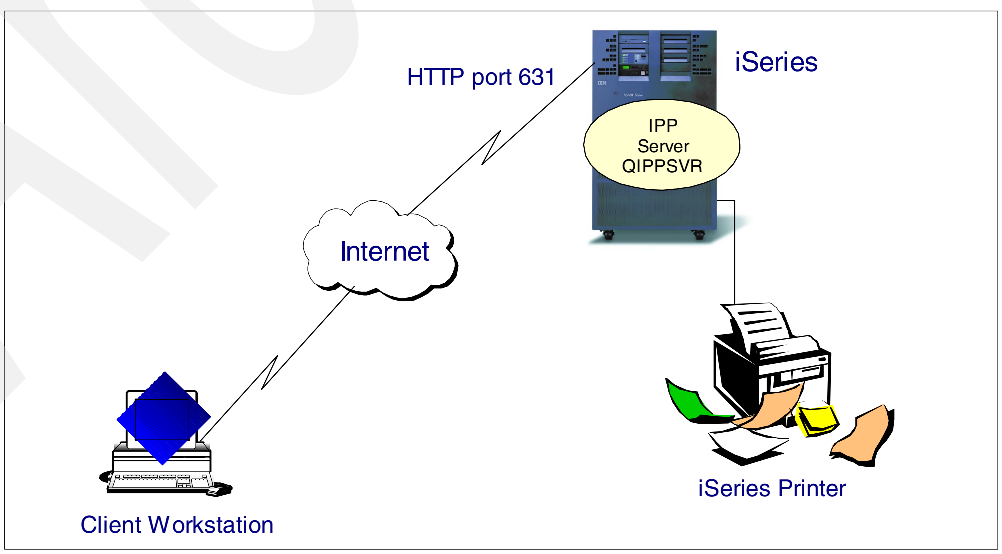


Figure 59. The iSeries as an IPP Server

The IPP client can request and obtain the status of the print job, but this can only be as good as what the status OS/400 itself can obtain about the print job. Examples of such requests could include:

- Inquiry about the capabilities of an iSeries printer
- Submission of a print job to this printer, with modified job characteristics such as number of copies, if required
- Inquiry about the status of the print job
- Management of the print job to completion

This is therefore a "Stage 1" implementation of IPP support on the iSeries (formally, the support is IPP version 1.0). At the present time, you cannot print from the iSeries (using it as an IPP client) via the Internet to an IPP-enabled printer. However note that IPP clients, such as Microsoft Windows 98 and 2000 and IBM Infoprint Manager for Windows NT/2000 and AIX, could be used. IBM also markets an increasing number of IPP-enabled printers such as Infoprint 21 and Infoprint 70.

3.1.3.1 iSeries IPP printer operations

OS/400 uses standard iSeries printers, output queues, and spooled files for IPP printing. Of these, the iSeries output queues are central to IPP printer operations. Examples of IPP printer operations required by the IPP standard include:

- Print-job
- Validate-job
- Get-printer-attributes
- Get-jobs
- Cancel-job
- Get job-attributes

The following optional IPP operations are also supported:

- Pause-printer
- Resume-printer
- Purge-job
- Hold-job
- Release-job
- Restart-job

It may not be immediately obvious is that the IPP Server on the iSeries allows IPP printer operations to access iSeries spooled files that were not created by the IPP Server. In other words, we mean the list of "normal" spooled files on that output queue. This raises the possibility of using IPP to manage spooled files from an IPP client by retrieving a list of current print jobs from the output queue.

3.1.3.2 iSeries IPP job operations

IPP job operations are performed on iSeries spooled files. Examples of job operations include:

- Cancel-job
- Get-job-attributes
- Hold-job
- Release-job
- Restart-job

3.1.3.3 IPP job template attributes

Job template attributes are the characteristics of the submitted IPP print jobs. On the iSeries, the equivalent would be spooled file attributes. Support for job template attributes, regardless of whether they are included in create requests, is optional in the IPP standard. Generally speaking, OS/400 supports most of these through iSeries spooled file attributes. See Table 15.

Table 15. Correlation between IPP job template attributes and actual iSeries operations

IPP job template attribute	Description	iSeries IPP-specific description
Job-priority	IPP value of 1 to 100	The IPP value is mapped into the iSeries spooled file "output priority" value of 1 to 9.
Job-hold-until	IPP possible values of: - No-hold - Indefinite - Day-time - Evening - Weekend - Etc.	The IPP values of “no-hold” and “indefinite” are mapped to the iSeries spooled file "hold spool file” attribute. The IPP values of “day-time”, “evening”, “weekend”, etc. are not supported. iSeries spooled files do not support an "allowable print time" attribute, even though iSeries output queues do have such an attribute.
Job-sheets	IPP possible values of: - None - Standard	The IPP values of “none” and “standard” are mapped to the iSeries spooled file "file separator" attribute if the iSeries output queue does not specify job separators. If the iSeries output queue specifies job separators, only the IPP value of “standard” is supported.
Copies	IPP integer value	The IPP value is mapped into the iSeries spooled file "copies" attribute. The iSeries has a limit of 255 copies.
Finishings	IPP possible values of: - None - Staple - Saddle-stitch - Edge-stitch - Staple-top-left - Etc.	The IPP value is mapped into the iSeries spooled file "corner staple", "edge stitch", and "saddle-stitch" attributes. The physical device must support these operations. Supported by spooled files of type AFPDS.
Page-ranges	IPP supports single or multiple page ranges such as: 1-3, 5-7, 9-11	iSeries spooled files only support one page range. The first IPP page range will be used, any additional pages ranges will be returned to the client as unsupported attributes. Supported by spooled files of type AFPDS, SCS, and LINE.

IPP job template attribute	Description	iSeries IPP-specific description
Sides	IPP possible values of: - One-sided - Two-sided-long-edge - Two-sided-short-edge	The IPP value is mapped into the iSeries spooled file "print on both sides" (duplex) attribute. The physical device must support these operations. Supported by spooled files of type AFPDS, SCS, and LINE.
Number-up	IPP possible values of: 1 2 4	The IPP value is mapped into the iSeries spooled file "multiple up" attribute. The physical device must support these operations. Supported by spooled files of type AFPDS, SCS, and LINE.
Orientation-requested	IPP possible values of: - Portrait - Landscape - Reverse-landscape - Reverse-portrait	The IPP value is mapped into the iSeries spooled file "page rotation" attribute. The physical device must support these operations. Supported by spooled files of type AFPDS, SCS, and LINE.
Media	IPP media size values	The IPP value is mapped into the iSeries spooled file "measurement method", “page width” and “page length” attributes.
Media	IPP input-tray values	The IPP value is mapped into the iSeries spooled file “source drawer” attribute.
Print-quality	IPP possible values of: - Draft - Normal - High	The IPP value is mapped into the iSeries spooled file "print quality" attribute. The physical device must support these operations. Supported by spooled files of type AFPDS, SCS, and LINE.

3.1.4 Setting up the IPP Server for iSeries

The required steps to set up an IPP Server at OS/400 V5R1 are:

1. Confirm your system meets the IPP Server requirements.
2. Start the iSeries HTTP server.
3. Access the iSeries Tasks page through your Web browser.
4. Configure the IPP Server for iSeries.
5. Configure an IPP printer or printers.
6. Start the iSeries IPP Server.
7. Configure an IPP client.

3.1.4.1 IPP Server requirements

Although support for IPP is built into OS/400 V5R1 (option 3 of OS/400, 5722-SS1), the following products are also required to configure, manage, and administer the service:

- HTTP Server for iSeries (5722-DG1)
- IBM Developer Kit for Java (5722-JV1)

If you require printing over a secure connection, you also require SSL support, which is obtained using the following products:

- Digital Certificate Manager (option 34 of OS/400, 5722-SS1)
- Cryptographic Access provider (5722-AC2 or AC3)

3.1.4.2 Starting the HTTP server

To check the status of the iSeries HTTP server, use the command:

```
WRKACTJOB SBS(QHTTPSVR)
```

There should be two or more ADMIN jobs running in subsystem QHTTPSVR. If not, start the server by using the command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

You can also achieve the operations using Operations Navigator as shown in Figure 60. Click **Network-> Servers-> TCP/IP** and look for the status of a server called **HTTP Administration**.

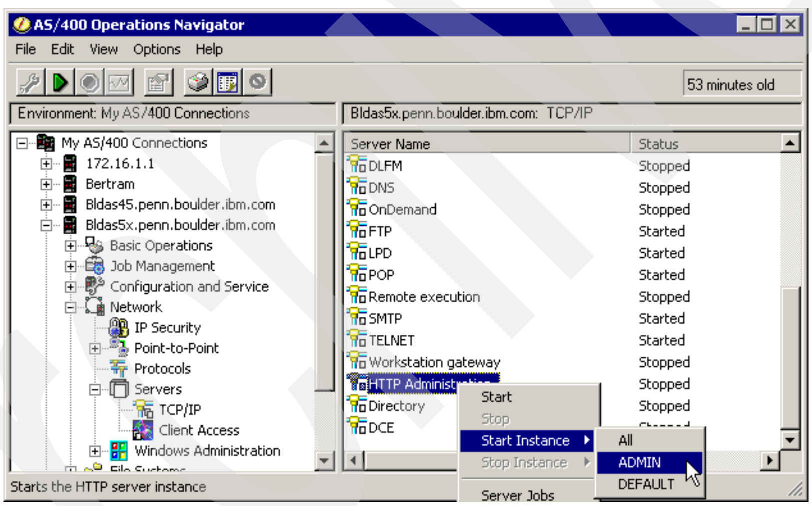


Figure 60. Starting the HTTP server using Operations Navigator

3.1.4.3 Configuring the IPP Server

Start your Web browser and enter the following URL to access the iSeries Tasks page (Figure 61):

```
http://your_system_name:2001
```

Your Web browser must support frames and JavaScript, and your iSeries user ID must have special authority *IOSYSCFG. You also need read/write authority to the following files:

- /QIBM/UserData/OS400/Ipp/conf/qippsvr-cust.conf
- /QIBM/UserData/OS400/Ipp/conf/printer.properties

You are prompted for your iSeries user ID and password.

Figure 61. iSeries Tasks Web page

Select the **IBM IPP Server** link. This starts the IPP Server Administrator. Next, click the **Configuration** link.

The Basic configuration menu option provides a page to enable secure sockets. See Figure 62. If you want your print data transmitted over a secure connection, you should enable SSL connections. You must select a unique port number on your system: 6310 is suggested. If you need to check for possible port conflicts, use the OS/400 command:

```
NETSTAT OPTION(*CNN)
```

Or you can use Operations Navigator and navigate to **Network-> TCP/IP Configuration-> Connections**.

When SSL connections are enabled, the IPP Server administrator registers the IPP Server with the Digital Certificate Manager. The application name registered is QIBM_IPP_QIPPSVR. Using the Digital Certificate Manager, you should associate a system digital certificate with the QIBM_IPP_QIPPSVR application. For non-SSL data, the IPP Server uses port 631.

Figure 62. Configuring the SSL connection in the IPP Server

Make sure that you have configured your SSL connection, if you want to use this, before you move on to the next step of creating the IPP printer descriptions. Also note that you must stop and restart the IPP Server again if you make any server configuration changes, including any IPP printer configuration (create/delete) changes.

3.1.4.4 Configuring an IPP printer

From the IPP Server Web page, click **Configuration-> Basic Configuration**. To create an Internet printer configuration, you can use either an iSeries printer or an iSeries output queue. The latter is provided for use with remote writers, for remote output queues that do not normally have an associated printer device description. It may be more logical to use a print device as the basis for the new IPP device, in which case an output queue of the same is used by the IPP Server. Whichever you use, the objects must already exist on the iSeries server.

Continue with the “wizard” steps. We recommend that you *do not* associate a specific printer file with the IPP printer, at least not during an initial setup. If you opted to use SSL, you can now select it for use with this particular printer. When prompted for an authentication method, select **Basic**.

You reach a page asking for the type of data to be transmitted. This largely depends on the target printer you have chosen:

- Select **AFPDS** if the printer is configured as *IPDS, AFP=*YES in its device description.
- Select **SCS** if the printer is configured as a native twinaxial printer (typically 3812 or an impact printer emulation).
- If you are sending PCL or PostScript data from your IPP client and the printer is capable of printing these data streams (for example a LAN-attached ASCII laser printer), select **Other**. This causes the spooled file data to be sent as

*USERASCII and the iSeries will not convert or modify it. An exception to the latter involves use of the Infoprint Server product. One of the functions enabled by this product is the automatic transformation of ASCII data into that of the target printer (AFP or PostScript). This is described in Chapter 7, “Converting PCL, PostScript, or PDF to AFP” on page 173.

The last few displays concern the naming of the new IPP device. You can use the same name as the iSeries printer or a different one. When the wizard finishes, the URL of the new IPP printer is displayed in the browser message frame (at the foot of the page), for example:

```
http://your_system_name:631/ipp/printers/
your_ipp_printer_configuration_name
```

This is the URL by which your IPP client will access (print to) the IPP printer.

3.1.4.5 Starting the IPP Server

Use the Administration page to start the IPP Server. Simply click the **Start** button (Figure 63).

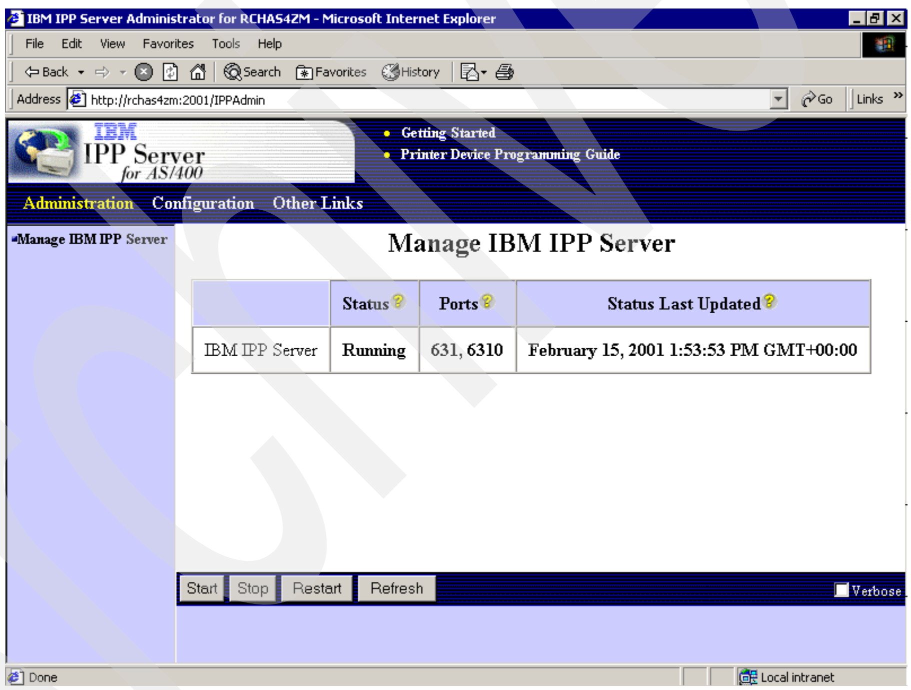


Figure 63. Starting the IPP Server

If the WRKACTJOB display (or Operations Navigator) is still displayed, you now see two IPP Server jobs (QIPPSVR) running in QHTTPSVR. The fully-qualified job name is also displayed in the browser message frame.

3.1.4.6 Setting up an IPP client

With the IPP Server running and a known IPP printer URL created, you can now configure an IPP client. For example, in Windows 2000, select **Start-> Settings-> Printers-> Add Printer-> Network Printer-> Connect to a printer on the Internet or on your intranet**. Enter the URL of your IPP printer configuration. You may be prompted for a user ID and password, and for a security certificate, depending on the level of security you chose earlier. If you receive a message about an “unknown” printer, click **OK**.

When the Add Printer wizard completes, you can use this printer to print to your iSeries printer using IPP. Remember that you need to use the appropriate printer driver for the target iSeries printer. The iSeries does not perform any data stream transformations to your output.

3.2 DDS and printer file enhancements

The functions available through OS/400 printer files have been expanded with V5R1.

3.2.1 CVTLINDTA

At OS/400 V5R1, there is a single printer file parameter change with the introduction of the Convert line data (CVTLINDTA) facility. CVTLINDTA causes the output from a printer file specifying *LINE data and a page definition to be converted to AFP before the data is spooled. The result is an AFPDS formatted spooled file.

The reason for including this option is because *LINE data spooled files:

- Cannot be viewed with the AFP Viewer within Operations Navigator
- Cannot be sent to an ASCII printer using Host Print Transform

An AFPDS spooled file (Figure 64) overcomes these limitations.

```
                            Change Printer File (CHGPRTF)

Type choices, press Enter.

Back side overlay:
  Overlay . . . . . . . . . . .   *FRONTOVL     Name, *FRONTOVL, *NONE, *SAME
    Library . . . . . . . . . .                 Name, *LIBL, *CURLIB
  Offset down . . . . . . . . .                 0-57.790, *SAME
  Offset across . . . . . . . .                 0-57.790, *SAME
  Constant back . . . . . . . .                 *NOCONSTANT, *CONSTANT, *SAME
Convert line data . . . . . . .   *NO           *NO, *YES, *SAME
IPDS pass through . . . . . . .   *DEVD         *DEVD, *NO, *YES, *SAME
User resource library list . . .  *DEVD         Name, *DEVD, *NONE...
               + for more values
Corner staple . . . . . . . . .   *NONE         *NONE, *BOTRIGHT...
```

Figure 64. Setting the CVTLINDTA option on the Change Printer File (CHGPRTF) display

Note that there is significant difference between the CVTLINDTA printer file parameter (provided free of charge with OS/400) and the CRTAFPDTA CL command (provided with the priced Infoprint Server for iSeries product). Table 16 illustrates these differences.

Table 16. Comparison of CVTLINDTA and CRTAFPDTA

CVTLINDTA	CRTAFPDTA
Printer file parameter	CL command
Acts on data to be spooled	Acts on existing spooled file

CVTLINDTA	CRTAFPDTA
Creates an AFPDS spooled file, on the OS/400 spool	Creates up to four physical file members – AFP output, indexing, AFP resources, and a merged AFP file – in the iSeries Integrated File System (IFS)
AFPDS spooled file may contain references to unresolved resources	Merged AFP file may contain resolved AFP resources or AFP indexes
Included with OS/400 at no charge	Included with the priced program product Infoprint Server for iSeries

You typically use CVTLINDTA for use with the AFP Viewer, perhaps while developing an AFP application. You can also use it to produce an AFP spooled file for printing to an ASCII printer (Host Print Transform cannot convert *LINE data printer files).

With the advent of Infoprint Designer for iSeries, it is likely that increasing numbers of customers will generate line data to use the AFP resources created by the Designer product.

3.2.2 Data Description Specifications (DDS) enhancements

DDS can be used to create more advanced printer files. Only an overview of the enhancements for OS/400 V5R1 is described here. For more detailed information, including syntax, refer to the V5R1 *DDS Reference* available online at the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm`

You may find the *DDS Reference* under **Programming-> Programming support** in the Information Center topics.

3.2.2.1 BARCODE

A total of 20 barcode symbologies and variations are now supported using this DDS keyword. They are listed here, together with their descriptive names:

MSI	MSI Data Corporation (Modified Plessy)
UPC-A	Universal Product Code-A
UPC-E	Universal Product Code-E
UPC-2	Universal Product Code 2-digit supplement
UPC-5	Universal Product Code 5-digit supplement
EAN-8	European Article Numbering-8
EAN-13	European Article Numbering-13
EAN-2	European Article Numbering 2-digit supplement
EAN-5	European Article Numbering 5-digit supplement
INDUST2OF5	Industrial 2 of 5 (Standard 2 of 5)
MATRIX2OF5	Matrix 2 of 5
INTERL2OF5	Interleaved 2 of 5
CODEABAR	Codabar
CODE128	Code 128
CODE3OF9	Code 3 of 9 (Code 39, USD-3)
POSTNET	US Postal Service Postnet (Postal Numeric Encoding Technique)
RM4SCC	Royal Mail 4 State Customer Code (UK)
AP4SCC	Australian Postal 4 State Customer Code
DUTCHKIX	Dutch 4 State Customer Code (Klantindex)
JPBC	Japan Postal Barcode

The barcode height may now be specified for most of these barcode symbologies, expressed either in print lines (1 to 9) or in centimeters or inches. The UOM parameter in the printer file determines which of the latter to use.

For V5R1, the performance of multiple barcodes printed on the same page has been improved.

3.2.2.2 BOX

Color support for the box lines has been added, using one of the five color models (see the description for the COLOR keyword below). Shaded boxes may also be created.

3.2.2.3 CCSID

Coded Character Set Identifier (CCSID) is used to “tag” data with information about its character encoding, character set and code page, character size, and any special coding requirements. A value of 65535 indicates that no conversion is done to the data; this is usually the default system value setting for QCCSID. For V5R1, additional CCSID values have been added to support UCS-2 (Unicode) printing. See 3.3, “Unicode” on page 100, for more information.

3.2.2.4 CDEFNT

CDEFNT invokes an AFP Coded Font (character set/code page combination). For scalable (outline) fonts, a width value may now be specified, giving anamorphic scaling to characters (where the height is different from the width). If the width is specified, the height must also be specified. The point size may also be expressed in fractions of a point. See Figure 65 for an example.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A*
     A          R REC1
     A            FLD1           8A     10 13CDEFNT(QFNTCPL/X0BRTR)
     A*
     A            FLD2          10A     11 13CDEFNT(QFNTCPL/X0BRTP +
     A                                       (*POINTSIZE 10.1))
     A            FLD3          10A     20 13CDEFNT(QFNTCPL/X0BRTP +
     A                                       (*POINTSIZE 5.0 3.0))
```

Figure 65. Various ways of specifying a coded font

Note that this is only possible for scalable (outline) fonts. They must exist on the iSeries host and the printer must be capable of receiving and printing such downloaded outline fonts.

3.2.2.5 COLOR

This field-level keyword creates color output, using one of the following five color models, of which the last four are new with V5R1:

- Color name
- RGB (red/green/blue)
- CMYK (cyan/magenta/yellow/black)
- CIELAB
- Highlight color

The color name model supports eight named colors: black, blue, brown, green, pink, red, turquoise, and yellow.

The RGB color model requires three integer values for the red, green, and blue values within a range of 0 0 0 (black) and 100 100 100 (white).

The CMYK color model requires four integer values, each expressed as a percentage from 0 to 100.

The CIELAB color model requires three values. The first is the luminance value (0 to 100.00). The second two values specify the chrominance differences, from -127 to 127.

The Highlight color model requires two values. The first is the required color, from 0 to 65535, and is device-dependent. A value of 0 uses the default color of the printer device. The second value indicates the coverage, as a percentage from 0 to 100.

Typically the Highlight color model is used with the IBM Infoprint Hi-Lite Color Printer (Model 4005-HCI). It can also be used to create gray scaling on monochrome IPDS printers such as the Infoprint 60, 62, 200, 300, and 4000.

Host Print Transform supports only the original color name model.

3.2.2.6 FNTCHRSET

This specifies an AFP font character set. The same enhancements for CDEFNT are also available for FNTCHRSET.

3.2.2.7 FONT

This specifies a printer-resident AFP scalable (outline) font. The same enhancements for CDEFNT and FNTCHRSET are also available for the FONT keyword.

3.2.2.8 IGCCDEFNT

This specifies an double-byte character set (DBCS) AFP coded font. The same enhancements for CDEFNT are also available for the IGCCDEFNT keyword.

3.2.2.9 LINE

Support for drawing lines in one of the five color models has been added at V5R1. See the COLOR DDS keyword section for more information.

3.2.3 DDS support for Record Format Line Data

Limited DDS support for printer files that specified a printer device type of *LINE was added at V4R3. This allowed customers to produce spooled files using many of the DDS functions but also with a standard AFP page definition. Page definitions are now architected to support “record formats” (which are a key part of DDS). The original line of unformatted data may be selectively split up and reformatted (for example, positioned over several lines, with different fonts and other attributes).

At V5R1, support was added to DDS so that when compiling a *LINE printer file, the name of each DDS record format is added to each line data record.

3.3 Unicode

Unicode is an encoding scheme that provides a unique number for *every* character that may be stored in a system, regardless of the language. It is defined by the Unicode Consortium, a public body that publishes the Unicode Standard. It is also known as UCS-2, UTF8, UTF16, ISO/IEC 10646, or CCSID 1208, depending on the context.

By way of comparison, it may be noted that familiar encoding schemes, such as ASCII, encode some 256 characters, where UTF-8 covers in excess of 60,000 characters. The lower range of UTF-8 characters coincide with those of ASCII single-byte characters. UTF-16 is used to provide a function approximately equivalent to DBCS.

Support for Unicode has been adopted by several operating systems, including Sun Solaris, Microsoft Windows NT and 2000, and OS/400. It is now further developed in OS/400 V5R1.

3.3.1 OS/400 implementation of Unicode

Unicode actually provides standards for various other character representations such as line layout and character shape. Currently the only support provided by OS/400 is for Unicode code points that point to font glyphs (an absolute, specific image of a character, for example the lowercase character "a", in raster 300-pel format in the Helvetica roman medium character set at 12 point). This interim support for Unicode is actually the UCS-2 subset of Unicode.

At OS/400 V5R1, the implementation of Unicode-encoded data is achieved by one of two methods:

- Unicode data can be mapped to selected single- or double-byte EBCDIC encodings before the data is placed on the spool.
- The AFP Unicode Font RPQ program (5799-GJJ): It provides a set of AFP Unicode fonts equivalent to those in the AFP Font Collection (5648-B45). The application must select a font from this program for any Unicode-defined fields in the printer file.

Unicode-encoded data in *SCS files are always converted to EBCDIC before spooling. But for files of type *AFPDS, *LINE, and *AFPDSLINE, the data may be converted in this way or passed directly into the spooled file. The implementation is only available with DDS printer files (using the CCSID keyword).

3.3.2 Examples of use

Use the file-, record-, or field-level keyword to specify that a G-type field supports UCS-2 level 1 data instead of DBCS-graphical data. Each UCS-2 character is two bytes long.

The format of the keyword is:

```
CCSID(UCS2-CCSID | &UCS-2-CCSID-field | *REFC
        [*CONVERT | *NOCONVERT]
        [alternate-field-length])
```

The UCS-2-CCSID parameter is required. Use the UCS-2-CCSID parameter to specify a CCSID that uses the UCS-2 Level 1 encoding scheme for this field. You can specify the UCS-2-CCSID parameter either as a number up to 5 digits long or as a program-to-system field. You must define the program-to-system field with a length of 5 and with the S data type.

The *CONVERT parameter is optional and specifies whether the UCS-2 data is converted to a target CCSID specified on the CHRID parameter of the CRTPRTF, CHGPRTF, or OVRPRTF commands. *CONVERT is the default. If you specify the CCSID keyword with *NOCONVERT, the UCS-2 data is not converted to the target CCSID.

If *NOCONVERT is active for a printer file whose DEVTYPE is *AFPDS, *LINE, or *AFPDSLINE, the application must also use one of the AFP Unicode migration fonts. If you do not specify an AFP Unicode migration font, the output is interpreted as single-byte data and will probably be unprintable.

If *NOCONVERT is active and the file DEVTYPE is *AFPDS, specify an AFP Unicode migration font character set and code page with the FNTCHRSET keyword for printer files. If the file DEVTYPE is *LINE or *AFPDSLINE, specify the AFP Unicode migration font character set and code page in the page definition for the printer file.

If *NOCONVERT is specified for a printer file whose DEVTYPE is *SCS, a diagnostic message is issued when the printer file is used, and the UCS-2 data is converted to the target CCSID.

The alternate-field-length parameter is optional and is valid only when you specify the CCSID keyword at the field level and the *CONVERT parameter is active. Specify the alternate-field-length as the number of UCS-2 characters.

When UCS-2 data is involved in an output operation and the *CONVERT parameter is active, the data is converted from the associated UCS-2 CCSID to the target CCSID. Generally, the length of the data will change when this conversion occurs. Therefore, you can use the alternate-field-length value to specify a printed field length that is different from the default printed field length. The default printed field length of a “G” data type field is twice the number of characters that are specified for the field length.

For other syntax rules, refer to the online *DDS Reference* at the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm`

You may find the DDS Reference under **Programming-> Programming support** in the Information Center topics.

3.4 Java print support

This topic is covered in Chapter 13, “Printing from Java applications” on page 275. However, a summary of the new Java print classes available with OS/400 V5R1 is outlined here:

- New Java classes for creating formatted documents and reports using eXtensible Style Language (XSL) Formatting Object (FO) technology

- New Java classes for creating Advanced Function Presentation and Printing (AFP) documents

These classes are included in the V5R1IBM Toolbox for Java product. For a good starting point, see the IBM Toolbox for Java and JTOpen site on the Web at:
`http://www.ibm.com/iseries/toolbox`

3.4.1 Java classes for creating documents with XSL FOs

These classes provide a higher level interface for building and printing reports and documents. The output document formats include HP-PCL and Adobe PDF.

3.4.2 Java classes for creating documents with AFP

These classes provide a programming interface for writing application data to an OS/400 output queue in the form of record format line data. The application can specify whether the data is converted immediately to AFP when the spooled file is created ("pre-spool") or whether it is written to spool as line data. In the latter case, PSF/400 would convert the data to AFP when the spooled file is printed ("post-spool"). AFP page and form definitions (formatting objects) would be used to format the line data into AFP. The creation and use of these objects is covered in Chapter 12, "Using form definitions and page definitions" on page 255.

3.5 Miscellaneous print enhancements

The following enhancements have been enabled at OS/400 V5R1 through feedback from customers:

- The maximum number of spooled files created per job increased from 9,999 to 999,999 through the QMAXSPLF system value.
- The ability to associate a data queue with an output queue is already part of OS/400. In addition to this data queue support, a data queue may be associated with a job, or the system itself, using the environmental variable QIBM_NOTIFY_CRTSPLF together with the ADDENVVAR CL command. Spooled file attributes, such as the fully-qualified job name and spooled file name and number, are stored in the data queue when the spooled file is created. Such a data queue is useful when spooled files are created under a job that has transferred to another user profile. This function is documented in APAR SA87008 for V5R1 and *Printer Device Programming Version 5,* SC41-5713. The function is available for releases back to V4R2 via a PTF.
- The Retrieve Writer Information API (QSPRWTRI) now supports remote print writers in addition to "normal" print writers.
- The performance of the Retrieve Spool File List APIs QUSLSPL and QGYOLSPL has been improved.
- New support has been added to allow information about iSeries printers and NetServer print shares to be published into Lightweight Directory Access Protocol (LDAP) directories. This allows the user to write applications that query the LDAP directory for iSeries printer information such as a printer's location or capabilities. It also allows the user to configure iSeries printers directly on their Windows 2000 desktop by using the Add Printer wizard, and specifying that the information necessary to configure the printer be obtained from the Windows 2000 Active Directory.

Chapter 4. Using the Infoprint Server for iSeries PDF transform

Infoprint Server for iSeries (5722-IP1) enables the production of Adobe Portable Document Format (PDF) files from any iSeries output at V5R1. PDF files are probably the most common document type on the Internet and company intranets. Even home PC users generally have access to the free Adobe Acrobat Reader software required for viewing. Add to this the ability to read PDF documents on Apple Mac and UNIX systems, and the business case for producing PDF files from your iSeries server becomes very strong.

This chapter explains the following topics:

- Why use the Infoprint Server PDF transform
- How Infoprint Server PDF transform works
- Setting up your PDF virtual printer
- How to use the PDF transform
- Creating a PDF file for storing on an output queue
- Preparing a single PDF file for e-mail
- Creating multiple PDF files
- Additional considerations
- Comparing AFP and PDF files
- Viewing an AFP indexed file
- Viewing and indexing PDF Files with Adobe Acrobat

To use the PDF transform on the iSeries, you require only Infoprint Server for iSeries (5722-IP1). This product has a minimum OS/400 level of Version 5 Release 1. It is sold with a One-Time Charge (OTC), priced by system processor size (one of six software groups), and requires only one licence per system.

Infoprint Server is one of the products delivered as iSeries Keyed Media. This means that the product is available on a 70-day trial basis and is shipped with every copy of the OS/400 V5R1 media. After 70 days, a Software License Key is required to continue to use the product. If this is not ordered, received, and installed within the 70 days, further use of the product is disabled.

4.1 Why use the Infoprint Server PDF transform

As business applications are transformed to e-business implementations, the traditional print and distribute process needs to give way to electronic distribution (with printing optionally taking place at the point of use). “E-output” is a good descriptive term that for this transformation. Infoprint Server for iSeries provides two formats for the electronic distribution of output:

- PDF
- Portable AFP

Portable AFP is enabled with the Create AFP Data (CRTAFPDTA) and is discussed in Chapter 5, “Infoprint Server for iSeries: CRTAFPDTA” on page 149.

This section addresses using PDFs. When you use PDFs and integrated e-mail functions, the following kinds of “e-output” applications can be implemented:

- **Inbound electronic access**: Monthly sales reports are normally printed, burst, and mailed in hardcopy format to the sales regions. The desired transformation is to continue to produce the sales report in the iSeries output queue but in lieu of printing, convert it to PDF and place in the Integrated File System (IFS) for Web or client access.

- **Outbound electronic distribution**: Consider the same sales reporting application. The current process produces one print file that contains reports for multiple sales regions. The desired transformation is to electronically segment the reports for the individual regions and e-mail them as PDF files to the each region's sales manager.
- **Outbound distribution with segmentation**: Order entry has been Web-enabled. Producing hard copy order confirmations defeats the purpose of taking the order process online. The desired transformation is to produce order confirmations as PDF documents and e-mail them automatically to each customer.
- **PDF printing**: Any standard iSeries print file can be transformed to PDF and re-queue to an iSeries output queue. This would enable printing on PDF-capable printers.

The integrated PDF and e-mail subsystem provided with Infoprint Server has a number of characteristics that are important in implementing these kinds of applications:

- **Input spooled file type and complexity**: PDF transform services provide for the transformation of any standard iSeries (AS/400) output format into PDF. This includes standard SNA Character Set (SCS or line mode), extended SCS (for example, OV/400 formats), AFP, mixed mode AFP, IPDS, and line data spooled files. The one format that is not handled (and is not strictly standard) is USERASCII. The supported list comprises all levels of document complexity up to and including AFP.
- **PDF fidelity**: Several emulation modes and the manner is which the input data stream is transformed work to ensure accurate fidelity in the output PDF.
- **Performance options**: Options, such as whether to embedded fonts within the PDF, can be used to balance system and network performance. Embedded fonts ensure accurate fidelity but these fonts can be dropped if the receiving browser or client PC has equivalent fonts available.
- **PDF format**: The PDF produced by Infoprint Server is text-based, fully navigable, and high-performance. This means that the output data encoded in EBCDIC is converted to ASCII but remains data, not a bitmap.
- **Distribution options**: The PDF server is designed to provide three destination options. Once the PDF file is generated, it can be:
  - Passed to the integrated e-mail function
  - Written to the Integrated File System
  - Written back to an output queue

  This enables “outbound” electronic distribution via e-mail, “inbound” electronic distribution via browser or client access to the IFS, or direct printing of PDF to PDF-capable printers.
- **Integrated e-mail**: One of the output options is e-mail. This is an integrated function that occurs automatically based on the PDF subsystem setup.
- **Segmentation**: Segmentation is the capability to create multiple PDF files out of a single output file. This “electronic burst and bind” is the function we referenced earlier in the sales report scenario. With V5R1, there are several

new techniques, including changes to Data Description Specifications (DDS) group keywords by which you can insert triggers into a spooled file that signals to the PDF server that you want multiple PDF files created. The multiple output PDF files can be passed to e-mail, written to IFS, or written back to output queues.

- **Customization**: A user exit built into the PDF processing provides for making file placement or e-mail decisions "on-the-fly". For example, the user exit can take some element of data (for example, customer number) and do a lookup in the database for e-mail ID.

4.2 How Infoprint Server PDF transform works

PDF support is integrated into the iSeries output architecture by using a virtual printer technique. This section provides an introduction to the different components that comprise the PDF subsystem and how your PDF applications flow through them.

Several various components, including the printer file, DDS, PSF configuration object, and customer exit programs, control exactly how the PDF transformation process flows. These components determine whether single or multiple PDF files are created, where the PDF file or files are routed after transformation (directly to e-mail, to IFS, or back to an output queue), and any customization (e-mail content, e-mail ID, IFS file naming, error tracking) that is required.

A PSF configuration object linked to an IPDS device description is used to specify the PDF transform. Any spooled file that is placed in the output queue for this device is processed to PDF format.

Integrating the PDF support as a virtual printer enables OS/400 (specifically, PSF/400) to pre-process different types of spooled file formats supported on the iSeries to a common format – IPDS – prior to passing to the PDF transform subsystem. The PDF transform subsystem, therefore, is an IPDS to PDF transform.

Note

PSF/400 is an integral part of this process, but a license for PSF/400 is not required. A license for PSF/400 is only required if PSF/400 is driving and managing real IPDS printers.

Common document formats are gaining importance in the IT world. The IBM AFP architecture is one of the important portable document formats supported on different operating systems. You can use the PDF transform to take advantage of both architectures.

Let’s take a closer look at how the PDF and e-mail processes work as shown in Figure 66.

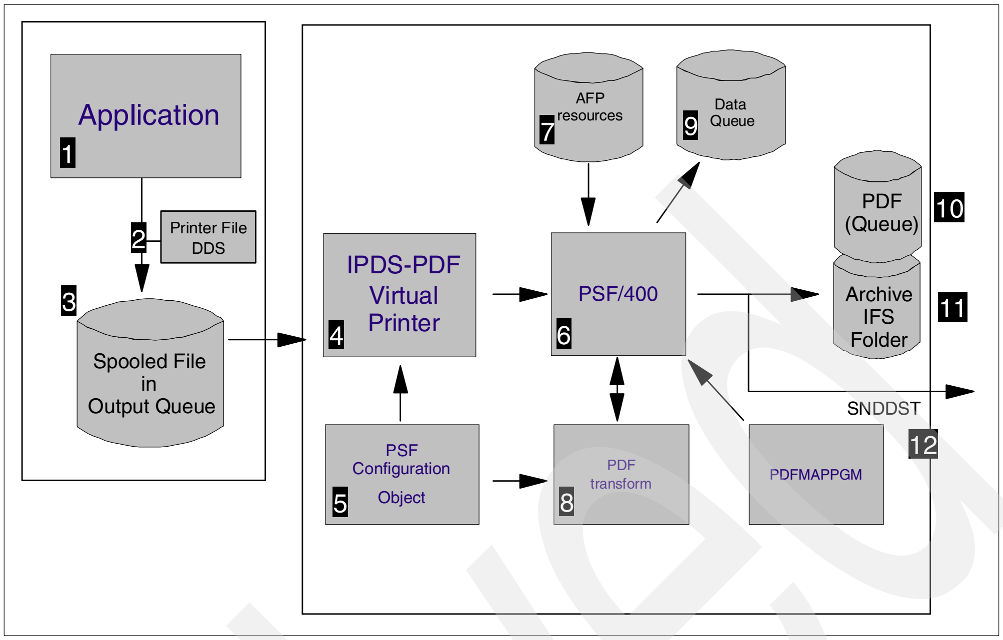


Figure 66. PDF transform flow

The numbers of the flow in Figure 66 correspond to the numbers in the following explanation:

1. An application program creates output, normally using either program-described (internal to the program) or externally-described (DDS) output specifications. The application program could also produce simple output and rely on external formatting objects (AFP formatting resources such as page definitions and form definitions) to complete the final output data stream.
2. Every iSeries program that creates output has an associated printer file. The printer file determines many aspects of what is actually written to the output queue, including data stream type (AFPDS, IPDS, SCS, line data, AFP mixed mode), page characteristics, and even detailed page formatting characteristics (if page/form definitions are referenced).

   Specific printer file parameters or DDS keywords may be used to directly specify the target e-mail ID for a spooled file using the User Defined Data (USRDFNDTA) parameter of the printer file.

   Optionally, the printer file can include DDS specifications. Changes to the DDS keywords Start Page Group (STRPAGGRP) and End Page Group (ENDPAGGRP) with V5R1 can be used to specify the breaks or “triggers” for segmenting the pages in a spooled file. This parameters can also be used to provide indexing and addressing for e-mail. A typical example is to group several pages related to an invoice for a unique customer. A single PDF for a group is sent to a dedicated e-mail address. Chapter 14, “End-to-end example” on page 283, provides an example of how to use most of the Infoprint Server components.
3. All iSeries spooled file are written to an output queue, and all spooled file types except USERASCII are supported by the Infoprint Server PDF transform. These iSeries spooled files types include:

- IPDS
- AFP
- Line data
- AFP mixed mode
- SCS, including DBCS and OV/400 extensions to SCS

There are some printer-specific applications that use embedded printer commands (escape sequences) in the supported spooled file types. These embedded commands are not supported with the PDF transform.

The transform process is invoked as soon as a supported spooled file appears as “Ready” in the output queue.

4. Because the PDF transform subsystem works as an virtual IPDS printer, a printer device description needs to be created for each different PDF process. A specific loopback address is used in place of a standard TCP/IP address.

   A PSF configuration object is linked to the device description using the user defined object parameter. It influences the exact appearance of the resulting PDF file(s). How to create and set up a virtual printer is described in 4.3, “Setting up your PDF virtual printer” on page 108.

5. With V5R1, the PSF configuration object has been expanded to include a series of additional parameters that are used to define and control how the PDF transform operates. Those parameters include:

   - Destination of the generated PDF file to- e-mail, IFS, or output queue
   - PDF device emulation type
   - PDF paper size drawer 1 and drawer 2
   - Single or multiple PDF files
   - PDF data queue
   - Sender of e-mail
   - PDF user exit program
   - Generate PDF output

6. Print Services Facility 400 (PSF/400) is the overall manager of the PDF and e-mail process. Since the PDF transform functions as a virtual printer, the same interfaces that PSF/400 manages in printing to a real IPDS printer (for example, PSF configuration object, printer file, AFP resource objects) are used. Management of the actual PDF transform process is an added task. Licensing PSF/400 is not required when PSF/400 is used in this role.

7. The integrated AFP architecture on iSeries uses a series of resource objects external to the print data stream to produce the final output. Again, PSF/400 manages the retrieval and processing of the AFP resources to compose the IPDS data stream. This IPDS data stream will pass from PSF/400 to the PDF transform with all required AFP resources as:

   - Coded fonts
   - Font character sets
   - Code pages
   - Page segments
   - Overlays
   - Form definitions
   - Page definitions

8. This is the actual PDF transform component, converting input IPDS to PDF output. Transform services include EBCDIC to ASCII translation, AFP to Adobe Type 1 dynamic font transformation, image conversion, and

single/multiple PDF file creation. The PDF data are returned to PSF/400 after the transform process is completed.

9. This data queue is where the PDF and e-mail process will log completion data.
10. Writing the resulting PDF file of files back to an iSeries output queue is one of three options with the PDF transform. This would enable spooled files to be sent directly to PDF-capable printers.
11. Writing the resulting PDF file or files to an Integrated File System directory is another option. Using *STMF as the PDFGEN parameter in the PSF configuration object controls this action.
12. E-mail passing the resulting PDF file or files to the e-mail process is the third option. Using *MAIL as the PDFGEN parameter in the PSF configuration object triggers this result. The Send Distribution (SNDDST) e-mail server function is automatically invoked for the physical e-mail operation

    An optional feature of the e-mail process is the insertion of e-mail exit program (PDFMAPPGM). This program is used to lookup a valid e-mail ID based on data in the USRDTA field or data within the Start Page Group keyword in DDS.

A PDF may be used to distribute information in a common format. It is viewable with the Adobe Acrobat Reader. Or you can use Adobe Acrobat to add information like annotations or index to your PDF files. The following spooled file elements may be converted:

- Text
- Graphics
- Images
- Barcodes (see “Barcode displayed in a PDF file” on page 144)
- Color elements and resources

4.3 Setting up your PDF virtual printer

The PDF transform works as a virtual IPDS printer. Print Service Facility/400 PSF/400) is used to invoke the transform process. From a user point of view, the transform looks like an IPDS printer.

Some of the characteristics of the virtual printer are used to create the PDF document. This section show how to create the virtual printer description used by the PDF transform. A minimum of one description is required.

We can talk about different possible processes using the transform. For each possible destination as an output queue, IFS, or e-mail, the original spooled data may be split into different PDF files regarding the group information created with STRPAGGRP ENDPAGGRP keyword in DDS. The PSF configuration options have to be defined to activate the destination.

Figure 67 shows the relationship between the IPDS virtual printer, the PSF configuration object, and the PDF destination. It also represents a typical implementation of multiple PDF functions. Each PDF output queue and printer device description are set up for a different function. In this example, PDF queue 1 is set up to convert input spooled files to PDF and re-queue them to an iSeries output queue. PDF queue 2 is set up to write single or multiple PDF files to IFS. And, PDF queue 3 is set up to produce single or multiple PDF files and automatically e-mail them using the Send Distribution e-mail function. Since different applications may have different PDF and e-mail requirements, you might set up multiple PDF queues for e-mail, each differing in setup parameters or in which e-mail exit program is used.

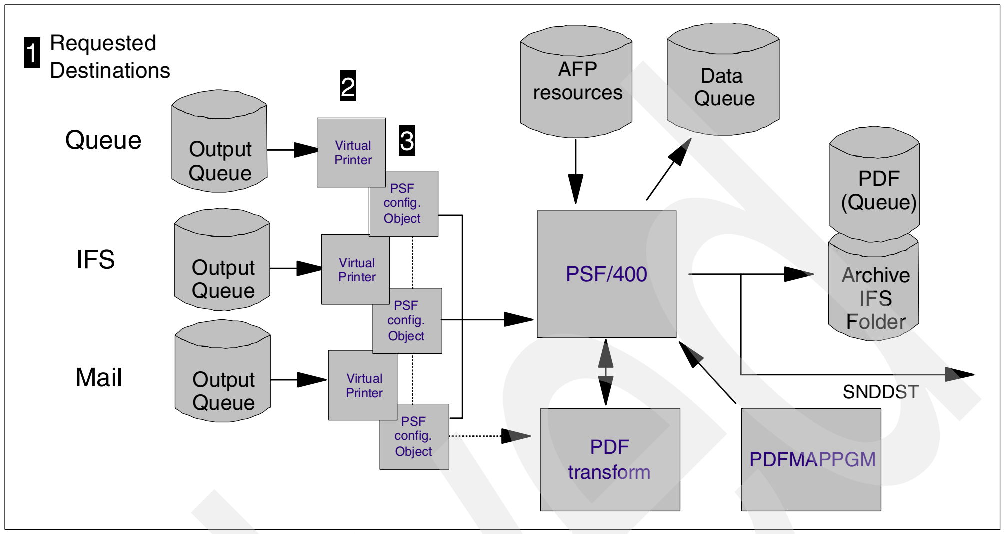


Figure 67. Multiple PDF virtual printer

Note the following explanations for Figure 67:

1. Any supported spooled file may be placed in the appropriate output queue. The different setting defined in the PSF configuration object are used to place the PDF file in the correct destination.
2. These are the virtual printers. A device description is linked to the PSF configuration object. Refer to 4.3.1, "Creating the virtual printer device description" on page 109, for additional information.
3. The PSF configuration object is used to specify IPDS printer characteristics.

4.3.1 Creating the virtual printer device description

The printer device description is created as a virtual printer. You need to create at least one PDF printer device description on your system for each PDF destination desired (IFS, output queue, e-mail). The user-defined object in the printer device description should reference the name of the PDF configuration object that has the detailed PDF creation options.

The transform emulates an IPDS printer and uses the following hardware IPDS configuration settings:

- vpa chk=off
- x offset=0
- y offset=0
- page=whole
- edge-edge=on
- font sub=on
- gcs=char scal
- bar code=auto
- box draw=off

Use the PDFDEVTYPE option in the PSF configuration object to further specify IPDS printer characteristics. You can find additional information in 10.2.4, “PDF transform options” on page 232, about the PSF configuration object related to IPDS/AFP to PDF transform.

Figure 68 shows the first display of the Create Device Description (CRTDEVPRT) command.

```
                   Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Device description . . . . . . . > IPDS2PDFName
 Device class . . . . . . . . . . > *LAN         *LCL, *RMT, *VRT, *SNPT, *LAN
 Device type  . . . . . . . . . . > *IPDS        3287, 3812, 4019, 4201...
 Device model . . . . . . . . . . > 0             0, 1, 2, 3, 4, 10, 13, 301...
 LAN attachment . . . . . . . . . > *IP        *LEXLINK, *IP, *USRDFN
 Advanced function printing . . . > *YES          *NO, *YES
 Port number  . . . . . . . . . . >  5002         0-65535
 Online at IPL  . . . . . . . . .   *YES           *YES, *NO
 Font:
   Identifier . . . . . . . . . . >  011          3, 5, 11, 12, 13, 18, 19...
   Point size . . . . . . . . . .   *NONE         000.1-999.9, *NONE
 Form feed  . . . . . . . . . . .   *TYPE         *TYPE, *CONT, *CONT2, *CUT...
 Separator drawer . . . . . . . .   *FILE         1-255, *FILE
 Separator program  . . . . . . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
 Printer error message  . . . . .   *INQ          *INQ, *INFO
                                                                       More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 68. Create Device Description

Set the required options for the virtual device description as follows:

- **Device description**: Define the name of the virtual printer.
- **Device class**: Type `*LAN` for the virtual IPDS printer.
- **Device type**: Type `*IPDS`.
- **Device model**: The device is always `0` for an IPDS printer.
- **Advanced function printing**: Specify `AFP(*YES)`.
- **Port number**: Specify a unique port number for each virtual printer. A valid value range is from 0 through 65535. We do not recommend using port 5001, which is normally used for IPDS printers.
- **Font identifier**: Specify a default font to be used with this device.

Press the Page Down key to access to the next options as shown in Figure 69.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Message queue  . . . . . . . . .   *CTLD         Name, *CTLD, *SYSOPR, QSYSOPR
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
 Activation timer . . . . . . . .   170           1-2550, *NOMAX
 Image configuration  . . . . . .   *NONE         *NONE, *IMGA01, *IMGA02...
 Maximum pending requests . . . .   6             1-31
 Print while converting . . . . .   *YES          *NO, *YES
 Print request timer  . . . . . .   *NOMAX        1-3600, *NOMAX
 Form definition  . . . . . . . .   F1C10110      Name
   Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
 Remote location:
   Name or address  . . . . . . .      127.0.0.1



 User-defined options . . . . . .   *NONE         Character value, *NONE
               + for more values
                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 69. Specifying the remote location name

To define this printer device description as a PDF virtual printer, the remote location must contain the TCP/IP loopback address 127.0.0.1. Figure 70 shows the next display of the Create Device Description (CRTDEVPRT) command.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 User-defined object:
   Object . . . . . . . . . . . .   MYPSFCFG     Name, *NONE
     Library  . . . . . . . . . .   JHO           Name, *LIBL, *CURLIB
   Object type  . . . . . . . . .   *PSFCFG      *DTAARA, *DTAQ, *FILE...
 Data transform program . . . . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
 User-defined driver program  . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
 Publishing information:
   Support duplex . . . . . . . .   *UNKNOWN      *UNKNOWN, *SIMPLEX, *DUPLEX
   Support color  . . . . . . . .   *UNKNOWN      *UNKNOWN, *NOCOLOR, *COLOR
   Pages per minute black . . . .   *UNKNOWN      1-32767, *UNKNOWN
   Pages per minute color . . . .   *UNKNOWN      1-32767, *UNKNOWN
   Location . . . . . . . . . . .   *BLANK
 Text 'description' . . . . . . .   *BLANK

                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 70. User defined object

Set the following options:

- **User-defined object**: Specify the name of the PSF configuration object. A unique object can be defined for a printer description.

- **Library**: Specify the Library where the PSF configuration object is stored.
- **Object type**: Indicate that the object specified as a user-defined object is a PSF configuration object.

If you try to generate the virtual device description before the PSF configuration has been created, a message appears that indicates that the object is missing.

The PSF configuration object is used to set the IPDS/AFP to PDF transform parameters. It is not possible to change these parameters with a temporary command as an overwrite. You need to create a printer device description and a PSF configuration object for each different setting of the IPDS/AFP to PDF transform.

4.4 Installing and planning the PDF transform

With the appropriate OS/400 media loaded, type `GO LICPGM` and select option `11` (Install licensed programs). Infoprint Server for iSeries is product ID 5722-IP1. To verify whether your system has the product installed, either use option 10 (Display installed licensed programs) from the GO LICPGM menu, or enter the Display Software Resources (DSPSFWRSC) command and look for a display similar to the example in Figure 71.

```
                    Display Software Resources
                                                          System:   DEMO720A
Resource
   ID      Option  Feature  Description
5716DCT    *BASE    5050    Language Dictionaries for OS/400
5716DCT    *BASE    2924    Language Dictionaries for OS/400
5716DCT    22       5050    UK English Dictionary - Option 22
5722AC2    *BASE    5050    Crypto Access Provider 56-bit for AS/400
5722AC2    *BASE    2924    Crypto Access Provider 56-bit for AS/400
5722AF1    *BASE    5050    IBM AFP Utilities for AS/400
5722AF1    *BASE    2924    IBM AFP Utilities for AS/400
5722CM1    *BASE    5050    Communications Utilities/400
5722CM1    *BASE    2924    Communications Utilities/400
5722DG1    *BASE    5050    IBM HTTP Server for iSeries
5722DG1    *BASE    2924    IBM HTTP Server for iSeries
5722IP1    *BASE    5050    IBM Infoprint Server for AS/400
5722IP1    *BASE    2924    IBM Infoprint Server for AS/400
5722JC1    *BASE    5050    AS/400 Toolbox for Java
                                                                  More...
```

Figure 71. Using DSPSFWRSC to verify Infoprint Server for iSeries installation

Using the F11 key on this display also identifies the product library QIPS.

4.5 Creating a PDF file for storing in the IFS

Before you continue the configuration, you must decide what purpose the PDF file will serve. Section 4.1, "Why use the Infoprint Server PDF transform" on page 103, outlines some of these uses. It is important that you clearly identify the purpose of the PDF file because this determines the steps you need to create it.

Producing a PDF file and storing it in the iSeries IFS means that iSeries output is available for applications and users in a convenient, easily-viewable format.

Users' access methods might include file sharing via iSeries NetServer or through a company intranet and Web browser. Applications can take the PDF file and send it to other systems, such as those for e-mail or archive.

The most likely scenarios are:

- Store a PDF file or files in the iSeries Integrated File System (IFS)
- Place a PDF file back on an OS/400 output queue
- E-mail a single PDF file
- E-mail multiple PDF files

The following sections cover the steps you must take to achieve these scenarios.

4.5.1 Creating directories to store the PDF file

PDF files are of ASCII format, stored on the iSeries as *stream files*. Stream files have no internal structure recognized by the OS/400 operating system (contrast with the files/member/fields structure of the traditional OS/400 library system). Perhaps a more familiar example of stream files is OS/400 documents stored in the document library file system. The Integrated File System is simply a way of accessing these different file systems. Examples include:

- **QDLS:** The Document Library Services file system - traditional OS/400 documents in folders
- **QSYS.LIB:** The "traditional" OS/400 library file system
- **root:** The "root" file system, the one most often associated with use of the IFS

Before we create PDF files, we must first decide which file system we want to use, and then create one or more directories within it. The two possibilities are described in the following sections.

4.5.1.1 Using the root file system

To create a directory called *reports* in the root file system, type the following command on an OS/400 command line:

```
CRTDIR reports
```

You can also use the equivalent MKDIR command. The **reports** directory is created within your current directory. The latter can be determined by using the DSPCURDIR command. For access to the many useful interactive commands necessary for IFS navigation, type:

```
GO FILESYS
```

The directory in which the PDF file will be stored is identified in the PDFDIR parameter in the PSF configuration object (see 4.5.2, "Creating a PSF configuration object" on page 116). The syntax is:

```
/directory-name
```

But, the actual resulting location is:

```
/directory-name/job-name/job-user
```

Consider this example:

```
/reports/QPADEV00B/SIMON
```

The actual PDF file name is then stored in the last named subdirectory. Its file name is in the format:

```
job-number_spooled-file-number_spooled-file-name_date_sequence-number.
pdf
```

This format is made up of the following elements:

job-number The job number of the original spooled file.

spooled-file-number The file number of the original spooled file.

spooled-file-name The original spooled file name.

date The two digit month appended to the two digit day appended to the four digit year when the PDF transform completed. (Note the latter – it is *not* the date that the original spooled file was created.)

sequence-number A six-character sequence number. It is set to 000001 if PDFMULT is *NO. If PDFMULT is *YES, the sequence number is increased to uniquely identify each PDF file generated for the job.

Figure 72 shows an example of the naming convention that results from creating a PDF file.

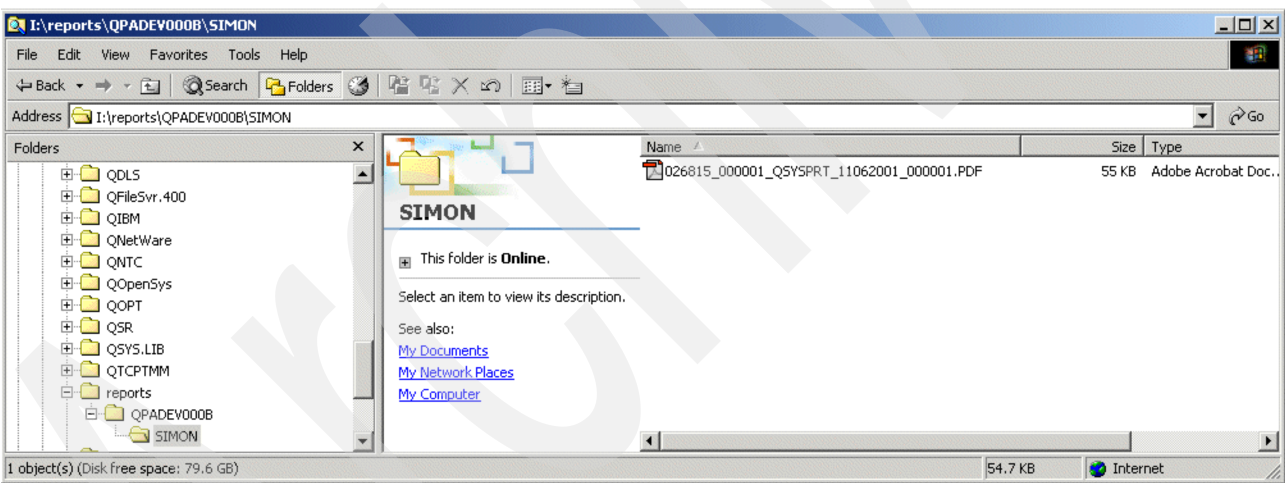


Figure 72. Naming convention for a PDF file created using Infoprint Server for iSeries

In this example, the following filename was created together with the “PDF” suffix:

- **job-number**: 026815
- **spooled-file-number**: 000001
- **spooled-file-name**: QSYSPRT
- **date**: 11062001
- **sequence-number**: 000001

You can also use the WRKLNK command to verify the above results using a 5250 interface (Figure 73). Navigate to the **reports** directory, and then use option `5` to display the successive subdirectories until you reach the PDF file. Note its file type of STMF (stream file).

```
                           Work with Object Links

Directory  . . . . :   /reports/QPADEV000B/SIMON

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type     Attribute    Text
      026815_000001_QSYS >   STMF









                                                                     Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field           F23=More options
```

Figure 73. Locating the PDF file using the WRKLNK command

4.5.1.2 Using the QDLS file system

To create a directory called *emails* in the QDLS file system, type the following command on an OS/400 command line:

```
CRTFLR emails
```

You can use the WRKFLR command to confirm this.

The principal reason for using the QDLS is for e-mail purposes at a later time; the SNDDST process expects to find PDF output files stored in the shared folders directory (QDLS/directory-name). However if you simply want to create and e-mail PDF files in one operation, it is *not* necessary to use a QDLS directory.

You must also make the directory accessible to other users (specifically the print writer). Otherwise the PDF file will not be written into this directory. The reason for this is that QDLS folders are normally created with *PUBLIC *EXCLUDE authority. If the PDF transform cannot write to the folder, the process will fail with reason code 21 (Security violation). Use the EDTDLOAUT command to change the authority; for example, set Public authority to *USE.

When the PDF transform runs, the resulting PDF file is actually stored in the following location:

/QDLS/directory-name/*job-name*/*job-number*[2]/job-user-name/
sequence-number/spooled-file-name/date

Notes

- The *job-name* is truncated to eight characters.
- The last two characters of the *job-name* are added to the beginning of the *job-number*.

The PDF file name is much simpler than before, consisting only of a six-character sequence number. It is set to 000001 if PDFMULT is *NO. If PDFMULT is *YES, the sequence number is increased to uniquely identify each PDF file generated for the job.

Consider this example:

```
/QDLS/emails/QPADEV00/0C028555/SIMON/
000003/QSYSPRT/11092001/000001.pdf
```

The resulting directory structure and file name are shown in Figure 74.

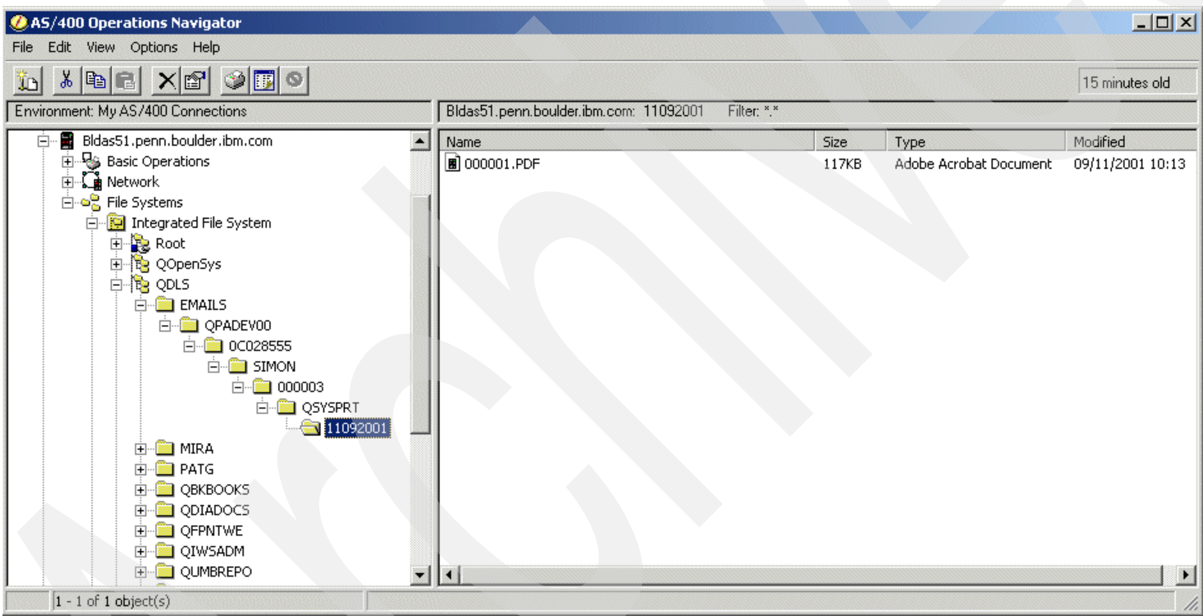


Figure 74. PDF file stored in QDLS directory structure

The naming conventions for the different file systems may seem complex. However, if you are creating and storing PDF files on a daily basis, for example, you have a ready-made filing system for storing the files. The /root file system stores most of the variable information in the PDF file name, the QDLS file system stores this information in the form of successive subdirectories. These two differences may influence which file system you use.

4.5.2 Creating a PSF configuration object

We suggest that you create a PSF configuration object for each specific task you want to perform. For example, to convert iSeries output to PDF files stored in the IFS, first create a PSF configuration object called PDF2IFS. You may also have another PSF configuration object called PDF2EMAIL, for example.

Chapter 10, “PSF configuration object and the iSeries server” on page 221, shows you how to use Operations Navigator to create PSF configuration objects using a Windows GUI view. Alternatively, use the following native OS/400 command in a 5250 session:

```
CRTPSFCFG
```

Then press F4 (Prompt). A display like the example in Figure 75 appears. Complete the following parameters as shown on the display:

- **PSF configuration**: Type a convenient name (for example “PDF2IFS”) and library (for example QGPL).
- **Generate PDF output**: To place the PDF output into the IFS, type `*STMF` (“Stream File”).

```
                     Create PSF Configuration (CRTPSFCFG)

 Type choices, press Enter.

 PSF configuration  . . . . . . . > PDF2IFS       Name
   Library  . . . . . . . . . . .    QGPL         Name, *CURLIB
 User resource library list . . .   *JOBLIBL      *JOBLIBL, *CURLIB, *NONE
 Device resource library list . .   *DFT          Name, *DFT
                + for more values
 IPDS pass through  . . . . . . .   *NO           *NO, *YES
 Activate release timer . . . . .   *NORDYF       *NORDYF, *IMMED...
 Release timer  . . . . . . . . .   *NOMAX        1-1440, *NOMAX, *SEC15...
 Restart timer  . . . . . . . . .   *IMMED        1-1440, *IMMED
 APPC and TCP/IP retry count  . .   15            1-99, *NOMAX
 Delay between APPC retries . . .   90            0-999
 Acknowledgment frequency . . . .   100           1-32767
 Printer response timer . . . . .   *NOMAX        5-3600, *NOMAX
 Generate PDF output  . . . . . .   *STMF         *NONE, *SPLF, *STMF, *MAIL

                                                                       Bottom
```

Figure 75. Create PSF Configuration: Setting PSF configuration and Generate PDF output

Press Enter and change the new parameters as explained here and shown in Figure 76:

- **PDF device emulation type**: Set this to *IP40300 for most cases.
- **PDF paper size drawer 1**: Set this as required (usually *LETTER or *A4). This defines the paper size for drawer 1 that will be used in creating the PDF file.

```
                  Create PSF Configuration (CRTPSFCFG)

 Type choices, press Enter.

 PSF configuration  . . . . . . . > PDF2IFS       Name
   Library  . . . . . . . . . . . >   QGPL        Name, *CURLIB
 User resource library list . . .   *JOBLIBL      *JOBLIBL, *CURLIB, *NONE
 Device resource library list . .   *DFT          Name, *DFT
               + for more values
 IPDS pass through  . . . . . . .   *NO           *NO, *YES
 Activate release timer . . . . .   *NORDYF       *NORDYF, *IMMED...
 Release timer  . . . . . . . . .   *NOMAX        1-1440, *NOMAX, *SEC15...
 Restart timer  . . . . . . . . .   *IMMED        1-1440, *IMMED
 APPC and TCP/IP retry count  . .   15            1-99, *NOMAX
 Delay between APPC retries . . .   90            0-999
 Acknowledgment frequency . . . .   100           1-32767
 Printer response timer . . . . .   *NOMAX        5-3600, *NOMAX
 Generate PDF output  . . . . . . > *STMF         *NONE, *SPLF, *STMF, *MAIL
 PDF device emulation type  . . .   *IP40300      *IP40240, *IP40300, *P4028...
 PDF paper size drawer 1  . . . .   *LETTER       *LETTER, *LEGAL...
                                                                      More...
```

Figure 76. Create PSF Configuration: Specifying device emulation type and paper size drawer 1

Now press the Page Down key and set the remaining parameters as explained here and shown in Figure 77:

- **PDF paper size drawer 2**: This is used only if your original spooled file referenced a page picked from drawer 2. You usually set it to match the value for drawer 1.
- **PDF directory**: Enter a directory name here. The directory must already exist. See 4.5.1, "Creating directories to store the PDF file" on page 113.

```
                  Create PSF Configuration (CRTPSFCFG)

 Type choices, press Enter.

 PDF paper size drawer 2  . . . .   *LETTER       *LETTER, *LEGAL...
 Multiple PDF files . . . . . . .   *NO           *NO, *YES
 PDF data queue . . . . . . . . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *CURLIB
 PDF directory  . . . . . . . . . > /REPORTS



 Text 'description' . . . . . . . > 'PSFCFG to create & store PDFs in "REPORTS"'








                                                                       Bottom
```

Figure 77. Create PSF Configuration: Specifying PDF paper size drawer 2 and PDF directory

4.6 Creating a PDF file for placing on an output queue

PDF files are normally printed from a Windows application, via an ASCII print driver, most commonly a PCL or PostScript printer driver. In other words, the PDF file is transformed into either PCL or PostScript by the print driver and sent to the printer. This is actually overhead on the PC processing. It is possible to send PDF files directly to some printers – those that are “PDF-capable”. Typically the latter printers have a hard disk to receive (spool) the PDF file, from where the PDF is rasterized into the format necessary for printing.

Printers from IBM that are PDF-capable include the Infoprint 21 and Infoprint 70. If you have a PDF-capable printer and you access it from the iSeries, you can send the PDF files to the printer, as well as store them in a file system. To do this, change the PSF configuration object parameters as follows (or, more likely, use a dedicated PSF configuration object for this purpose):

```
CRTPSFCFG PSFCFG(QGPL/PDF)
PDFGEN(*SPLF)
PDFDEVTYPE(*IP40300)
PDFMULT(*NO)
PDFSENDER(*NONE)
PDFOUTQ(IP70LPR/QUSRSYS)
TEXT('PSF configuration to print pdf')
```

It is important that the target printer is PDF-capable. Otherwise, you may receive garbage output. Note that most PDF-enabled printers require specific hardware features to print PDF files. If a PDF is sent to an Infoprint 21 printer without a hard drive, it will fail. With the 1000 series and the ImageQuick feature, we used the queue name TEXT, which proved to be successful.

You need to create the remote queue specified in the PSF configuration object. The LR/LPD used with the Remote Output Queue is written to the disk on the hard drive and then processed. The data stream is opened. It is determined that it is a PDF file and processed accordingly.

Remote output queue (LPR/LPD)

A remote output queue is a way to automate the LPR command. The configuration of remote output queues is described in a number of publications, including:

- *OS/400 Printer Device Programming V4R2*, SC41-5713
- *AS/400 Printing IV*, GG24-4389
- *AS/400 Printing V*, SG24-2160

At a minimum, you need the IP address of the remote system or server (the printer itself often performs this role) and the queue name that already exists on that remote system. If the remote system is a printer, use the built-in internal print queue name that most printer LPDs have.

A sample table of common print queue names is listed in *AS/400 Printing V*, SG24-2160. Or you can find a more extensive and up-to-date list on the SupportLine Knowledge Base Web site as explained at the beginning of Appendix A, “Configuring LAN-attached printers” on page 317.

4.7 Preparing a single PDF file for e-mail

The setup tasks for generating PDF files as e-mail attachments are similar to the tasks for saving the files in the IFS or creating new spooled files in an output queue. But this time there is the added step of determining how to associate an e-mail address to a given spooled file or segment.

There are a number of different approaches you may take depending on the type of spooled file you are dealing with, what you want to do with it, and whether you have access to modify application objects.

The first important consideration is how you are sending your PDF file:

- Sending the entire file as one PDF to one destination.
- Segmenting the original spooled file into smaller PDF files to be sent to multiple different destinations. An example of doing this is the case where you have a spooled file that contains invoices or statements for multiple different customers, and each customer is to receive only the pages that apply.

The other consideration is the degree to which you have access to make changes to the application, such as:

- Changing or overriding the printer file
- Modifying the DDS
- Generating the spooled file as *AFPDS
- Generating applications using the AFP Toolbox

Figure 78 illustrates the various methods of specifying the e-mail address for the PDF file or files. This section describes the USRDFNDTA MAILTAG (A) and USRDTA (B) options for sending single files. The choice between the two options depends on whether you can change or override the printer file.

Refer to 4.8, “Creating multiple PDF files” on page 127, for details on how to specify the e-mail address when you are breaking one large spooled file into multiple separate PDF files to e-mail to different destinations. The use of the AFP Toolbox (not shown on the decision tree) is also described as an alternative for setting addresses for a multiple PDF scenario.

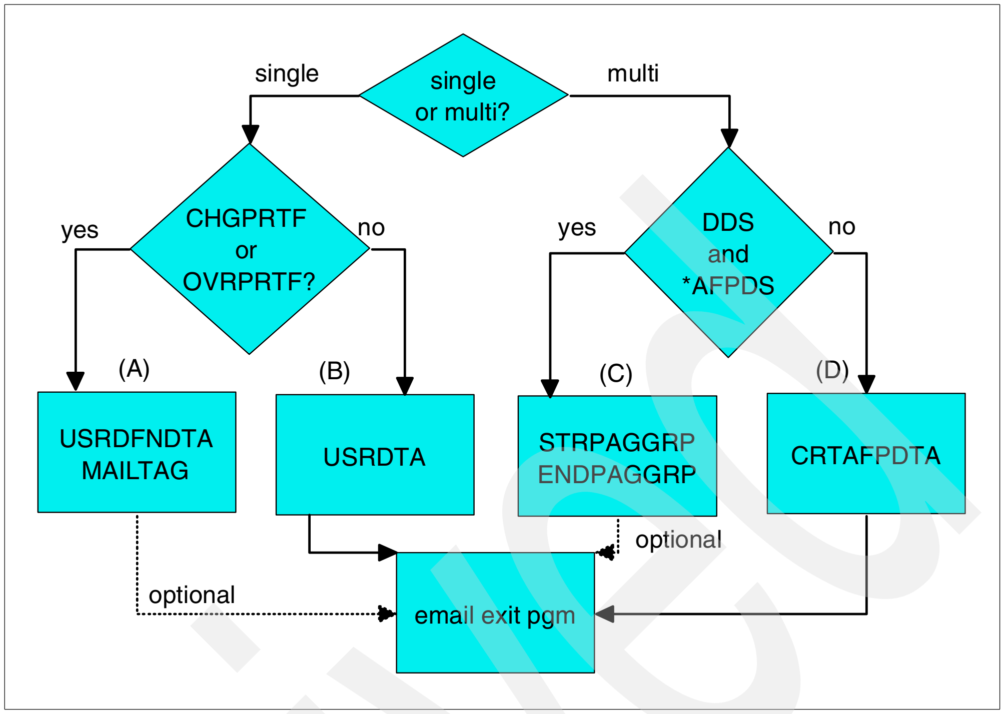


Figure 78. Options for specifying the e-mail address

4.7.1 Using USRDFNDTA to specify an e-mail address

There are new options available with V5R1 within the USRDFNDTA parameter in the Create, Change, or Override Printer File commands. If you have access to change your printer file, or if you can issue an Override Printer File command prior to generating the spooled file, use this parameter to address your e-mail. In doing so, you may avoid the requirement of an e-mail exit program.

This corresponds to branch (A) in Figure 78.

4.7.1.1 PSFCFG for single e-mail, no exit program

Figure 79 shows an example of the Create PSF Configuration display that is used to send your spooled files as single PDF e-mail attachments, if you are not using an e-mail program. (Only those parameters that pertains to PDF creation are shown.)

```
                    Create PSF Configuration (CRTPSFCFG)

Type choices, press Enter.

PSF configuration . . . . . . . > EMAIL1        Name
  Library  . . . . . . . . . . .     QGPL        Name, *CURLIB

Generate PDF output  . . . . . . > *MAIL         *NONE, *SPLF, *STMF, *MAIL
PDF device emulation type  . . . > *IP40300      *IP40240, *IP40300, *P4028...
PDF paper size drawer 1  . . . .   *LETTER       *LETTER, *LEGAL...
PDF paper size drawer 2  . . . .   *LETTER       *LETTER, *LEGAL...
Multiple PDF files . . . . . . .   *NO           *NO, *YES
PDF data queue . . . . . . . . . > PDFDTAQ       Name, *NONE
  Library  . . . . . . . . . . . >   QGPL        Name, *CURLIB
Sender of electronic mail  . . . > EMAILMGR      Name, *SPLFOWN, QSPLJOB
PDF user program . . . . . . . .   *NONE         Name, *NONE
  Library  . . . . . . . . . . .                 Name, *CURLIB
Text 'description' . . . . . . . > 'Email PDF files as single file'
```

Figure 79. PSF configuration object for single e-mail, no user exit program

On this display, specify the following parameters:

- **Generate PDF output** (PDFGEN): Specify *MAIL to tell Infoprint Server for iSeries to generate the PDF file as an e-mail attachment.
- **PDF device emulation type** (PDFDEVTYPE): IBM recommends that you set this to *IP40300, which corresponds to the functions of an IBM Infoprint 40 set to 300 dots per inch.
- **PDF paper size drawer 1 and 2** (PDFPPRDWR1 and PDFPPRDWR2): Set this to the paper size used by the application.
- **Multiple PDF files** (PDFMULT): Set this to *NO to send each file as a single e-mail.
- **PDF data queue** (PDFDTAQ): Specifies the name and library of the data queue where PDF logs the transformation completion notifications. If you specify this parameter, the data queue must exist and have a minimum length of 752 characters. See E.2, “PDF data queue structure” on page 346, for additional information.
- **Sender of electronic mail** (PDFSENDER): Specifies the user from whom the PDF output file is e-mailed. The sender must be a valid user profile and be enrolled in the system distribution directory. Any non-delivery messages caused by invalid e-mail addresses are sent to this user. For additional information on non-delivery messages, see Chapter 14, “End-to-end example” on page 283.
- **PDF user program** (PDFMAPPGM): Set this to *NONE to indicate that an e-mail exit program is not necessary. Valid e-mail addresses *must* be supplied by the USRDFNDTA parameter.

4.7.1.2 Printer DEVD for single e-mail, no exit program

Create a device description that uses this PSF Configuration object by specifying:

```
CRTDEVPRT  DEVD(EMAIL1)
            DEVCLS(*LAN)
            TYPE(*IPDS)
            MODEL(0)
```

```
LANATTACH(*IP)
AFP(*YES)
PORT(5003)
FONT(416 12)
RMTLOCNAME('127.0.0.1')
USRDFNOBJ(QGPL/EMAIL1 *PSFCFG)
TEXT('Use for PDF email, single files, no exit pgm')
```

4.7.1.3 Using USRDFNDTA

To use this combination of the PSF configuration object and device description to send your e-mail, you must be able to perform one of the following actions to set the address:

- Create a new printer file using CRTPRTF, or change an existing printer file using CHGPRTF.
- Modify the CL used to run program to insert an Override Printer File (OVRRPRTF) command. This must be done prior to running the application program that generates the spooled file.
- Interactively override the printer file prior to running the program.

Here is an example of using the Change Printer File command with the User Defined Data (USRDFNDTA) parameter:

```
CHGPRTF mylib/myprtf OUTQ(EMAIL1) USRDFNDTA('MAILTAG(jacques_h@organic.ch)
MAILSENDER(MSHNIER)')
```

The same parameters may be specified on the CRTPRTF or OVRPRTF commands.

In this example, the output is directed to the output queue, EMAIL1, where it is picked up by Infoprint Server and sent as a PDF attachment in an e-mail to jacques_h@organic.ch.

The optional MAILSENDER value, if specified, overrides the value set in the PSF Configuration object for the PDFSENDER parameter. In this example, the sender is MSHNIER instead of EMAILMGR. As with the PDFSENDER, MAILSENDER must be a valid user profile and must have a directory entry in the system distribution directory.

Note

With V5R1, the USRDFNDTA parameter is not available in the Change Spooled File Attributes (CHGSPLFA) command. It *must* be used at the printer file level, prior to generating the spooled file.

Figure 80 shows an example of an e-mail received by a Lotus notes user.

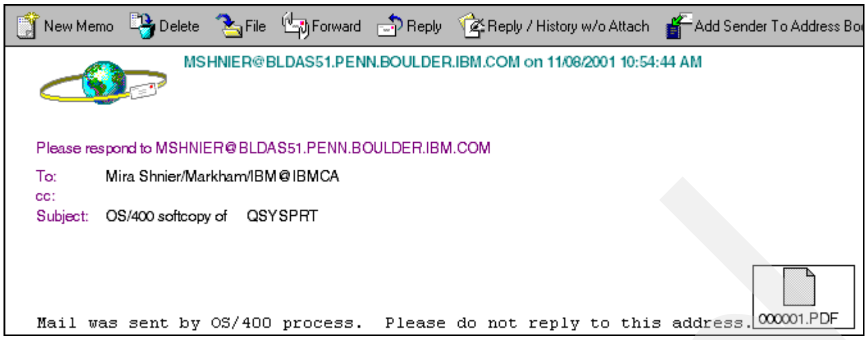


Figure 80. Lotus note with PDF attachment

Note the following features:

- The sender name, in this case MSHNIER, was specified using the USRDFNDTA keyword. BLDAS51.PENN.BOULDER.IBM.COM is the iSeries from which this e-mail was sent.
- The subject line contains the name of the original spooled file, in this case QSYSPRT.
- The mail message was not customized. This is the default message that is generated.
- The PDF attachment is given the name 000001.PDF.
- The original mail was sent to an address in the format user@company. Lotus Notes converts that address into its own format.

Clicking the attachment icon launches Adobe Acrobat. The document, as shown in Figure 81, is displayed. The document used in this case was a simple screen print.

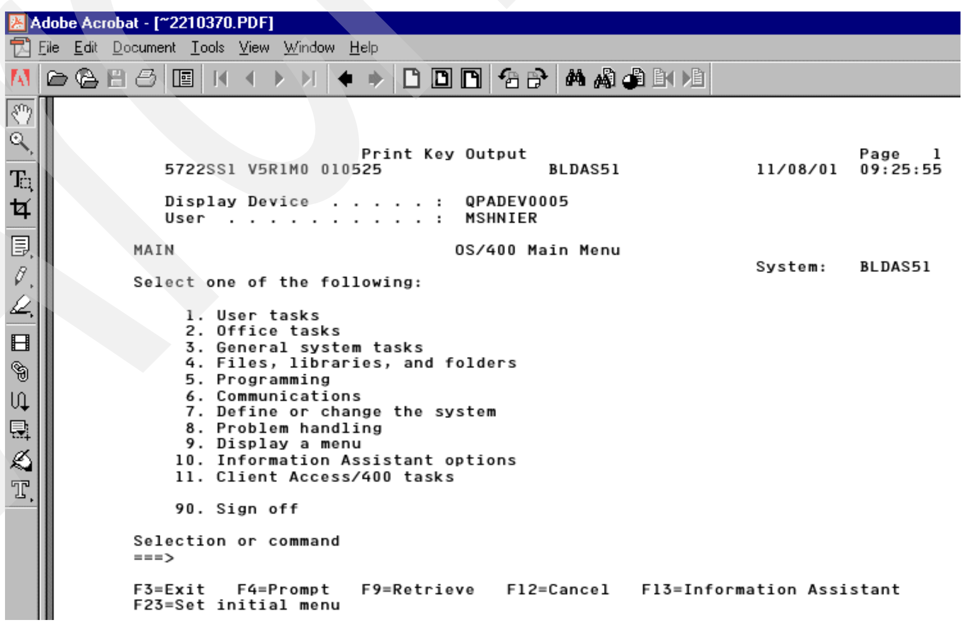


Figure 81. Sample PDF of a sample screen print

4.7.2 Addressing your e-mail with USRDTA and an e-mail exit program

In the previous section, it was indicated that in V5R1, the USRDFNDTA parameter must be specified in the printer file before running the application. It is not available for use in the Change Spooled File Attribute (CHGSPLFA) command.

If you do not have access to make the required application changes prior to creating the spooled file, there is an alternate technique you can use. This involves entering a unique value in the USRDTA parameter of the spooled file. This parameter is available from the CHGSPLFA command, as well as the CRTPRTF, CHGPRTF, or OVRPRTF commands. This method requires using an e-mail exit program to convert the 10-character value used in the USRDTA parameter to a valid e-mail address.

This corresponds to branch (B) in Figure 78.

4.7.2.1 PSFCFG for single e-mail, with exit program

Figure 82 illustrates the parameters used by Create PSF Configuration that are used to send single e-mail files if using an e-mail exit.

```
               Create PSF Configuration (CRTPSFCFG)

 Type choices, press Enter.

 PSF configuration  . . . . . . . > EMAIL1EXIT    Name
   Library  . . . . . . . . . . .     QGPL        Name, *CURLIB

 Generate PDF output  . . . . . . > *MAIL         *NONE, *SPLF, *STMF, *MAIL
 PDF device emulation type  . . . > *IP40300      *IP40240, *IP40300, *P4028...
 PDF paper size drawer 1  . . . .   *LETTER       *LETTER, *LEGAL...
 PDF paper size drawer 2  . . . .   *LETTER       *LETTER, *LEGAL...
 Multiple PDF files . . . . . . .   *NO           *NO, *YES
 PDF data queue . . . . . . . . . > PDFDTAQ       Name, *NONE
   Library  . . . . . . . . . . . >   QGPL        Name, *CURLIB
 Sender of electronic mail  . . . > EMAILMGR      Name, *SPLFOWN, QSPLJOB
 PDF user program . . . . . . . .   EXIT1         Name, *NONE
   Library  . . . . . . . . . . .     QGPL        Name, *CURLIB
 Text 'description' . . . . . . . > 'Email PDF files as single file with exit'
```

Figure 82. PSF configuration object for single e-mail, no user exit program

You need to complete the following parameters:

- **Generate PDF output** (PDFGEN): Specify `*MAIL` to generate the PDF file as an e-mail attachment.
- **PDF device emulation type** (PDFDEVTYPE): IBM recommends setting this to `*IP40300`, which corresponds to the functions of an IBM Infoprint 40 set to 300 dots per inch.
- **PDF paper size drawer 1 and 2** (PDFPPRDWR1 and PDFPPRDWR2): Set this to the paper size used by the application.
- **Multiple PDF files** (PDFMULT): Set this to `*NO` to send each file as a single e-mail.
- **PDF data queue** (PDFDTAQ): Specifies the name and library of the data queue where PDF logs the transformation completion notifications. If you specify this parameter, the data queue must exist and have a minimum length

of 752 characters. See E.2, “PDF data queue structure” on page 346, for additional information.

- **Sender of electronic mail** (PDFSENDER): Specifies the sender from whom the PDF output file is e-mailed. The sender must be a valid user profile and be enrolled in the system distribution directory. Any non-delivery messages caused by invalid e-mail addresses is sent to this user. For additional information on non-delivery messages, see Chapter 14, “End-to-end example” on page 283.
- **PDF user program** (PDFMAPPGM): Specify the name and library of the user e-mail exit program. This program is used to look up a valid e-mail address based on the information in the USRDTA field.

4.7.2.2 Printer DEVD for single e-mail, with exit program

Create a device description that uses this PSF Configuration object by specifying:

```
CRTDEVPRT  DEVD(EMAIL1EXIT)
            DEVCLS(*LAN)
            TYPE(*IPDS)
            MODEL(0)
            LANATTACH(*IP)
            AFP(*YES)
            PORT(5004)
            FONT(416 12)
            RMTLOCNAME('127.0.0.1')
            USRDFNOBJ(QGPL/EMAIL1EXIT *PSFCFG)
            TEXT('Use for PDF email, single files, with exit pgm')
```

4.7.2.3 Changing the value for USRDTA

The User Data field may be modified using the Change Spooled File Attributes (CHGSPLFA) command. You may specify up to 10 characters. A quick way is to do this interactively. Display a list of your spooled files using a command such as WRKSPLF or WRKOUTQ. Then enter option 2 next to the desired spooled file and type the USRDTA information on the command line.

In Figure 83, you see an example of the Work with All Spooled Files (WRKSPLF) display. The User Data for the first spooled file is already set. The process is about to be repeated for the second spooled file.

```
                          Work with All Spooled Files

Type options, press Enter.
  1=Send   2=Change   3=Hold   4=Delete   5=Display   6=Release   7=Messages
  8=Attributes        9=Work with printing status

                                Device or                    Total     Cur
Opt  File        User        Queue       User Data   Sts   Pages    Page  Copy
     EMAIL       MSHNIER     MSHNIER     CUST000426  RDY       8              1
 2   EMAIL       MSHNIER     MSHNIER                 RDY       8              1

                                                                          Bottom
Parameters for options 1, 2, 3 or command
===> usrdta(CUST000136)
F3=Exit   F10=View 4   F11=View 2   F12=Cancel   F22=Printers   F24=More keys
```

Figure 83. Changing the USRDTA value for a spooled file

From within a program, you can override or change the USRDTA field by placing a CALL to the system program QCMDEXC.

4.7.2.4 Exit program to map USRDTA

The name and library of the e-mail exit program are specified in the PSF Configuration object. Each time Infoprint Server process a PDF file that is to be sent as an e-mail, the exit program is called.

Information about the input spooled file is passed to the exit program via the input data structure. The program uses this information to look up or generate a valid e-mail address. In addition, the exit program may also be used to build a message of up to 255 characters. This message appears in the body of the e-mail, along with the PDF attachment. The e-mail address and message are passed back to Infoprint Server via an output data structure.

In this scenario, we use the value from the USRDTA field, which is made available to the program via positions 647 to 656 of the input data structure. This value is used in a lookup operation to an external file to obtain the e-mail address and optionally other information that may be used in the custom message.

A sample of a e-mail exit program is found in E.1, “E-mail exit program” on page 343. Along with the generation of the e-mail address and message, it also performs some additional tasks in the case of a failure of the lookup operation. It may be used as a template for addressing using USRDTA with some minor modifications. Consider the following points:

- Note that the USRDTA field is made available to the exit program in positions 647 to 656 of the Input Data Structure (INPUTDS).
- The existing sample uses the first six characters from the MAILTAG field as key to the lookup operation. Modify the program to use the 10-character USRDTA field.
- Modify the LOOKUP file to match the expected data from the USRDTA field. In particular, change the length of the key field.
- The result of the LOOKUP program returns one or more e-mail addresses, which you pass via the ADDRESS field in Output Data Structure (OUTDS) starting in position 288. In the sample program, a length of 255 was assigned to this field. If necessary, the length may be changed up to 16 MB long.
- Customize the Message and not-found actions to suit your specific application needs.

4.8 Creating multiple PDF files

In previous sections, we describe how to set up the PSF Configuration object to generate single files on the IFS, single spooled files, or single e-mails. Infoprint Server for iSeries enhances these operations by providing the means to take one original spooled file and generate multiple PDF files in the IFS, multiple new PDF spooled files, or multiple files being directed to different e-mail destinations.

You may need to do this is if your application generates one spooled file containing multiple customer statements. You want to segment it on statement boundary to generate multiple PDF files.

4.8.1 PSF configuration object for multiple PDF files

The PSF configuration object used to create multiple files for print, storage, or e-mail is similar to the corresponding PSF Configuration object used for single files for each of the three file types. The main difference is that the parameter Multiple PDF files (PDFMULT) is now set to *YES.

Figure 84 shows an example of the parameters used to create a PSF Configuration object to generate multiple PDF files. In this particular case, the files are to be sent via e-mail, and a user exit program is specified.

```
                    Create PSF Configuration (CRTPSFCFG)

Type choices, press Enter.

PSF configuration . . . . . . . > EMAILMULT     Name
  Library . . . . . . . . . . .     QGPL        Name, *CURLIB

Generate PDF output . . . . . . > *MAIL         *NONE, *SPLF, *STMF, *MAIL
PDF device emulation type . . . > *IP40300      *IP40240, *IP40300, *P4028...
PDF paper size drawer 1 . . . .   *LETTER       *LETTER, *LEGAL...
PDF paper size drawer 2 . . . .   *LETTER       *LETTER, *LEGAL...
Multiple PDF files . . . . . . .  *YES          *NO, *YES
PDF data queue . . . . . . . . . > PDFDTAQ      Name, *NONE
  Library . . . . . . . . . . . >   QGPL        Name, *CURLIB
Sender of electronic mail . . . > EMAILMGR      Name, *SPLFOWN, QSPLJOB
PDF user program . . . . . . . .  EXITMULT     Name, *NONE
  Library . . . . . . . . . . .     QGPL        Name, *CURLIB
Text 'description' . . . . . . . > 'Email PDF files as multiples'
```

Figure 84. PSF configuration object for multiple PDF file

To generate the multiple PDF spooled files or multiple files on the IFS, refer to the PSF Configuration objects described 4.5, “Creating a PDF file for storing in the IFS” on page 112. To generate multiple files, change the value for Multiple PDF Files from *NO to *YES.

4.8.2 Printer file for multiple PDF file

Create a device description that uses the PSF configuration object for creating multiple PDF files by specifying:

```
CRTDEVPRT  DEVD(EMAILMULT)
           DEVCLS(*LAN)
           TYPE(*IPDS)
           MODEL(0)
           LANATTACH(*IP)
           AFP(*YES)
           PORT(5005)
           FONT(416 12)
           RMTLOCNAME('127.0.0.1')
           USRDFNOBJ(QGPL/EMAILMULT *PSFCFG)
           TEXT('Use for PDF email, single files, with exit pgm')
```

4.8.3 Segmenting the spooled file

Whether you are generating multiple PDF files to be re-spooled in an output queue, placed in the IFS, or sent as an e-mail, you need a way to tell the PDF subsystem of Infoprint Manager for iSeries how you want it to break up the original spooled file. This is done by inserting *group tag* records into the spooled file.

There are three ways to generate applications with group tags:

- Using DDS printer file keywords
- Using the Infoprint Server Create AFP Data (CRTAFPDTA) command to generate index information
- Using the AFP Toolbox APIs

In the case of e-mail delivery, you need to provide valid e-mail addresses to the PDF subsystem. The information provided by the group tag record may contain a valid address, in which case, an e-mail exit program may not be necessary. If the tag is more generic, for example a customer number, you must use the e-mail exit program to look up the e-mail address.

This section describes the three methods of inserting the group tag records used by all three delivery methods. It also covers specific information on specifying an e-mail address, with and without the use of an e-mail exit program. Referring back to the address decision tree in Figure 78 on page 121, this section covers the branches labeled C and D.

4.8.3.1 Using DDS printer file for indexing

If you have access to the original application program and DDS, *and* are in a position where you can generate your spooled files as AFPDS, you can use the DDS record-level keywords, STRPAGGRP and ENDPAGGRP, to tell Infoprint Server for iSeries how to break the original file into groups of pages. This corresponds to branch C in Figure 78 on page 121.

One PDF file is generated between every STRPAGGRP and ENDPAGGRP keyword pair. If you choose to specify groups, all of the data must be inside a group. Groups of pages cannot be nested or overlapped, each group must be ended with the ENDPAGGRP keyword before another can begin.

The group name should be unique within a document. The maximum number of characters in the group name is 250.

The group name may be hard coded into the DDS. However, it is more likely that it would be passed as a variable using *program-to-system fields*. Program-to-system fields are defined by entering a “P” in position 38 of the DDS specification for the field. The program would supply the value, which in turn, would be used as a parameter for the STRPAGGRP keyword.

The examples in Figure 85 show how to specify the STRPAGGRP and ENDPAGGRP keywords.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
     A     R RECORD1
     A                                    STRPAGGRP('CUST68839')
     A
     A     R RECORD2
     A                                    STRPAGGRP('simon_h@uk.ibm.com')
     A
     A     R RECORD3                      STRPAGGRP(&GROUP)
     A       GROUP             50A  P

     A     R RECORD4                      ENDPAGGRP
```

Figure 85. Examples of the STRPAGGRP and ENDPAGGRP keyword

In the example, the STRPAGGRP keyword in RECORD1 contains a valid e-mail address. An e-mail exit program is not necessary if this format is used.

RECORD2 starts a group named CUST68839. If this tag were used for an e-mail application, an exit program must be used to map that to a valid e-mail address.

RECORD3 illustrates the use of a program-to-system field. The program would assign a value to the field GROUP, just as it would any other field you would normally print. The difference is that there is a “P” in position 38 of the specifications. This tells DDS not to print the field, but rather, to use it as a variable in one of the other keywords, in this case, STRPAGGRP. The use of an exit program would be determined by whether the value for GROUP was a valid e-mail address or just a key to a lookup file.

RECORD4 uses the ENDPAGGRP keyword to end a group that was previously started with the STRPAGGRP keyword.

In the example of an invoice or statement, consider using the STRPAGGRP keyword immediately after writing the invoice header or address information, and the ENDPAGGRP after writing the totals or summary information.

Using the exit program with the STRPAGGRP keyword

If you are generating your PDF files for the e-mail function, and the value used in the STRPAGGRP keyword is a valid e-mail address, you do not necessarily need an e-mail exit program. You may choose to use one if you want to add a message to the body of the e-mail.

You may find it convenient to keep the maintenance and selection of e-mail addresses separate from the application program that generates the spooled file. For example, an invoice program already has the customer number available, so the modifications to output it with the STRPAGGRP keyword are not extensive. A separate set of programs and files can be used to maintain the e-mail address lists. The e-mail exit program is then used to map the tag used in STRPAGGRP with the valid address.

The e-mail exit program sample in E.1, “E-mail exit program” on page 343, may be used as a template. The value provided by STRPAGGRP is available as the Mail Tag field of the input data structure. Use this (or a substring) to look up the e-mail address.

4.8.3.2 Using CRTAFPDTA indexing

The Create AFP Data (CRTAFPDTA) command may be an alternative to using the DDS keywords STRPAGGRP and ENDPAGGRP to segment the spooled file. These DDS keywords require you to generate spooled files that are in AFPDS format and modify the DDS. There are a number of common scenarios where these conditions may not be met:

- If the spooled file is generated as line data and uses a page definitions for formatting. This is the case for users of Infoprint Designer for iSeries using the Layout tool.
- The spooled files are generated using program-defined printer files. This is common if the application was originally written for the IBM System/36.
- The spooled files are generated by a purchased package and you do not have the ability or access to make the required modifications to the program and DDS source.

If you find that any of the above conditions apply, you need an alternate method to instruct Infoprint Server how to break up your file and destination to whom the segments are to be mailed. The CRTAFPDTA command, which is another component of Infoprint Server for iSeries, may be the solution.

This corresponds to branch D in Figure 78 on page 121.

Requirements for using CRTAFPDTA

The CRTAFPDTA command has two important requirements that must be met in order to use it to generate the index records:

- The input spooled file must be in line data format.
- The information used to generate the index information must be in predictable positions within the printer file.

Line data

If you are using Infoprint Designer for iSeries to create your application, you are generating line data in order to use the page definitions generated by the Layout function.

Many other applications on the iSeries generate SCS data stream. These are often heritage applications originally written for the S/36 or ERP applications that do not provide any formatting capability. Line data is very similar to SCS. If you have the ability to change or override the printer file, it is an easy task to convert it to produce line data instead. You can use either of the following commands:

```
CHGPRTF mylib/myprtfile DEVTYPE(*LINE)
OVRPRTF mylib/myprtfile DEVTYPE(*LINE)
```

Section 2.10.1, “Changing printer file to *LINE data” on page 82, describes a technique to create a separate printer file with the appropriate parameters and place it higher up in the library list. This technique may help you generate line data for use with CRTAFPDTA.

Predictable data positions

The indexing function of CRTAFPDTA needs to know where to look in the spooled file for the information it needs to identify the break points between the different parts of the spooled data. If the file is to be sent as an e-mail, you need to provide meaningful data, such as customer number to use for the lookup operation in the e-mail exit program.

Refer to the Chapter 4, “Using the Infoprint Server for iSeries PDF transform” on page 103, and to Chapter 5, “Infoprint Server for iSeries: CRTAFPDTA” on page 149, for a detailed explanation on how to setup the CRTAFPDTA command to identify the segments of the spooled file and add index records.

Using CRTAFPDTA for PDF creation

The CRTAFPDTA command generates output on the IFS. If the target directory is in the QSYS.LIB subsystem, the data can be accessed as a physical file member.

The PDF generation in Infoprint Server works on spooled files. The data in the physical file created by CRTAFPDTA must be re-spooled using the Print AFP Data (PRTAFPDTA) command before it is converted to PDF.

Chapter 14, “End-to-end example” on page 283, describes a complete scenario, from developing an application using Infoprint Designer and sending multiple PDF files as e-mail attachments. It includes detailed step-by-step instructions for each stage of the process:

- Modifications to the Infoprint Designer project (if necessary)
- A single, one-time modification to the application (if necessary)
- Using CRTAFPDTA to index the data
- Using PRTAFPDTA to re-spool the indexed file
- Viewing the indexed spooled file with the AFP Viewer
- Creating a PSF configuration object and an output queue for PDF conversion
- Creating an exit program to lookup e-mail addresses
- Monitoring for undeliverable mail
- Other maintenance tasks

If you are using CRTAFPDTA to index your spooled file in order to generate multiple PDF files, please refer that the appropriate sections of that chapter.

Using the e-mail exit program with CRTAFPDTA

The exit program in E.1, “E-mail exit program” on page 343, was designed to be used with the index keys generated by CRTAFPDTA. Customize it to suit your individual requirements.

The data generated as the key tag field has an eight-character numeric string appended to ensure that the keys are unique. You need to perform a substring operation to separate out the key, such as the customer number, prior to performing the lookup function.

4.8.3.3 Using the AFP Toolbox

The AFP Toolbox is a package that contains procedure function calls to be included in an RPG, COBOL, or C++ program. It is intended for a highly skilled programmer to build AFPDS applications directly. See 1.1.5, “AFP PrintSuite – AFP Toolbox” on page 11, for more information.

The AFP Toolbox includes procedures for Begin Group and End Group, which work in a similar fashion to the DDS keywords STRPAGGRP and ENDPAGGRP. For specific syntax information, see the *IBM AFP Toolbox for Multiple Operating Systems User's Guide*, S544-5292.

Use of the AFP Toolbox is not described in Figure 78 on page 121. If it is used by the developer, the group procedures allow them to segment the file for generating multiple PDF files if necessary.

e-mail exit considerations for AFP Toolbox

As with the DDS implementation, it may not be necessary to use an e-mail exit program if you use a valid e-mail address when you generate the spooled data. However, you may choose to use an exit program to customize the e-mail message.

4.8.4 Naming convention for multiple files

When multiple PDF files are generated, they are assigned sequential names, depending on the type of file being generated.

4.8.4.1 Multiple files on the IFS

To store the files on the IFS, refer to 4.5, “Creating a PDF file for storing in the IFS” on page 112, for information on the complete directory structure. If the files are generated in the QDLS directory, the files will be named 000001.PDF, 000002.PDF, and so on. If the /root directory is being used, the file name will contain other information, but the last ten characters will be 000001.PDF and so on.

If you are implementing an application based on storing files in the IFS, you can use information that is logged in the PDF data queue to help match these sequential names with the information from the index tag that is used to segment the file, such as invoice number.

4.8.4.2 Multiple files for e-mail

The name of the PDF attachment used for e-mail follows the same naming convention that is used for storing the files on the IFS, namely 000001.PDF, 000002.PDF, and so on. The subject line of the e-mail contains the name of the original file. We recommend that a meaningful message be generated using the e-mail exit.

4.8.5 Results of sending multiple e-mails with user exit

In the example shown in Figure 86, a spooled file containing multiple invoices is segmented, and each component is sent to a different address. An exit program is used to map the mail tag to a valid e-mail address. A custom message is generated based on information provided from a lookup file.

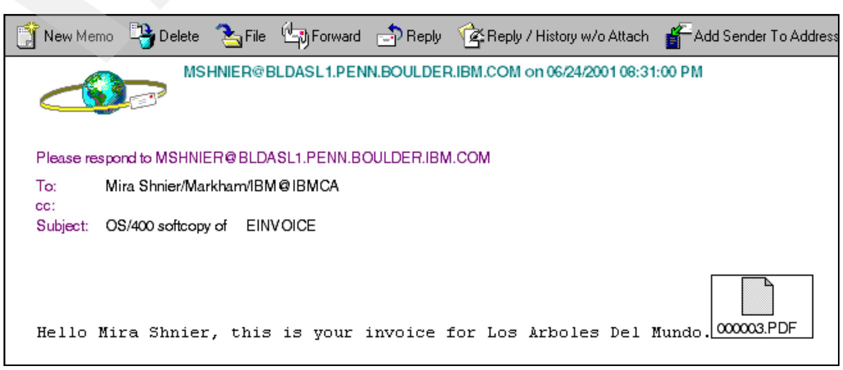
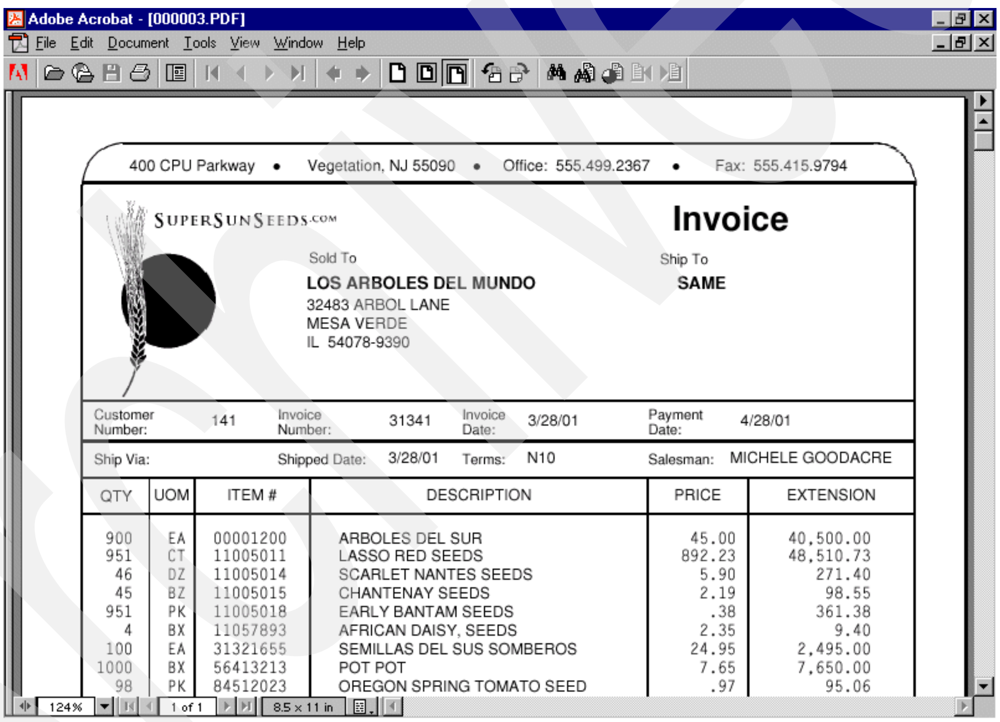


Figure 86. Lotus notes e-mail with a PDF attachment and custom message

Note the following features:

- The sender name, in this case MSHNIER, was specified using the USRDFNDTA keyword. BLDASL1.PENN.BOULDER.IBM.COM is the iSeries server from which this e-mail was sent.
- The subject line contains the name of the original spooled file, in this case EINVOICE.
- The mail message is customized.
- The PDF attachment is given the name 000003.PDF. This invoice was the third of many invoices in the original spooled file. The other invoices from the same spooled file were sent to different addresses.
- The original mail was sent to an address in the format *user@company*. Lotus Notes converts that address into its own format.

Clicking the attachment icon launches Adobe Acrobat. The invoice document, shown in Figure 87, is displayed.



QTY	UOM	ITEM #	DESCRIPTION	PRICE	EXTENSION
900	EA	00001200	ARBOLES DEL SUR	45.00	40,500.00
951	CT	11005011	LASSO RED SEEDS	892.23	48,510.73
46	DZ	11005014	SCARLET NANTES SEEDS	5.90	271.40
45	BZ	11005015	CHANTENAY SEEDS	2.19	98.55
951	PK	11005018	EARLY BANTAM SEEDS	.38	361.38
4	BX	11057893	AFRICAN DAISY, SEEDS	2.35	9.40
100	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	2,495.00
1000	BX	56413213	POT POT	7.65	7,650.00
98	PK	84512023	OREGON SPRING TOMATO SEED	.97	95.06

Figure 87. PDF of the invoice attachment

4.9 Additional considerations

This section describes additional information you may want to consider when designing your PDF generation application.

4.9.1 Other documentation

Please refer to the "Usage Notes" and "Error recovery" sections in the *Infoprint Server for iSeries User's Guide*, G544-5775, for additional considerations that affect the transform and how you use the transformed data. The topics that are covered include:

- Barcodes
- Color
- Error checking
- File size (also see 4.9.2, “PDF file size” on page 135, in this document)
- Find function
- Fonts
- Other unsupported IPDS functions
- Page numbering
- Printing considerations
- User of printer features
- Viewing considerations
- Messages
- Troubleshooting

4.9.2 PDF file size

To ensure document fidelity when a PDF is generated, Adobe fonts are built based on the AFP fonts used and are included in the document. This allows the document to be generated as text as opposed to a raster image. As indicated in the User’s Guide, the PDF generated is increased in size by approximately 110 KB for each font referenced in the input document.

During testing, we observed the results shown in Table 17.

Table 17. Results from testing the PDF file size

File description	Number of pages	Size of PDF
Short JOBLOG	2	58 KB
Medium JOBLOG	19	94 KB
Long JOBLOG	217	493 KB
Single Invoice	1	328 KB
Multiple Invoices	6	337 KB

The JOBLOGS only used one font, whereas the Invoice application used three different outline fonts.

Some users have expressed concern regarding the overhead caused by using multiple fonts in terms of storage space and line traffic. A PTF is being generated to give those users an alternative. With this new function, you specify PDFINCFNT(*NO) as a PSF Defined Option on the PSF configuration object. If the PDF device type is IP40300 or IP40240, then the fonts are not included inline.

If the user chooses not to have the fonts embedded, the IPDS Type 1 font name character string is moved to the PDF font controls. When the document is viewed Acrobat Reader maps IBM's core font names to the equivalent Adobe/client environment set of core fonts. For any font name character strings for which Acrobat does not have an equivalent, Acrobat uses the Adobe multi-master font substitution program to select the available font that constitutes the "best fit".

4.9.3 E-mail exit extras

The primary purpose of the e-mail exit program is to convert a mail tag or the USRDTA field to a valid address for an e-mail. Creating a custom message to display with the PDF attachment comes a close second.

With the information provided to the exit program in the input data structure, there are a number of other things you may consider such as:

- If the lookup fails, send a message to an operator.
- If the lookup fails, copy the PDF file to an area on the IFS.
- Write a record to a log file containing information about the status of each PDF file processed. This may contain information that is not normally written to the data queue, such as the message text. You may find such a file may be easier to manage than the data queue.
- Use other information such as the job name or file name to build the e-mail address or user message.
- Choose not to send a given e-mail.

4.9.4 Using the e-mail exit to customize storage on the IFS

The Infoprint Server function that creates PDF files on the IFS uses a particular directory and file naming convention in an attempt to make the files identifiable and unique. The files and directories created by the PDF transform are owned by the spooled file owner and have a public authority of *EXCLUDE.

This may not suit the needs of all customers. As an alternative, it is possible to set up the PDF transform as if you are going to perform the *MAIL function and specify the use of an e-mail exit program. In that program, you can:

- Have complete control over the target directory structure
- Rename the PDF file
- Change the object authorization of the PDF file

Under normal circumstances, when you select *MAIL, the e-mail process cleans up the PDF files it generates once it is finished processing the entire spooled file. However, at the point in time when the exit program is running, the PDF file is accessible. Its name and path are passed to the program as part of the input data area. The trick is to “grab” these files in the using the exit program and store them in a manner of your own choosing, before they are erased.

In the exit program in E.1, “E-mail exit program” on page 343, the file is only copied if the e-mail address lookup fails. You can modify that program to copy all files that it processes, instead of sending them as e-mails. At that point, you can change the security settings or give it a new name.

4.9.5 Other information available in the exit

Earlier discussions in other parts of this chapter focused on using the Mail Tag field (as set by the STRPAGGRP keyword or CRTAFPDTA) or the USRDATA field as the key to the address lookup operation. The input data structure also contains the job name, number and user, and the original spooled file name and number. This information can also be used as part of the address lookup, message customization, or other processing of the PDF file.

You can take this one step further and use this information to call system API QUSRSPLA programs that let you retrieve the entire spooled file attributes. That gives you additional information, such as the form type or the program used to create the file. For more information about QUSRSPLA, see *AS/400 System API Reference*, SC41-5801, or the iSeries Information Center on the Web at:

`http://publib.boulder.ibm.com/html/as400/v5r1/ic2924/index.htm`

4.9.6 Sending a file to multiple destinations

It is possible to use USRDFNDTA or STRPAGGRP to specify multiple destination addresses for a given e-mail. There is a limit of 250 characters to the length of the values that each of those support.

If you choose to maintain the list of addresses elsewhere, the list can be imported by the e-mail exit. The address field that is available in the output data structure may be as long as 16 MB if you require.

If you specify multiple addresses in the exit program, each one must be delimited by single quotation marks, for example:

```
'bill_s@last.leaf.com' 'lee_s@paydaypete.com'
```

An example of using this technique may be to periodically distribute a company financial report to a number of its executives. You don't want to have to modify the e-mail exit program every time an employee changes. Use a separate application to maintain the names. Place the name of the distribution list in the USRDTA field. The e-mail exit can then import the current list of addresses as needed.

4.9.7 Using open and close to segment a spooled file

Generally speaking, the iSeries works more efficiently if large spooled files remain intact. There is a certain amount of overhead if you print or manage multiple small files versus one large file. Nevertheless, if you are not able to use the DDS keywords to insert the group tags, and the data is not predictable enough for CRTAFPDTA, the final solution may be to modify the application program simply *close* and *open* the printer file at each group boundary. This generates multiple spooled files that are then converted to PDF independently.

Specify a value of *NO for the *Multiple PDF output* parameter in the PSF configuration object if you are breaking up the file within the application before you pass it to the PDF output queue.

4.9.8 Proliferation of the PSF configuration object and device description

You need to give some consideration to the proliferation of PSF Configuration objects and printer device descriptions needed for the various PDF creation tasks you want to do. Clearly the single files versus multiple files have to go to separate queues, as do files that are stored on the IFS using the *STMF function versus those files being sent by e-mail.

Beyond that, you may choose to create different configurations to deal with different applications. For example, you may want to put a marketing message on your invoice e-mails, but not on purchase orders.

4.9.9 Using the PDF data queue

The IPDS to PDF process creates entries in a data queue object if you specify one in the PDFDTAQ parameter in the PSF configuration object. To use this function, the data queue must exist and have a minimum length of 752 characters. Use the iSeries QRCVDTAQ API to read the entries. For more information on QRCVDTAQ, see the iSeries Information Center on the Internet at:

`http://publib.boulder.ibm.com/html/as400/v5r1/ic2924/index.htm`

Using the data queue helps you identify other types of problems that may occur. For example, if you forget the single quotes on the e-mail address in the LOOKUP table, it is treated as an invalid address and the file is not sent. A return value of ‘05’ is logged in the data queue.

You can use the information in the data queue as a trigger to begin a secondary process if you store the file on the IFS. For example, once you know a file is successfully placed in the IFS, use the information in the data queue to post it on a Web site.

You can find the layout of the fields in the data queue in E.2, “PDF data queue structure” on page 346.

4.9.10 Other e-mail solutions

Infoprint Server for iSeries uses the SNDDST command to perform its e-mail functions. It addresses the document, adds message text, and attaches the PDF file.

Other tools are available on the iSeries that add more sophistication to the e-mail function. You may choose to store the PDF files on the IFS for subsequent processing by one of these other methods:

- Lotus Domino for iSeries provides a rich application development environment that tightly integrates mail, workflow, document management, and many other important features.
- The Java Mail API is used to create a mail client capable of sending multimedia mail messages, as well as enabling a full fledged Internet Mail Access Protocol (IMAP) implementation supporting folders, authentication, and attachment handling. The JavaMail implementation provides MIME encoding and decoding capabilities natively, allowing developers to focus on content and business logic, not data conversion.
- The QTMMSENDMAIL system API is used to enable mail applications. It is quite powerful but may require a moderate to extensive programming effort.

4.10 Comparing AFP and PDF indexed files

Infoprint Server for iSeries can generate PDF files and AFP portable format. This section provides information about viewing the capabilities of the Adobe Acrobat and the AFP Workbench.

From a more general point of view, Adobe Acrobat and Adobe Acrobat Reader were created to work with a structured document as a book. The AFP Workbench is designed to work with AFP resources documents and files. This section may be used to define which document format may be used.

4.10.1 Considerations choosing the file format

The most important point is to determine the file format you want use. The file format required by a customer may depend of the use and of the integration in his IT solution.

- **Adobe Acrobat**: Acrobat may be used to view and modify document appearance. Additional elements, such as notes or bookmarks, may be added with Acrobat as shown in 4.12, "Viewing and indexing PDF files with Adobe Acrobat" on page 142. Acrobat does not provide an automatic way to define a single bookmark at the same place across several pages.
- **Adobe Acrobat Reader**: Adobe Acrobat Reader has the same viewing capabilities as Adobe Acrobat. However, you cannot modify the document appearance or add bookmarks or notes.
- **AFP Workbench:** AFP Workbench or called AFP Viewer is part of Client Access and may be used to see AFP spooled files and AFP resources. In addition, the AFP Viewer supports AFP indexed spooled files, which may be indexed on the IBM @server zSeries, pSeries, and iSeries servers.

Depending on your environment, AFP Workbench or Adobe Acrobat may fit your requirements. Indexing information is used for archiving purposes. If your archiving environment uses the AFP architecture (see Chapter 16, "Using Content Manager OnDemand for iSeries" on page 311, for more information), the AFP Workbench offers better capabilities.

Due to the popularity of Adobe Acrobat Reader, the PDF format offers a good way to make information available. Table 18 provides a comparison between Adobe Acrobat 4.5 and the AFP Workbench 1.54, which may be different depending on your release of the product.

Table 18. Adobe Acrobat comparison

Functions	Adobe Acrobat	AFP Workbench
Indexing manually	You can create a bookmark in a PDF file. See 4.12.2, "Creating a new bookmark" on page 144, for additional information.	You cannot create an index with the AFP Workbench. AFP Workbench work with index information created with the Group DDS keyword or using their CRTAFPDTA capabilities of the Infoprint Server.
Indexing automatically	Bookmarks may not be created automatically across the document.	You cannot create an index with the AFP Workbench. AFP Workbench works with index information created with the Group DDS keyword or using their CRTAFPDTA capabilities of the Infoprint Server.
Find	PDF files may be scanned with the "find" option to search a specific text stream.	AFP files may be scanned with the "find" option to search a specific text stream.
Search indexed information	Bookmarks are displayed in a navigation pane. The navigation pane can be activated by clicking the corresponding icon.	Indexed information may be found with "Find Group" icon or option. The group information is displayed an additional window.

Functions	Adobe Acrobat	AFP Workbench
Security	The appearance of PDF files may be changed with Adobe Acrobat. Other tools or utilities may be used to changed the appearance of the document and text. A password may be set to the PDF file to enhance the security.	The AFP data stream cannot be modified with AFP Workbench or other IBM utility.
Size	The size of PDF files appear to be much bigger as the same data in an AFP indexed format. See 4.9.2, "PDF file size" on page 135.	

4.11 Viewing AFP indexed files with AFP Workbench Viewer

The Advanced Function Presentation Workbench for Windows is a platform for the integration of AFP-enabling applications and services. AFP Workbench is part of Client Access/400. The Viewer provides the ability to browse the following types of files residing on your workstation or network:

- AFP documents, such as those generated by IBM's Document Composition Facility (DCF) and BookMaster for AFP device types
- AFP documents created by AFP Conversion and Indexing Facility (ACIF), which processes S/390, RS/6000, or iSeries line data using a page definition (PAGEDEF)
- AFP overlays and page segments, such as those created by IBM's Overlay Generation Language (OGL)
- Output from workstation applications that is converted to an AFP document using the IBM AFP Printer Driver for Windows (supplied with Workbench)
- Displaying page segments and overlays referenced by an AFP file where those resources are either contained in the file or located in the resource directories defined on the More Preferences dialog box

This section provides a short introduction about how to work with groups information and how to navigate in an indexed AFP document.

4.11.1 Indexing an AFP file

There are several methods to create indexed files on the iSeries. Here are two methods to create AFP indexed files:

- Using DDS group keyword
- Using CRTAFPDTA

AFP files created and indexed on S/390 or AIX systems are supported as well. In this case, it is important that all resources are included in the file.

4.11.2 Using AFP Workbench with indexed AFP files

The AFP Workbench can display indexed AFP files. A group and trigger may be used to navigate in the file. A *group* is a named collection of pages that forms a logical subset of a document. You may use one of the following ways to open an indexed file:

- Open Operations Navigator. Expand the system environment structure and select **Basic Operation** to access to the Output Queue containing the spooled files. Double-click the AFP spooled file to invoke the AFP Viewer. Note that the AFP resources have to assessable to the AFP Viewer.
- Use the CRTAFPDTA command to produced a portable AFP file with all resources included.

Figure 88 shows an indexed AFP file displayed with AFP Workbench.

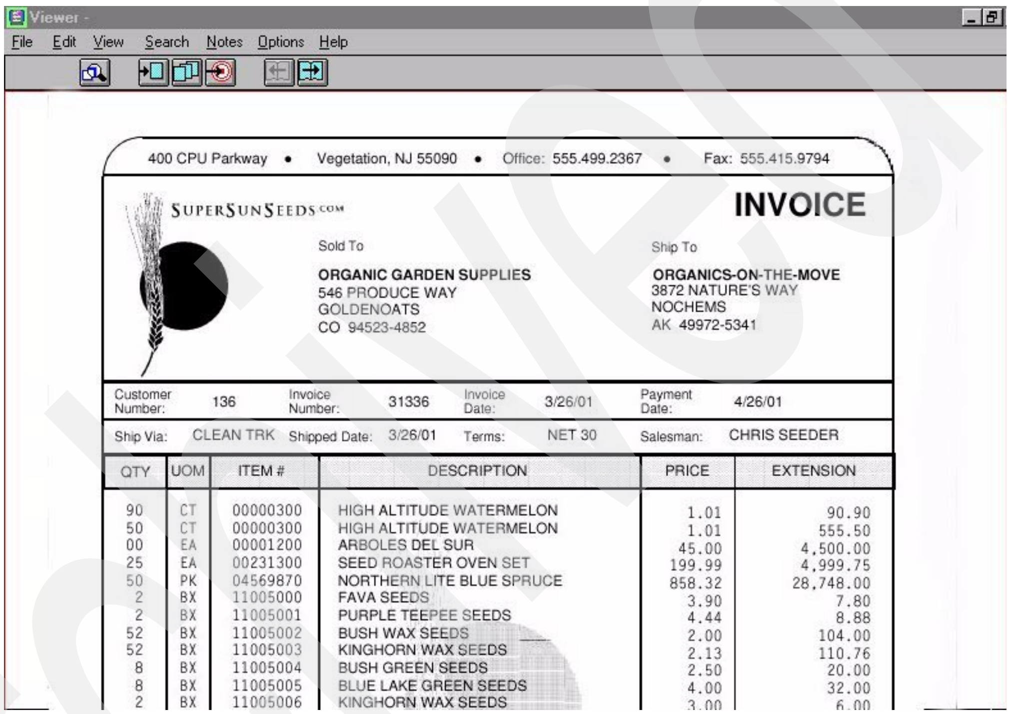


Figure 88. AFP Workbench

The AFP Workbench is working with a group. Groups are identified in an AFP document using the indexing functions of the AFP Conversion and Indexing Facility (ACIF) or AFP Application Programming Interface (AFP API) programs.

From this point, you may select the Group icon to access to the indexed information as shown in Figure 89.

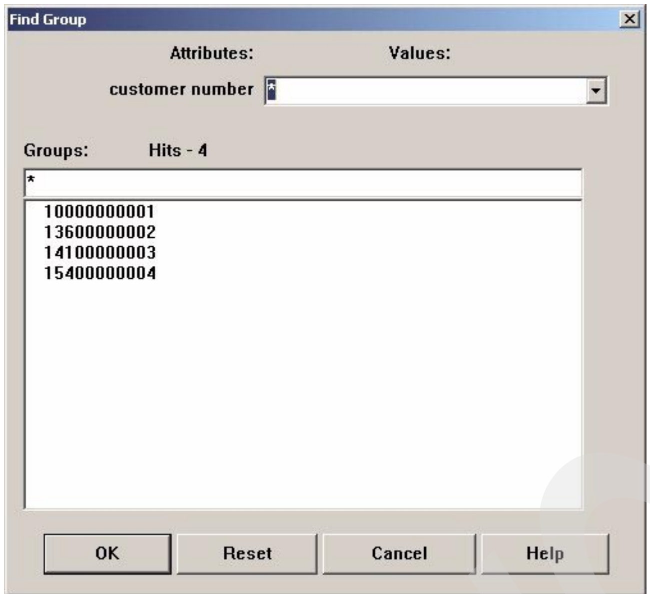


Figure 89. Find Group

Use the Find Group dialog box to view the first page in a group of pages. A group of pages is defined by indexing tags included in an AFP document file. This function is not available if the file does not contain group-level indexing tags.

Select an attribute value to display a list of groups that have been tagged with the selected value for the attribute. Select additional attributes and associated values to limit the list of groups as shown in Figure 89.

A list of attributes appear as defined in the file. The values for the selected attribute are displayed in the Value list. Select a value for one of the listed attributes as explained here:

1. Select the arrow and choose the value you want to see. The groups that have been tagged with the selected values are displayed.
2. Select a group from the list and click **OK**. Or double-click a group in the list to select the group and display the first page in the group.

4.12 Viewing and indexing PDF files with Adobe Acrobat

Infoprint Server creates a full text searchable PDF. For example, you can search the text in the document to find a specific word or page. An other example is to place bookmarks in the document to help the user to navigate in a PDF document.

Infoprint Server does not have the ability to place bookmarks automatically in the PDF file. However, Adobe Acrobat may be used to add bookmarks as explained in the following sections.

4.12.1 Introduction to bookmarks in a PDF file

A bookmark is a link to representative text in the navigation pane of Adobe Acrobat or Acrobat Reader. Each bookmark in the navigation pane goes to a different view or page in the document. You can use electronic bookmarks, as you would paper bookmarks, to mark a place in a document to which you want to return. You can also use bookmarks to modify the view of its destination, therefore, directing your reader's attention where you want it. Bookmarks allow you to jump within a PDF document, to another document (PDF or non-PDF), or to a Web page. And you can use them to perform such actions as playing a movie or sound clip, executing a menu item, or submitting a form.

You can use bookmarks to navigate in a PDF document using Adobe Acrobat or Adobe Acrobat Reader. However, Adobe Acrobat Reader does not have the ability to add or organize bookmarks.

Figure 90 shows an example of a small indexed PDF file using Adobe Acrobat.

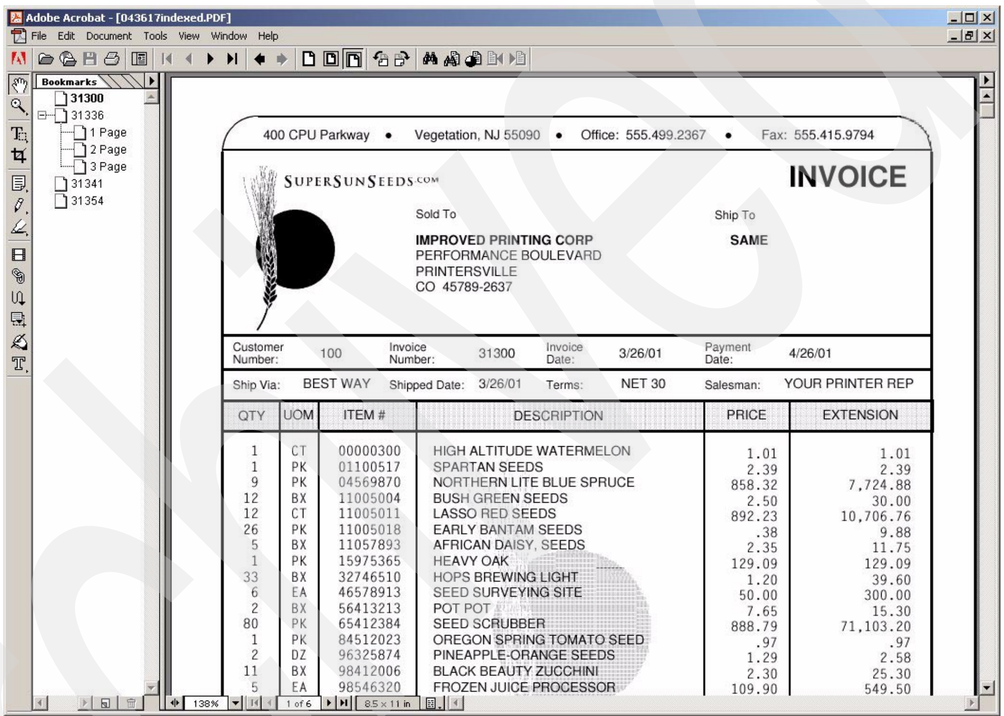


Figure 90. Indexing your PDF files with Adobe Acrobat

The right part of the window shows a document page with some of the supported elements, like the logo (image) line box and text. Adobe Acrobat has been used to mark the invoice number as a bookmark. The indexed invoices numbers are displayed left pane of Figure 90 as a organized structure. This structure may be use to navigate in the document from one invoice to the other one.

The tree structure has been organized into a hierarchy to show that the invoice contains several pages. In this case, a second bookmark was created with a second-level hierarchy.

You may easily note the difference between Adobe Acrobat and Adobe Acrobat Reader. Acrobat Reader can read and display the complete PDF file. You may use the tree structure and search capability, but Adobe Acrobat is required to create bookmarks as shown in this following sections.

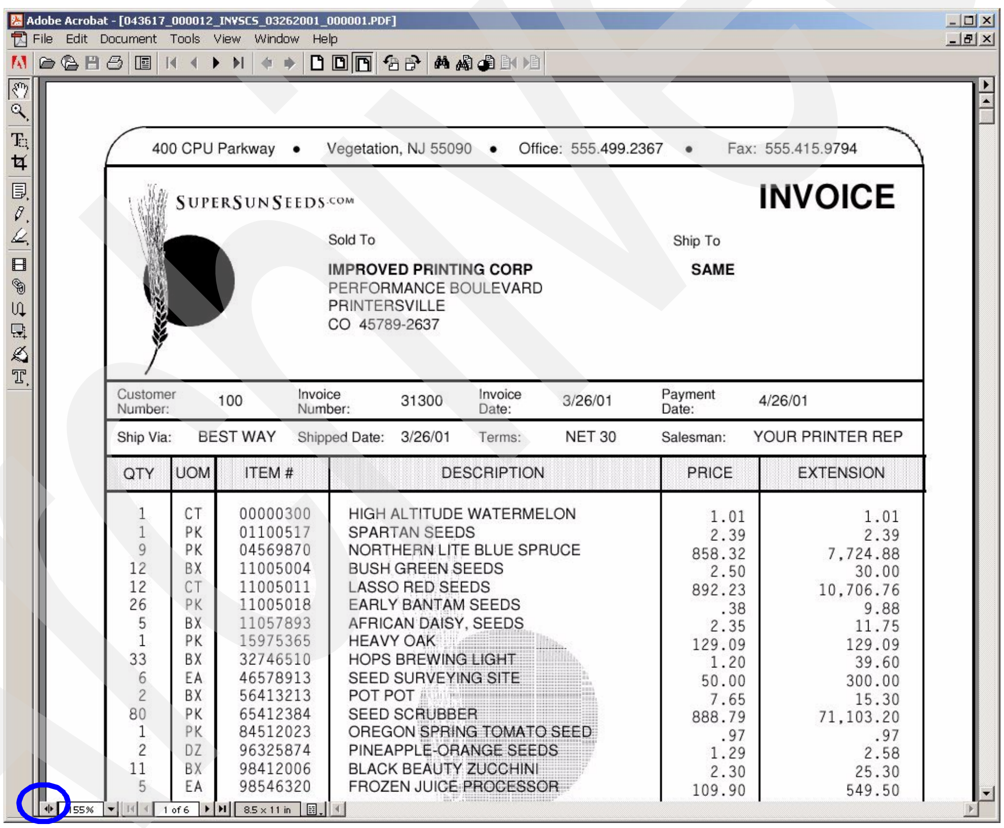
Barcode displayed in a PDF file

You may note that most of barcode symbology is supported by the IPDS/AFP transform. The transform converts the barcode as an image placed in the PDF file to be displayed. The quality of this image depends on the transform configuration set in the PSF configuration object and may not be used by a scanning device like a regular printed barcode.

4.12.2 Creating a new bookmark

This section describes how to create new bookmark in a PDF document using Adobe Acrobat Version 4.05c. Different version of Acrobat may require different functions or tasks.

Open the PDF file with Adobe Acrobat to start. The PDF file is displayed in the Acrobat window as shown in Figure 91.



400 CPU Parkway • Vegetation, NJ 55090 • Office: 555.499.2367 • Fax: 555.415.9794

SUPERSUNSEEDS.COM

INVOICE

Sold To

IMPROVED PRINTING CORP
PERFORMANCE BOULEVARD
PRINTERSVILLE
CO 45789-2637

Ship To

SAME

Customer Number: 100 Invoice Number: 31300 Invoice Date: 3/26/01 Payment Date: 4/26/01

Ship Via: BEST WAY Shipped Date: 3/26/01 Terms: NET 30 Salesman: YOUR PRINTER REP

QTY	UOM	ITEM #	DESCRIPTION	PRICE	EXTENSION
1	CT	00000300	HIGH ALTITUDE WATERMELON	1.01	1.01
1	PK	01100517	SPARTAN SEEDS	2.39	2.39
9	PK	04569870	NORTHERN LITE BLUE SPRUCE	858.32	7,724.88
12	BX	11005004	BUSH GREEN SEEDS	2.50	30.00
12	CT	11005011	LASSO RED SEEDS	892.23	10,706.76
26	PK	11005018	EARLY BANTAM SEEDS	.38	9.88
5	BX	11057893	AFRICAN DAISY, SEEDS	2.35	11.75
1	PK	15975365	HEAVY OAK	129.09	129.09
33	BX	32746510	HOPS BREWING LIGHT	1.20	39.60
6	EA	46578913	SEED SURVEYING SITE	50.00	300.00
2	BX	56413213	POT POT	7.65	15.30
80	PK	65412384	SEED SCRUBBER	888.79	71,103.20
1	PK	84512023	OREGON SPRING TOMATO SEED	.97	.97
2	DZ	96325874	PINEAPPLE-ORANGE SEEDS	1.29	2.58
11	BX	98412006	BLACK BEAUTY ZUCCHINI	2.30	25.30
5	EA	98546320	FROZEN JUICE PROCESSOR	109.90	549.50

Figure 91. Adobe Acrobat window

Select the **hide/show** bookmarks icon (circled in the bottom left corner of the example in Figure 91) to open the navigation pane shown in Figure 92.

Click the **Bookmarks** tab in the navigation pane to bring the Bookmarks palette to the front.

Use the Text select tool (circled in Figure 92) to mark the text you want use as bookmark label. Right-click and select the **New Bookmark** option. The new bookmark appears in the navigation pane. Repeat the previous action for any additional bookmark.

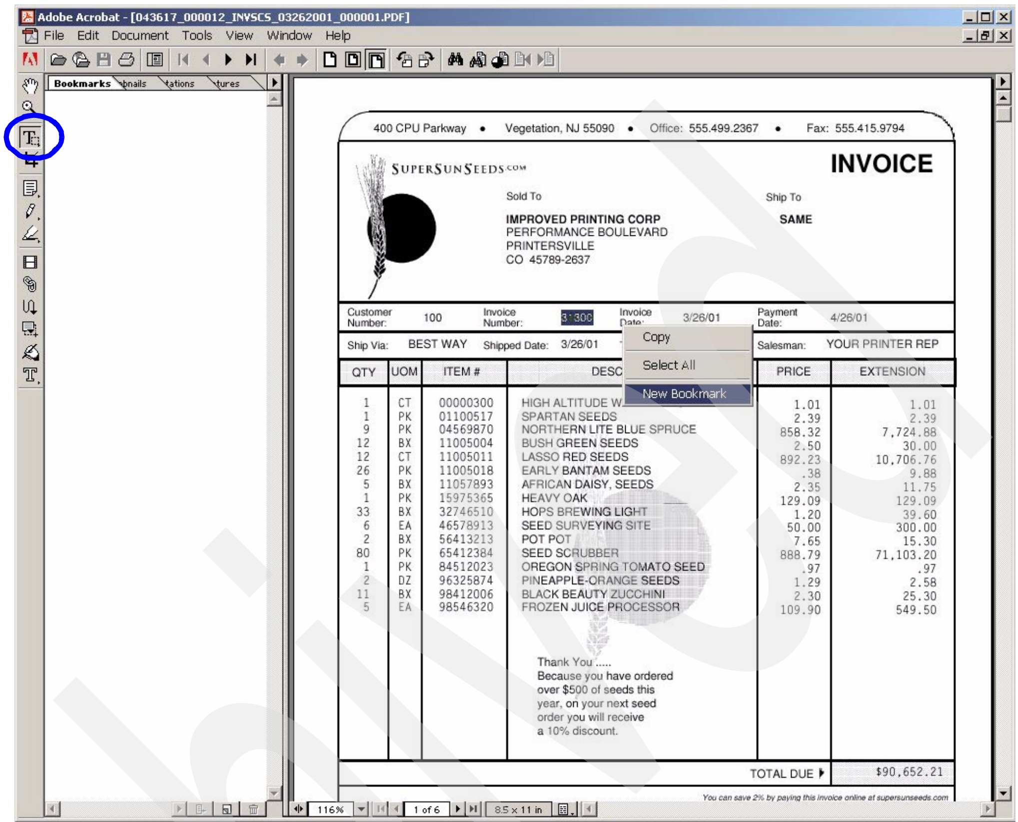


Figure 92. Adobe Acrobat navigation pane for creating a new bookmark

After all bookmarks are created, you can edit and change the bookmark label. Select the bookmark and click the highlighted bookmark label. The label appears in edit mode. At this point, you can add, remove, or complete the label.

Make sure the correct location and magnification are set. Then, go to another page in the document and test the bookmark.

Figure 93 shows the Adobe Acrobat window with the additional bookmarks in the navigation pane. In this example, the invoice number is marked on each page as a bookmark. The three bookmarks for invoice 31336 are edited and the page information is added manually.

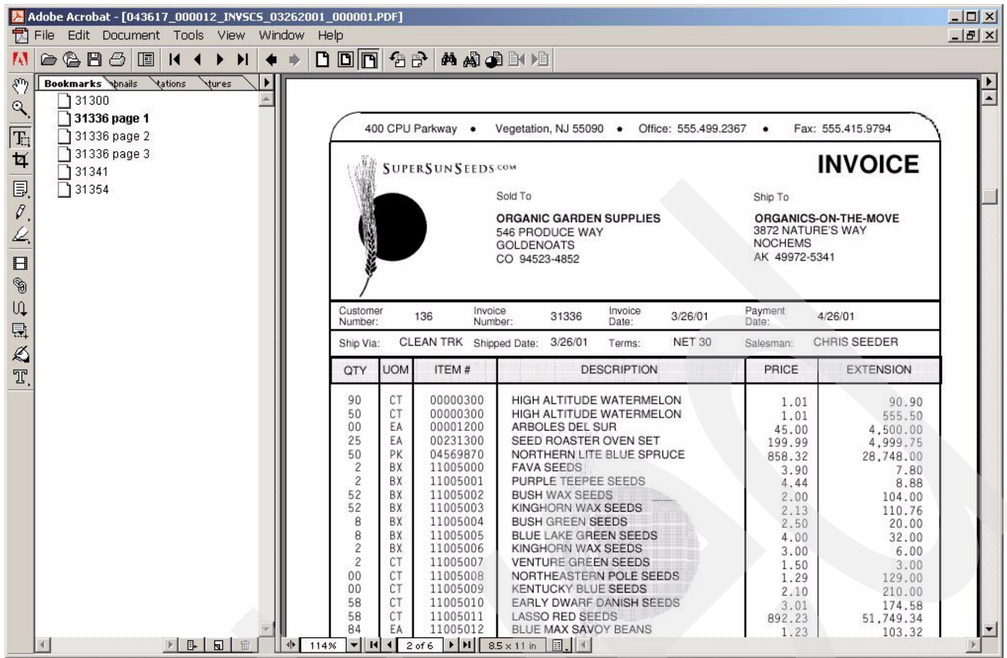


Figure 93. Adobe Acrobat with additional Bookmarks

Adobe Acrobat Bookmarks

Unfortunately it is not possible to add multiple bookmarks for a unique field (the invoice number in the example) across the document with Adobe Acrobat.

4.12.3 Adobe bookmark hierarchy

You can give an additional professional touch to your PDF files and create a hierarchy structure to your bookmarks. You can nest a list of bookmarks to show a relationship between the common pages of invoice 31336 as described in 4.12.2, "Creating a new bookmark" on page 144. Nesting creates a parent/child relationship, and you can expand and collapse this hierarchical list as desired.

To nest a bookmark under another bookmark, follow these steps:

1. Click the bookmark or range of bookmarks you want to nest.
2. Drag the icon or icons underneath the first letter in the parent bookmark; a black bar shows the position of the icons.
3. Click **OK**. The bookmark is nested. However, the actual pages remain in their original location in the document.

Regarding the invoice example, begin with the last page of the invoice page that was set, which in this example is page 3 of invoice 31336 and then pages 2 and 1. Because the last move is placed above the last bookmark inside a hierarchy, the bookmark will be in the correct sequence.

Figure 94 shows how the bookmark hierarchy appears in Adobe Acrobat.

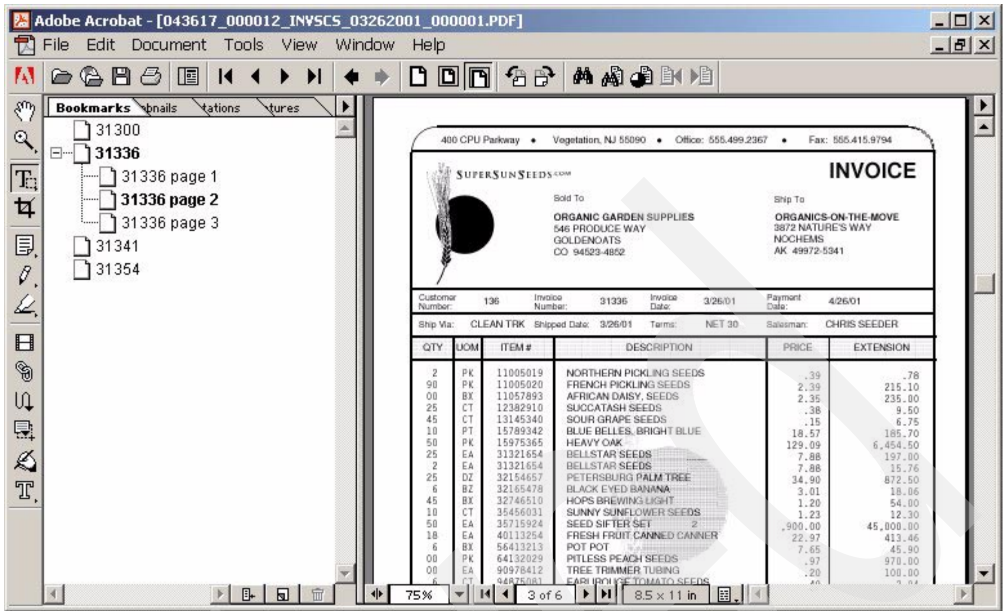


Figure 94. Nesting a bookmark

In this example, an additional bookmark is created for invoice 31336 page 1. This has the advantage of showing a complete hierarchy structure in the second level and the invoice number with a + symbol when the structure is collapsed. Note that the page indication was not added for the first structure level.

Chapter 5. Infoprint Server for iSeries: CRTAFPDTA

The Create AFP Data (CRTAFPDTA) command is one of the four major components of the Infoprint Server for iSeries product. This chapter describes how to use this command. It includes:

- A conceptual overview
- A discussion of the parameters that are used for each of the functions of the command
- A description of the structure and naming of the output files
- Information on how to use the output with the AFP Viewer
- A comparison to the OS/390 product, AFP Conversion, and Indexing Facility

The CRTAFPDTA command can be viewed as a multi-purpose tool that expands the use of AFP applications on the iSeries.

The CRTAFPDTA command transforms line (*LINE) or mixed (*AFPDSLINE) data into AFP (*AFPDS) for printing and viewing. It can index a document for viewing, archiving, or document retrieval. It optionally retrieves and packages AFP resources needed for printing or viewing. The output of CRTAFPDTA is stored in the Integrated File System (IFS) on the iSeries server.

5.1 Scenarios that use CRTAFPDTA

The following scenarios are examples of where the CRTAFPDTA is used to enhance or enable a new AFP based application:

- If you use the AFP Viewer, CRTAFPDTA can be used to add index and group records to the file. This assists customer service representatives to quickly access specific information directly. The AFP Viewer is shipped as part of Client Access Express, may be downloaded for free as a plug-in, or may be purchased as part of a stand-alone product – IBM AFP Workbench for Windows V2.
- If you use Infoprint Server for iSeries to generate multiple PDF files, CRTAFPDTA can be used to insert group records that are used to segment the file. The information in the group records can also act as the key for a lookup operation to determine the address if using the e-mail function.
- If you distribute spooled files from the iSeries to another system, use CRTAFPDTA to select and concatenate external resources. This avoids having to send them manually as a separate operation.
- If you plan to store documents for an extended period of time, and reproduce them at a later date, use CRTAFPDTA to concatenate the external resources to ensure the documents print correctly even if the resources, such as overlays or page segments, change.
- If you generate line data in your spooled files, and use Infoprint Designer for iSeries or other tools that generate page definitions, you may choose to use CRTAFPDTA to convert the line data to compose AFPDS for use with the AFP Viewer or to print to an ASCII printer using Host Print Transform.

- If you use an archive product, such as Content Manager Common Server, you may choose to use CRTAFPDTA to index the file and concatenate external resources.

5.2 CRTAFPDTA overview and concepts

The function of CRTAFPDTA can be looked at as four separate, but related tasks:

- Convert line and mixed-mode data to AFPDS
- Index the document
- Retrieve AFP resources
- Merge the results of the other three steps to one output file

5.2.1 Example of using CRTAFPDTA

Page definitions are powerful tools that separate document formatting from the application. Infoprint Designer now makes it very easy to create page definitions. Page definitions are applied only at print time. Therefore, a document that is built this way cannot be viewed directly using the AFP Viewer. To make line data documents that use page definitions available for viewing, including the appropriate AFP resources, use CRTAFPTA. This has the added advantage of adding index records to the file for easy navigation. Figure 95 shows the flow of data for the CRTAFPDTA command for an application that generates data for the AFP Viewer. The flow is explained here:

- An iSeries application (1) generates a spooled file, using a printer file (2), and places the output in an output queue (3). In this example, the spooled file is generated as line data.
- The CRTAFPDTA command (4) takes the spooled file and creates one to four files in the Integrated File System (5), depending on the options selected. External resources residing in iSeries libraries (6) are referenced and optionally copied to the IFS with the spooled data.
- In this example, the merged output is process by the Print AFP Data, (PRTAFPDTA) command (7) to generate a new spooled file. The new spooled file is in *AFPDS format and contains index records. The resources, such as overlays and page segments, are included inline. The resulting spooled file can be viewed using the AFP Viewer via the Client Access Express Navigator (8).

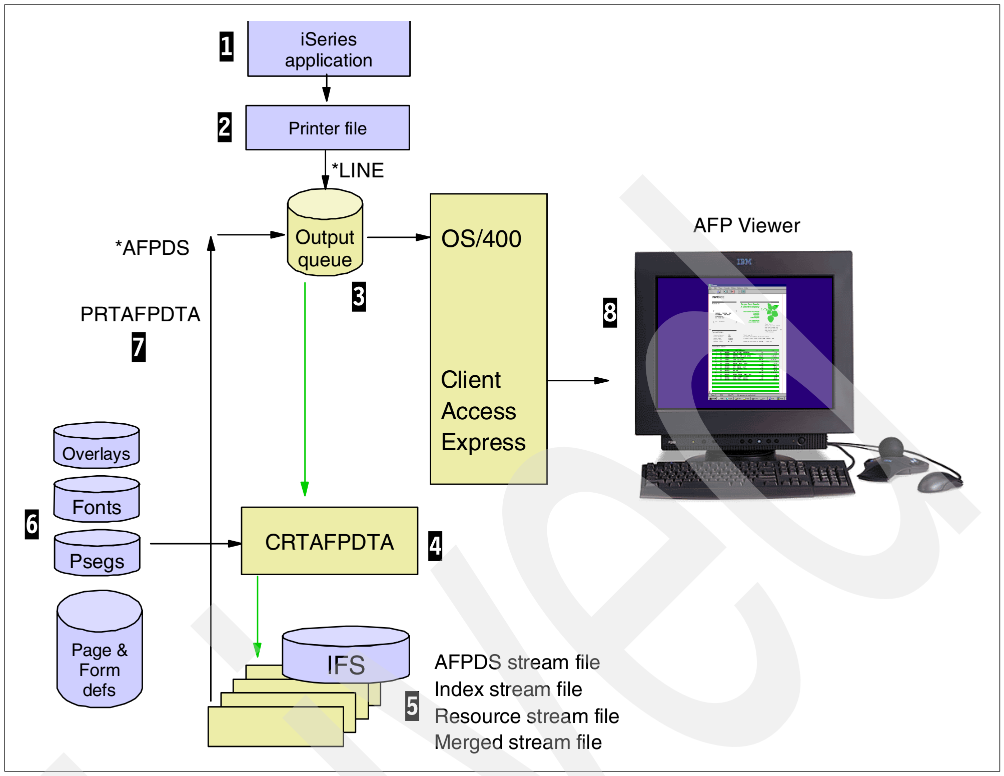


Figure 95. Example of CRTAFPDTA data flow for viewing

Section 5.5, "Using the AFP Viewer with the output of CRTAFPDTA" on page 161, describes other ways to use the AFP Viewer with the output of CRTAFPDTA.

5.2.2 Understanding the data streams used by CRTAFPDTA

Depending on the what is being done to a spooled file, there are different ways the data can be built and yet still be considered part of the AFP architecture.

Line data

When a spooled file is generated with DEVTYPE(*LINE), it contains line data. This is the simplest form of data used by AFP applications. Line data is very similar to SCS. Except for optional carriage control and font selection bytes, there is no formatting information in the data. The instructions on how the data is to be presented on the page comes from the external resources, the page definition, and the form definition. At print time, PSF/400 uses the formatting information in those resources and sends the spooled file directly to an IPDS printer. Other resources, such as overlays, page segments, and fonts, are referenced in the page definition and form definition.

Line data is becoming more prevalent on the iSeries because Infoprint Designer normally works with that format. There have been other recent enhancements to page definitions. For more information on line data and page definitions, see Chapter 12, "Using form definitions and page definitions" on page 255.

Mixed data

Spooled files generated with DEVTYPE(*AFPDSLINE) are said to contain mixed data. These spooled files consist mainly of line data with a limited number of AFP structured fields. A page definition and form definition are still needed to format this type of data on the page. This type of file is not usually generated using traditional iSeries programming.

AFPDS or MO:DCA-P
Spooled files that contain text plus formatting instructions are said to be in the Advanced Function Presentation Data Stream (AFPDS) format. The more current name is Mixed Object Document Content Architecture for Presentations (MO:DCA-P). When you generate a spooled file with DEVTYPE(*AFPDS), the data is in this format. Resources, such as overlays, page segments, and fonts remain external to the spooled file and are brought in by PSF/400 at print time.

> **Note**
>
> One exception is that the iSeries spooled file generator builds a Form Definition inline, if one is not specified in the printer file parameter FORMDF. This inline form definition is based on the spooled file attributes such as paper source and duplex.

This type of data stream is sometimes referred to as *composed AFP*. CRTAFPDTA can convert line data or mixed data to composed AFP.

Portable AFP
This refers to spooled files that have the external resources included inline as a resource group. The resources that may be put inline are fonts, overlays, page segments, or form definitions. These spooled files can be moved from system to system without being concerned if the resources reside on that target machine. Traditional iSeries programs cannot include resources inline. CRTAFPDTA can be used to generate portable AFP.

Index object file
This may be a separate file or merged with the AFP document. It contains Index information with pointers to Group tags in the AFP document.

5.2.3 Advanced uses of CRTAFPDTA

CRTAFPDTA was migrated from the AFP Conversion and Indexing Utility program available on OS/390. Some options and parameters have been included on the iSeries for reasons of consistency and compatibility. These parameters may not have much relevance for most iSeries users who use traditional programming techniques.

The term “traditional” programming is used in a number of places in this chapter. This refers to using a high-level-language program, such as RPG or COBOL, that references a printer file to generate a spooled file. In contrast, a “non-traditional” application may be one that uses APIs or the AFP Toolbox to generate the data. Spooled files generated on another system, such as AIX or OS/390, and transferred to the iSeries may also be considered “non-traditional”.

These “non-traditional” techniques may generate spooled files that have features that are not otherwise available through “traditional” programming techniques on the iSeries. This chapter focuses on functions of CRTAFPDTA as they apply to “traditional” spooled files. For information on dealing with the more advanced functions, see the *Infoprint Server for iSeries User’s Guide*, G544-5775, or the *AFP Conversion and Indexing Facility: User’s Guide*, S544-5285.

5.3 CRTAFPDTA command details

This section describes in detail each of the four tasks that CRTAFPDTA performs, as well as the parameters used by each task.

5.3.1 Common parameters

The following parameters are used by CRTAFPDTA for any of the tasks that it performs.

5.3.1.1 Spooled file name, job, and number (FROMSPLF, JOB, SPLNBR)

As with most any other command that acts on iSeries spooled files, you need to identify the spooled file by the job that created it and its unique name and number. The spooled file may contain line (*LINE), mixed (*AFPDSLINE), or AFP data (*AFPDS). It is important to note that the iSeries default spooled file type of *SCS is not supported.

5.3.1.2 Form definition and Library (FORMDF)

The form definition to use with this operation must be specified. You may name one here explicitly. Or you can designate CRTAFPDTA to use the form definition already associated with the spooled file by specifying *SPLF. Some applications may generate the form definition in line with the spooled data, in which case you would say *INLINE. This is the case for *AFPDS spooled files generated using normal programming and printer files on the iSeries.

5.3.1.3 To stream file (TOSTMF)

A file containing AFPDS is always generated. Use this parameter to name the AFPDS stream file or specify the target directory. For information on the naming convention used in this parameter, see 5.4, “Locating and naming the output of CRTAFPDTA” on page 160. If the original spooled file was already in the AFPDS format, the output file is essentially the same, with few modifications.

5.3.1.4 Image output (IMAGEOUT)

If you choose *IOCA for this parameter, CRTAFPDTA converts all image data to uncompressed IOCA format. This may increase the size of the output generated. IBM recommends that you select the default value of IMAGEOUT(*ASIS) unless you have a specific requirement for IOCA images.

5.3.2 Converting line and mixed-mode data to AFPDS

There are a growing number of tools and method available for the iSeries to create and work with page definitions and form definitions to format line data. These include:

- Infoprint Designer for iSeries
- Page Printer Formatting Aid
- enhancements to DDS for Record Format Line Data
- Java class LineDataRecordWriter

Each of these is described elsewhere in this document. They share the common advantage in that formatting of the output is separate from the application that generates the data.

There are a few limitations to using spooled files generated as *Line or *AFPDSLINE on the iSeries. Host Print Transform, which is used to print to ASCII printers, does not support these data streams. Nor does the AFP Viewer.

These functions do support composed AFP. Spooled files that are generated since *LINE can be processed by CRTAFPDTA to convert them to composed AFP. This file can be re-spooled and subsequently sent to an ASCII printer configured with TRANSFORM(*YES) to use Host Print Transform, or it can be viewed with the AFP Viewer.

5.3.2.1 Comparison to CVTLINDTA

A new parameter for printer files, Convert Line Data (CVTLINDTA), is an alternate way to convert spooled file that are in *LINE format to AFPDS. CVTLINDTA works differently than CRTAFPDTA. It is a parameter of the printer file used with DEVTYPE(*LINE). It causes the resulting spooled file to be generated as AFP. Table 16 on page 96 compares these two techniques. If the main reason for needing to use the function is to print to an ASCII printer, the CVTLINDTA command may suffice. Users of the AFP Viewer may choose to use CRTAFPDTA to take advantage of the indexing capability for faster retrieval of specific information.

5.3.2.2 Parameters used to convert line data to AFPDS

This process takes place any time the input spooled file contains line or mixed mode data.

To stream file (TOSTMF)

The composed AFP data is stored in the IFS file and directory as defined in the To stream file (TOSTMF) parameter. For information on the naming convention used in this parameter, see 5.4, “Locating and naming the output of CRTAFPDTA” on page 160.

Page definition and library (PAGDFN)

Specify the name of the page definition to use to format the line or mixed mode spooled file. You may name it explicitly with this parameter or reference a page definition that is already specified by the spooled file. A page definition may be placed inline by “non-traditional” programming techniques.

AFP characters (AFPCHARS)

Use this parameter to specify up to four coded font names or reference the fonts already specified in the spooled file attributes. These are used if the spooled file was designed to use Table Reference Characters (TRCs). This is not commonly used for “traditional” iSeries applications.

5.3.3 Indexing the document

Indexing is one of the primary tasks for which CRTAFPDTA is used. Indexing allows users of the AFP Viewer to quickly access pages in large spooled files. Pages are grouped based on specific information such as Customer Number. This can aid customer support personal in retrieving the necessary information quickly, rather than doing a string search in the entire document. See Figure 96 for an example of viewing a spooled file that contains group records created using CRTAFPDTA.

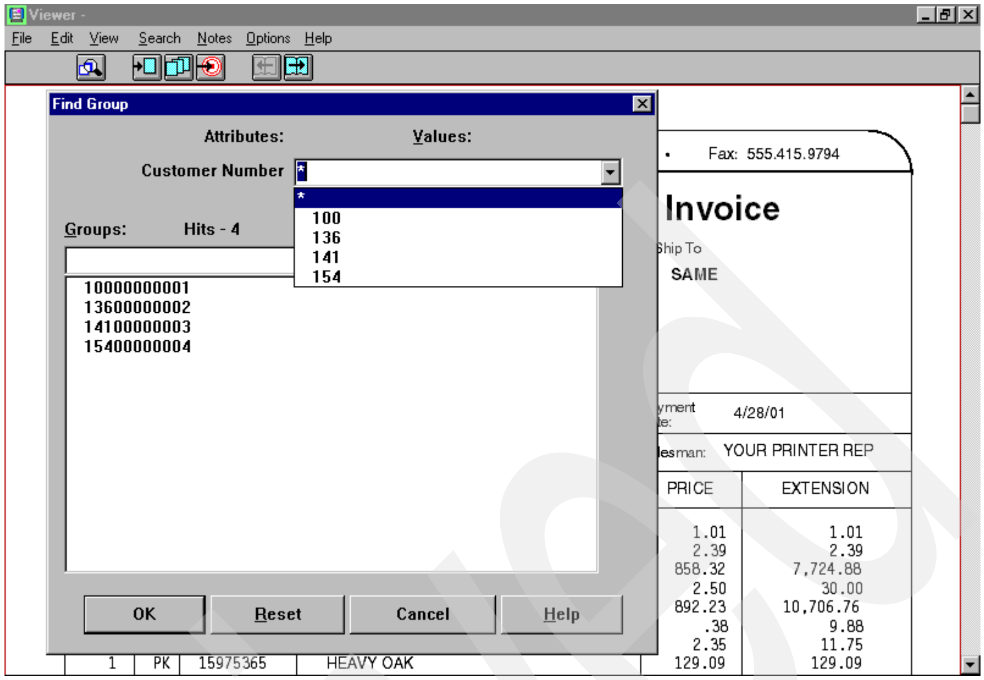


Figure 96. Using the AFP Viewer with an indexed spooled file

Index information can also be used by another component of the Infoprint Server for iSeries product, namely the PDF creation. You can break up a large spooled file into groups of pages to file separately on the IFS, send them as separate e-mail document attachments, or respool them separately as PDF files. CRTAFPDTA is one way to add the necessary information to the spooled file. Chapter 14, "End-to-end example" on page 283, describes this entire process.

5.3.3.1 CRTAFPDTA versus DDS

Alternate methods of adding index information to a spooled file include DDS keywords and the AFP Toolbox. Use the CRTAFPDTA command if you do not have access to the application or if you are using a Page Definition to format line data. Infoprint Designer for iSeries is the key enabler that generates this requirement.

5.3.3.2 Parameters used to index the document

The most important specifications to understand when working with the indexing function of CRTAFPDTA are:

- To index stream file (TOIDXSTMF)
- Index trigger (IDXTRG)
- Index tag fields (IDXTAGFLD)
- Index tag (IDXTAG)
- Index group (IDXGRP)

To index stream file (TOIDXSTMF)

The index operation is performed if you specify a value other than (*NONE) for the TOIDXSTMF parameter. Use this parameter to name the Index stream file or to specify the target directory. For information on the naming convention used in this parameter, see 5.4, "Locating and naming the output of CRTAFPDTA" on page 160. This file contains information based on the index tag fields with pointers to the AFPDS document file.

Group tags are inserted at the beginning of the appropriate pages in the AFPDS document.

Index trigger (IDXTRG)

Assume you have a large spooled file containing multiple client statements, each of which may consist of a different number of pages. These are referred to as a *group*. You need to tell CRTAFPDTA where to look for information that trigger's the beginning of each client's statements or group.

Typical information to use for this are literal strings that print on the first page of the group in predictable locations, such as “Page 1”, “New Balance”, or “Invoice Date”. If the spooled file is generated with carriage control bytes, you could use the fact that there is a “1” in the first column at the beginning of every page. You may specify up to four sets of strings to use as the Index trigger.

To be served as a group trigger, the combination of fields used for the IDXTRG must all appear on the first page of the group. That same combination of fields must not all appear on subsequent pages within the same group. Otherwise, you get a false group. If you use the carriage control byte as a trigger, it must be used with some other information to further define the group. Otherwise, CRTAFPDTA thinks that each page is the beginning of a new group.

The first trigger value that you specify has some special characteristics. It is the only trigger value for which you specify a record-number value of “*”. Each record in the file containing the value specified in the first IDXTRG specification is referred to as an *indexing anchor record*. The record-number in the specifications for any subsequent IDXTRG fields is the relative record number from this indexing anchor point. This must be a *positive* value between 0 and 255.

When CRTAFPDTA is processing the spooled file, it starts by looking in every record for the first trigger value. Once it finds a match, it checks that the remaining IDXTRG fields also match. If they do, then CRTAFPDTA knows it has the beginning of a new group and begins processing the index tag fields. After CRTAFPDTA extracts the necessary information, it begins looking for the first trigger field again.

Index tag fields (IDXTAGFLD)

The index tags are used by the AFP Viewer to further assist in quickly finding the information. Along with the customer number, a service representative may need to access a client's statement by statement number, due date, or postal code. The index tag fields are also used indirectly to generate the group name string that is used by the e-mail exit program.

The first step in creating the index tags is to define the individual fields from which they are to be built. These are the *index tag fields*. The index tag fields may consist of strings of information from the spooled file or literals.

Literal values do not have any added value when only one file is being considered, but in an archive or customer service situation, they may help the user distinguish between information in similar files. For example, you may insert a literal value for “October” to distinguish between records from the “November” file. Literals my also be used in conjunction with “non-traditional” applications that already have some index information inserted, or do not otherwise have information in predictable locations.

When using fields based on data, you specify the record-number, column position, and length. The record-number is the relative record number from the indexing anchor point, as defined in the first index trigger field in IDXTRG. The value for the record-number can be an integer from -255 to +255.

You may specify up to 16 index tag fields. These are numbered automatically in the order they are specified, namely *IDXTAGFLD1 to *IDXTAGFLD16. These names are used for building the index tags, and may appear in error messages to assist with problem determination.

Index tag (IDXTAG)

The index tag parameter does two things. It allows you to group index tag fields in a meaningful way to build the index tags and assigns a literal descriptors to them. For example, assume you want to build an index tag based on the client's phone number and choose to remove the punctuation. You specify the three components of the phone number as three separate index tag fields, and then concatenate them in the specification for the index tag. Assuming the phone number is printed as (xxx) xxx-xxxx, the parameters in the CRTAFPDTA command may look something like this:

```
IDXTAGFLD((1 76 3) (1 81 3) (1 85 4))
IDXTAG((‘Phone Number’ (*IDXTAGFLD1 *IDXTAGFLD2 *IDXTAGFLD3)))
```

You can specify up to eight index tags. These are numbered automatically, *IDXTAG1 to *IDXTAG8, for use in the index group name field, and for reference in error messages.

Index group (IDXGRP)

This parameter specifies which of the index tag values to use to define the group. We recommend that you use the most unique index value for the group name. The default value is *IDXTAG1.

This value is used by the AFP Viewer. This is also the value passed to the user exit program of the PDF generation feature of Infoprint Server.

Index Object (IDXOBJ)

This parameter identifies how much information CRTAFPDTA writes to the index object file. If you specify *GROUP, only group-level entries are placed into the index object file. If you specify *ALL, both page-level and group-level entries are placed into the index.

Select *ALL if your are indexing a file to use with AFP Viewer.

5.3.3.3 Records versus lines

When referring to the relative distance between the index anchor record and other trigger fields or the index tag fields, CRTAFPDTA is looking at the actual records in the spooled file. This is not necessarily the relative position when the information is printed. In the example in Figure 97, it appears as if the combination of “Hello” and “World” may make a good set of trigger fields. The record containing “Hello” could be defined as the trigger anchor record, with the record containing “World” appearing two lines after it.

```
Page 1:
This prints on line 1 of page1. Hello.
This prints on line 2 of page1.
This prints on line 3 of page1. World.

Page 2:
This prints on line 1 of page2. Hello.

This prints on line 3 of page2. World.
```

Figure 97. Sample of printed data

Based on this assumption, you run the CRTAFPDTA command with the following specifications for IDXTRG:

```
IDXTRG((* 34 'Hello') (2 34 'World')
```

After doing so, you discover that not all groups in the spooled file were identified correctly. Further inspection of the spooled file reveals the problem. As illustrated in Figure 98, the word “World” is in the second record after “Hello” on page 1 as expected. But on page 2, it is on the first record immediately after “Hello”. This is because a channel character of “0” was used to double space, instead of having a blank record inserted in the data.

```
Page 1:
1This prints on line 1 of page1. Hello.
 This prints on line 2 of page1.
 This prints on line 3 of page1. World.

Page 2:
1This prints on line 1 of the page. Hello.
0This prints on line 3 of the page. World.
```

Figure 98. Sample records in spooled file

If you are having problems with the index function, you may need to take a closer look at your data. Use the CPYSPLF command and specify CTLCHAR(*FCFC). The target physical file needs to be one byte longer than the spooled file records. If the spooled file is already in *LINE format and references a page definition, you must run the Change Spooled File Attributes (CHGSPLFA) command to set the PAGDFN value to *NONE before running CPYSPLF. Don’t forget to change the PAGDFN value back to the original value before printing.

For most applications, it is likely that the best choices for trigger fields are found near the top of the page, where there is usually less variability in where fields and records are positioned. The example in Chapter 14, “End-to-end example” on page 283, demonstrates how a simple program change can make it easier to define the index trigger fields.

There are additional examples of setting the trigger and index fields in both the *Infoprint Server for iSeries User’s Guide*, G544-5775, and the *AFP Conversion and Indexing Facility: User’s Guide*, S544-5285.

Other parameters

Other parameters used by the index function are:

- IDXCDEPAG
- IDXPAGNAM
- IDXGRPNAM
- IDXPAGLMT

They are described in the *Infoprint Server for iSeries User's Guide*, G544-5775. For more detailed information on these fields, see the corresponding sections on CPGID, CDFPAGENAMES, UNIQUEBNGS, and INDEXSTARTBY in the *AFP Conversion and Indexing Facility: User's Guide*, S544-5285.

5.3.4 Retrieving and packaging AFP resources

When generating a spooled file on the iSeries, the resources are usually external to the file. This is an advantage in that it improves performance and space utilization on the system.

One disadvantage, however, is that if a spooled file is sent to a different system, either another iSeries or another platform, the external resources must be migrated to that other system as a separate operation. They must also be managed in order to keep the two systems in sync with the same resources.

Another concern is change management. If a resource, such as a company logo, overlay, or signature, changes over time, you need to make sure that older documents print with the older versions of these resources, not the new copies. This is important for archive and customer service applications.

5.3.4.1 Parameters used to retrieve AFP resources

The following parameters are used to specify the creation of a resource stream file and name the resource types to be included.

To resource stream file (TORSCSTMF)

The resource retrieval operation is performed if you specify a value other than (*NONE) for the TORSCSTMF parameter. Use this parameter to name the resource stream file or to specify the target directory. For information on the naming convention used in this parameter, see 5.4, "Locating and naming the output of CRTAFPDTA" on page 160. This file contains resources used by the designated spooled file, based on the selection criteria of the RSCDTA parameter.

Resource data (RSCDTA)

This parameter specifies the types of resource data to be written to the resource stream file. The library list of the job is searched for the needed resources.

Applications developed on the iSeries using traditional programming techniques only use *FONT, *FORMDF, *OVL, and *PAGSEG. The *BARCODE, *GOCA, and *IOCA options are included for consistency with other platforms and for users of the AFP Toolbox.

You may notice that the resource type *PAGDFN is not included in the list of available options. Once CRTAFPDTA uses the page definition to generate AFPDS, the page definition itself is no longer of use.

CRTAFPDTA retrieves the designated resource types from the *LIBL, *CURLIB, or a specific library. It also searches the default font libraries for resources, QNDTCPL, QFNT01-QFNT19, and QFNT61-QFNT69.

The resources selected may have been specified in the original application in any number of ways:

- Form definitions specified directly in the CRTAFPDTA command
- Form definitions or overlays specified in the attributes of the spooled file
- DDS keywords referencing overlays or page segments

Resources referenced within other resources are also selected. For example, a form definition may reference an overlay that, in turn, references page segments and fonts.

If you are using the CRTAFPDTA to prepare a document to use the AFP Viewer, you do not need the capture the fonts in the resource file. This is because the AFP Viewer maps the requests for AFP fonts to fonts resident on the PC.

5.3.5 Creating a merged file

Depending on the needs of the application, it is likely that you need to concatenate or merge the resource file or index file with the AFPDS data. For example, when sending the spooled file to another system, you need the resources to be sent at the same time in one package. Similarly, the index information needs to be included when using the AFP Viewer.

5.3.5.1 Parameter for creating a merge file

There is only one parameter to specify to create a merge file.

To Merge Stream File (TOMRGSTMF)

The merge operation is performed if you specify a value other than (*NONE) for the TOMRGSTMF parameter. Use this parameter to name the merge stream file or to specify the target directory. For information on the naming convention used in this parameter, see 5.4, “Locating and naming the output of CRTAFPDTA” on page 160. This file contains information based on the index tag fields with pointers to the AFP

5.4 Locating and naming the output of CRTAFPDTA

The CRTAFPDTA command generates from one to four files on the iSeries IFS. There are a number of ways you can specify the names and locations of these files.

Data in the traditional file system is referenced by library, physical file name, and member. This same data is accessible via the IFS under the QSYS.LIB directory. For example, the member MYMBR in the physical file MYFILE in library MYLIBRARY can be accessed via the IFS directory naming convention as:

```
/QSYS.LIB/MYLIBRRARY.LIB/MYFILE.FILE/MYMBR.MBR
```

If you want to use the output from CRTAFPDTA in a process that is expecting a physical file member, you should direct the output to a directory in the QSYS.LIB file system. For example, if you need to re-spool the output using the PRTAFPDTA command, you must reference the data as a physical file member.

One easy way to do this is to change the current directory to one in the QSYS.LiB file system before running CRTAFPDTA. Use the Change Current Directory (CHGCURDIR) command:

```
CHGCURDIR '/QSYS.LIB/MYLIB.LIB/MYFILE.FILE'
```

If you are going to use the data in an application that can access the IFS files using the directory structure, you may specify any directory that suits the needs of the target application. For example, an application that posts the files to an intranet site may use the IFS naming convention.

TOxxxSTMF (*DFT)

For any of the four stream files generated, you may specify a value of *DFT. This places the output in the current working directory. If the current working directory is in the QSYS.LIB file system, the resulting files are named *outputafp.mbr*, *outputidx.mbr*, *outputrsc.mbr*, and *outputmrg.mbr*.

> **Note**
>
> When referencing the members using traditional file/member methods, do not include the ".mbr".

If the current directory is not in the QSYS.LIB file system, the resulting files are named *output.afp*, *output.idx*, *output.rsc*, and *output.mrg*.

TOxxxSTMF (to-xxx-stream-file-path)

If you specify a name for the target stream file path, the following rules apply:

- If you specify a file name without a path, the output file is placed in the current working directory.
- If you specify a directory or path without a file name, a file with the default name of *outputxxx.mbr* or *output.xxx* is created in the specified directory.
- All directories in the path must exist.
- If the path name is qualified, it must be enclosed in quotation marks.
- The path name cannot contain a pattern.
- If the file exists, it is overwritten.

5.5 Using the AFP Viewer with the output of CRTAFPDTA

The AFP Viewer is available as part of a variety of products. Choose the most appropriate implementation that suits your source for this tool and where the data resides. For more information on the AFP Viewer, see the description in 1.4.4, "AFP Viewer" on page 28.

5.5.1 Using the AFP Viewer via Operations Navigator

The AFP Viewer is included as part of Client Access Express and is easy to access via the Operations Navigator. In this mode of use, it expects the data to be in the form of an iSeries spooled file. One advantage of using the AFP Viewer in this manner is that external resources are automatically downloaded to the workstation.

A disadvantage of using the AFP Viewer in this manner is that the output of CRTAFPDTA needs to be re-spooled using PRTAFPDTA.

5.5.2 Viewing an AFP stream file

Although the AFP Viewer is shipped as part of Client Access Express, it can be run as a standalone application on the workstation. Chapter 8 of the *Infoprint Server for iSeries User's Guide*, G544-5775, describes how to set up this environment.

Users may prefer this method because it avoids the extra processing step required for PRTAFPDTA. The output files from CRTAFPDTA may be accessed directly from the IFS. For additional performance improvements, copy the file from the IFS to the hard drive on the workstation.

If the output from CRTAFPDTA is to be used in this manner, all the external resources except fonts need to be selected and merged with the AFP file.

5.5.3 Using the IBM AFP Workbench for Windows V2

You may choose to use the standalone product, IBM AFP Workbench for Windows (5622-416), if the user who wants to view AFP files is not otherwise licensed for Client Access. The AFP data could reside on the iSeries IFS or be copied to the user's workstation. All external resources except for fonts must be included.

5.5.4 Using the AFP Viewer plug-in with a Web browser

Most users are familiar with the use of the Adobe plug-in for viewing PDF files from a Web browser. There is a similar plug-in available from IBM to view AFP files. In a distributed environment, a Web serving application could be written that accesses the output from CRTAFPDTA from the IFS and delivers it to a user via their browser.

The AFP Viewer has the advantage over a PDF viewer in that AFP files tend to be much more compact than the equivalent PDF document. PDF documents generated by Infoprint Server for iSeries do not contain index information used for navigation within the document the way the AFP documents do.

You can download the AFP Viewer plug-in "as-is" for free from:
`http://www.ibm.com/printers/R5PSC.NSF/Web/afp_viewer_plug-in`

5.6 Command comparison with OS/390 ACIF

CRTAFPDTA was migrated from the OS/390 program AFP Conversion and Indexing Facility, which is commonly called ACIF. The overall function is similar, but there are some differences in implementation:

- CRTAFPDTA does not provide user exit points.
- ACIF does not automatically merge the resulting spooled files.
- With ACIF, you name specific libraries for each resource type. With CRTAFPDTA, the iSeries library lists are used.

- Some parameters, such as CTLCHAR or TBLREFCHAR, are picked up from the iSeries spooled file attribute. With ACIF, these must be entered as part of the command.

Table 19 lists the CRTAFPDTA parameters and the corresponding ACIF keywords.

Table 19. CRTAFPDTA-ACIF cross reference

CRTAFPDTA	ACIF
AFPCHARS	CHARS
CTLCHAR from spooled file attributes	CC and CCTYPE
FORMDF	FORMDEF, FDEFLIB
FROMSPLF, JOB, SPLNBR	INPUTDD
IDXCDEPAG	CPGID
IDXGRP	GROUPNAME
IDXGRPNAM	UNIQUEBNGS
IDXOBJ	INDEXOBJ
IDXPAGLMT	INDEXSTARTBY
IDXPAGNAM	DCFPAGENAMES
IDXTAG	INDEXn
IDXTAGFLD	FIELDn
IDXTRG	TRIGGERn
IGCSOSI from spooled file attributes	PRMODE
IMAGEOUT	IMAGEOUT
PAGDFN	PAGEDEF, PDEFLIB
RSCDTA	RESTYPE
TBLREFCHAR from spooled file attributes	TRC
TOIDXSTMF	INDEXDD
TOMRGSTMF	Not supported
TORSCSTMF	RESOBJDD
TOSTMF	OUTPUTDD

Table 20 lists ACIF keywords that do not have specific CRTAFPDTA equivalent parameters.

Table 20. ACIF parameters not used directly in CRTAFPDTA

ACIF	CRTAFPDTA
CC	CTLCHAR from spooled file attributes
CCTYPE	CTLCHAR from spooled file attributes
COMSETUP	Not supported

ACIF	CRTAFPDTA
FDEFLIB	Library portion of FORMDF
FONTECH	Not supported
FONTLIB	Not supported. *LIBL of splf creator used.
INDEXEXIT	Not supported
INPEXIT	Not supported
MCF2REF	Not exposed. Hard-coded to value of CF.
OBJCONLIB	Not supported
OUTEXIT	Not supported
OVLYLIB	Not supported. *LIBL of splf creator used.
PDEFLIB	PAGDFN Library portion of PAGDFN
PRMODE	IGCSOSI from spooled file attributes
PSEGLIB	Not supported. *LIBL of splf creator used.
RESEXIT	Not supported
RESFILE	Not supported
TRACEDD	Not supported. Use OS/400 job trace facility.
TRC	TBLREFCHAR from spooled file attributes
UNIQUEBNGS	IDXGRPNAM
USERLIB	Not supported

For more information on the ACIF keywords, refer to the *AFP Conversion and Indexing Facility: User's Guide*, S544-5275.

Chapter 6. Image file transform

This chapter introduces the new image converter part of Infoprint Server for iSeries. The three new image transforms are Windows based and may be used to convert objects from PC-based applications in AFP format or documents.

Infoprint Server image converter can convert GIF, JPEG, and TIFF images to AFP and PostScript objects.

6.1 Converting PC images to AFP objects

Since image application is so popular, the need to convert images in a different format becomes more important. In the past customers were able to use the AFP print driver and or the AFP Workbench viewer for this need. The new transforms may be called from a command in under Windows or from a Windows application.

Many customers who use PC-based image software and hardware want to create, print, and use these resources on the iSeries. Infoprint Server 400 provides PC-based conversion capabilities for the most common images formats.

The transforms may be used as an alternative way to produce AFP resources or AFP documents. Each converter may be invoked manually from a command prompt or from a program. The new transforms are a valuable alternative to the AFP print driver. They can be integrated in a PC-based application, such as images or scanning software.

Figure 99 shows an overview of the new Infoprint Server image transform.

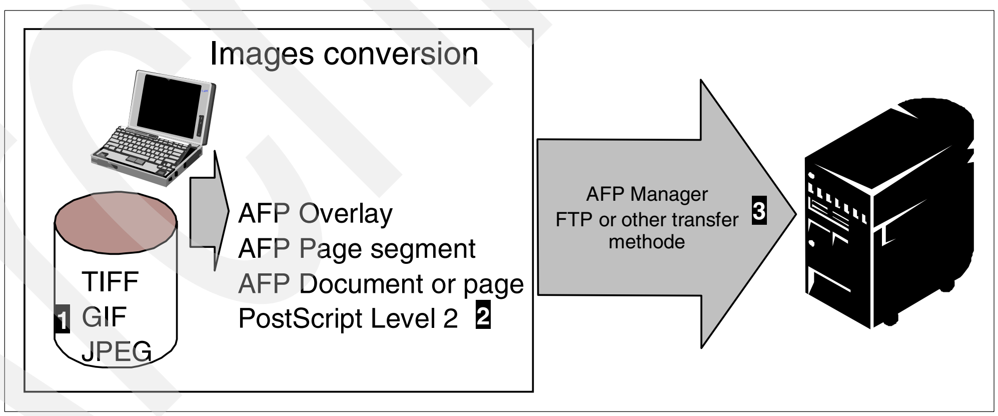


Figure 99. Images conversion flow

The process flow in Figure 99 is explained here:

1. PC-based applications or programs are used to create images.
2. The Infoprint Server image converter may be invoked, manually or in batch. The converted image may be placed on a disk for a PC or on an iSeries shared folder.
3. The output objects can be used on any system that supports AFP. The AFP Manager may be used to place AFP objects on the iSeries as explained in 6.3, “Converting TIFF, GIF, and JPEG images to AFP objects” on page 168. Other

transfer methods like FTP or Client Access/400 may be used to place the AFP objects on the iSeries server.

Table 21 shows the different capabilities supported by the image converter.

Table 21. Image conversion format

PC image object	Supported AFP object type
JPEG[1]	Document (or page), overlay, page segment
TIFF[2]	Document (or page), overlay, page segment
GIF[3]	Document (or page), overlay, page segment

1. Only Huffman coding with Baseline DCT JPEG algorithm is supported. At most, 8-bit grayscale and 24-bit color are supported.
2. Only Huffman coding with Baseline DCT JPEG algorithm is supported. Old style JPEG specification (Photometric Interpretation 6) is not supported. Use the Photometric Interpretation 7 to incorporate the JPEG compressed images in TIFF files. Additional color information, such as alpha data, is disregarded. Transparency maps are ignored. At most 8-bit grayscale and 24-bit color are supported.
3. Timing data from GIF animation sequences is ignored. Animation sequences are processed as unrelated images.

6.1.1 Other image transforms available on OS/400

Some image transform capabilities have been available on OS/400 in the past. Table 22 shows these transform and conversion capabilities. Note that these transforms are OS/400 based.

Table 22. Image transforms from OS/400 V4R2

Input data streams	Output data streams
TIFF GIF BMP PostScript Level 1*	AFPDS PCL

* A new inbound PostScript transform is available with Infoprint Server. Chapter 7, “Converting PCL, PostScript, or PDF to AFP” on page 173, provides additional information on this topic.

You can find additional information about the previous image transform in *AS/400 Printing V*, SG24-2160.

6.2 Image transform setup

To use the Windows-based transform, you must install them on your PC by following these steps:

1. Right-click **Network Neighborhood** and select **Find Computer...** Specify the name of your OS/400 in the Named: field.
2. When Windows finds your OS/400, double-click your system to display the contents.
3. Expand **QIB-> ProdData-> InfoprintServer-> Transforms-> Install-> Image**.
4. Double-click **setup.exe** as shown in Figure 100 and follow the instructions provided in the installation shield.

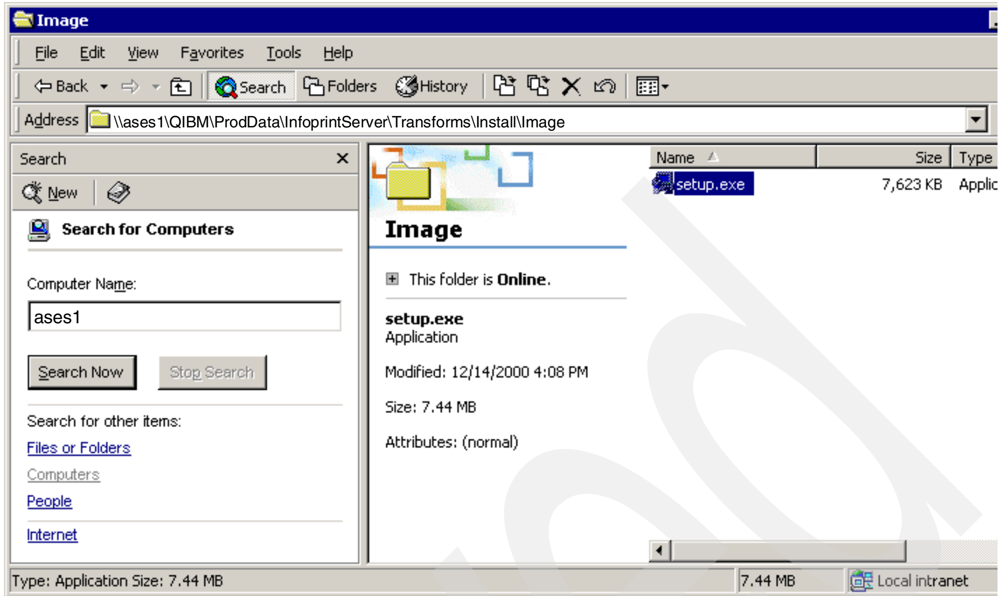


Figure 100. Install Image transform

Verify whether the program is installed on your PC. Use Windows Explorer and search for gif2afp.exe, tiff2afp.exe, and jpeg2afp.exe.

Continue with 6.2, "Image transform setup" on page 166, before you start to convert images.

6.2.1 Setting the configuration

Before you can use one of the transform, you must understand and set up the environment. All three transforms, jpeg2afp, gif2afp, and tiff2afp, may use default parameters or environment variables. The parameter values are determined by the following hierarchy (least significant first):

- Internal defaults
- Environment variables
- Configuration file
- Command-line arguments

Using the environment variables and configuration file setup is optional.

6.2.1.1 Using environment variables

You can use environment variables to specify parameters for the GIF to AFP transform. Environment variables have the form:

```
Variable: GIF2AFP_option, Value: value
```

Here *option* and *value* are the same as the command line and configuration file option and value. If the option does not have a value, specify a blank surrounded by single quotation marks (' '). You cannot use -calib as an environment variable.

To edit your Windows environment variables, follow these steps:

1. Right-click **My Computer** and select **Properties**.
2. Select the **Environment** page.

3. Specify the parameters you want by repeating the following steps as many times as necessary:
4. Complete the Variable and Value for the parameter you want to use:
   a. Click **Set**.
   b. Click **Apply**.
5. Restart your computer for the changes to take effect.

6.2.1.2 Transform configuration file

All transforms have a different configuration file that can be modified to use the parameter as a default. Not all parameters are eligible. The example in Figure 101 shows a typical configuration used to create APF page segments.

```
a=ioca10
cmp=g4
respath=.;<Install Path>\config
ms=5120
msf=0.01
term=yes
v=yes
pagetype=page
r=600
x=0
y=0
```

Figure 101. Typical configuration for creating AFP page segments

The configuration file may reduce the number of parameters you need to set in the command. The parameter used on the command line overrides the parameter set in the configuration file.

6.3 Converting TIFF, GIF, and JPEG images to AFP objects

We have chosen to create a page segment for this example, but any different supported type maybe used. If the wrong object type is selected, no conversion will be provided and no object will be created. Consider this example:

```
jpeg2afp.exe myfile.jpg -o myfile.afp -r300 -scale100% -alg afp -pagetype pseg
tiff2afp.exe myfile.tif -o myfile.afp -r300 -scale100% -alg afp -a ioca
-pagetype pseg
```

This example uses the following attributes:

-o	The name of the converted file resulting of the transformation
-r300	The resolution of the output file, in this case 300 dpi
-scale100%	Allows you to rescale the image inside of the defined size
-alg	Uses AFP halftoning algorithms for default printer
-a	Image format is IOCA 10
-pagetype	In this case, produces a page segment (pseg)

Many transform options are available and are listed in 6.4, “Image transform options” on page 171. You can find detailed information about the attributes in the *Infoprint Server for iSeries User’s Guide*, G544-5775.

Figure 102 shows the AFP output file using the AFP Workbench viewer. The AFP Workbench viewer is available with Client Access/400.

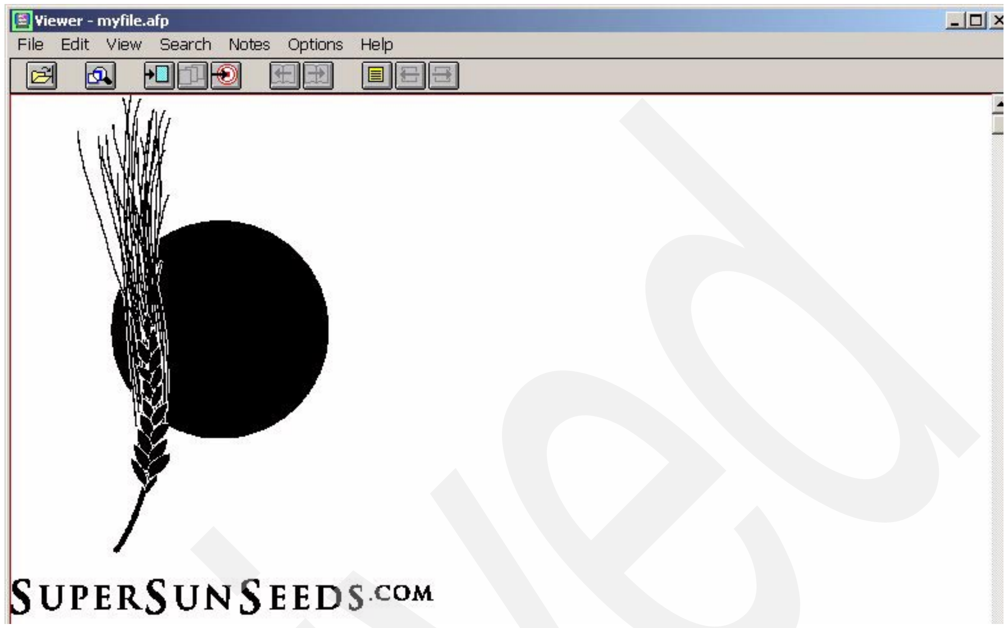


Figure 102. AFP resource output file

6.3.1 Uploading your AFP resources with AFP Manager

You have to upload the AFP resources if you want use it on the iSeries. Depending on the resource type you defined in the conversion, different transfer methods are possible, manually or using the new AFP Manager available with Client Access/400.

AFP Manager is used in this example. Start AFP Manager in Operations Navigator as shown in Figure 103.

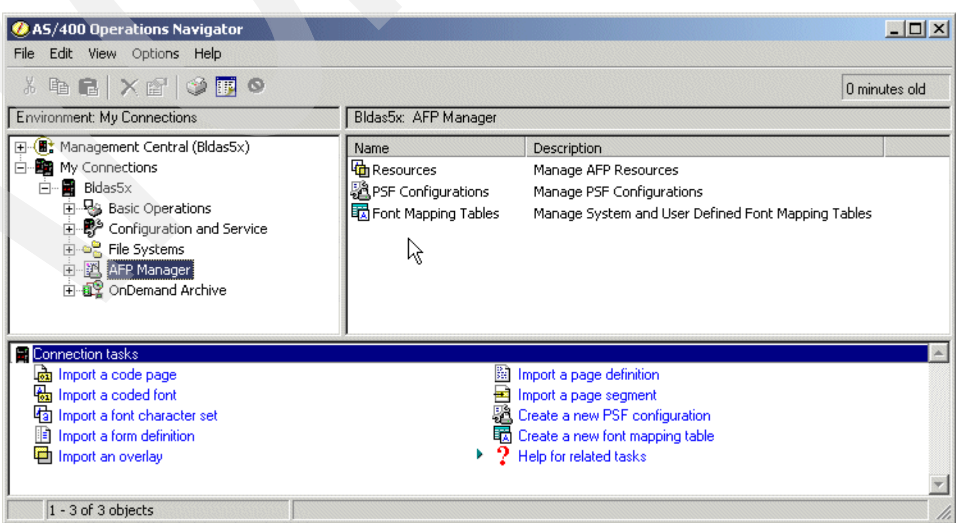


Figure 103. AFP Manager

The AFP Manager import function provides an upload function for AFP resources. Choose the corresponding import function for your resource. If your PC file does not match the requested format, no resource is created.

AFP Manager invokes the native command to upload the resource and create the iSeries object. The temporary files created for this process are deleted at the end of each upload request.

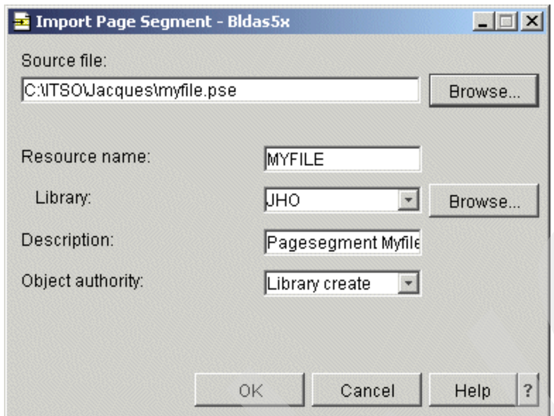


Figure 104. Import Page Segment

You complete the fields in the Import Page Segment window as explained here:

- **Source file**: Specify the name of the file you want to import. Click **Browse** to search for the file. The file must be in a supported format and correspond to the selected AFP resource.
- **Resource name**: Specify a name for the resource you want to create. The name cannot be more than eight characters long and must meet iSeries name requirements.
- **Library**: Specify the library where the new resource will be located. You can store it in the current library or a different library. To specify a different library, enter the name or click **Browse** to search for a library. The library you specify must exist.
- **Description**: Enter a description of the resource you want to create. This parameter is recommended, but not required. The description cannot contain more than 50 characters.
- **Object authority** [Library create]: When the library containing this object was created on the server using the Create Library (CRTLIB) command, a value was specified on the Create authority prompt (CRTAUT parameter). When you select Library create as the object authority, PSF/400 uses that value to determine the authority for this object. If the value specified on the Create authority prompt is changed, the new value does not affect any existing objects.

6.4 Image transform options

Many options are available for each different image transform. One or several options may be passed to the transform as an attribute. Most of the options are similar to those used on different image utilities. Table 23 outlines the options.

Table 23. Image transform options comparison

Functions	GIF to AFP	JPEG to AFP	TIFF to AFP
[]**-a** *ImageType*	√	√	√
[]**-alg** *ProcessingAlgorithms*	√	√	√
[-calib *calibration*]	√	√	√
[-choice *ImageChoice*]			√
[-C *ConfigurationFile*]	√	√	√
[-clean *cleanup*]	√		√
[-cmp *compression*]	√		√
[-crop *CropFactors*]	√	√	√
[-fit {trim | **scale**}]	√	√	√
[-force]		√	√
[-gcorr *GrayscaleMappingTable*]	√	√	√
[-ink *color*]	√		√
[-inv]	√		√
[-j *ScanOffsetFileName*]	√	√	√
[-l *ImageLength*]	√	√	√
[-M *MemoryBound*]	√	√	√
[-mp | -nomp]			√
[-ms *space*]	√	√	√
[-msf *SpaceFraction*]	√	√	√
[-o *OutputFile*]	√	√	√
[-outbits *NumberOfOutputBits*]	√	√	√
[-outcolor *OutputColorModel*]	√	√	√
[-p *PageRange*]	√	√	√
[-pagetype *PageType*]	√	√	√
[-paper *PaperSize*]	√	√	√
[-r *resolution*]	√	√	√
[-respath *ResourceSearchPath*]	√	√	√
[-rot *rotation*]	√	√	√
[-scale *ImageSize*]	√	√	√

Functions	GIF to AFP	JPEG to AFP	TIFF to AFP
[-sgcorr *ScannerCorrection*]	√	√	√
[-sniff I **-nosniff**]	√		√
[-term I **-noterm**]	√	√	√
[-thresh *HalftoneFile*]	√	√	√
[-v I **-nov**]	√	√	√
[-w *ImageWidth*]	√	√	√
[-wrkdir *WorkDirectory*]		√	√
[-x *LeftMargin*]	√	√	√
[-y *TopMargin*]	√	√	√
[-z] file	√	√	√

Chapter 7. Converting PCL, PostScript, or PDF to AFP

This chapter explains how to take advantage of the additional Infoprint Server conversion capabilities. PCL5e, PostScript, and PDF format may be converted to a full page image-based AFP data stream. The resulting AFP data stream may be printed on IPDS printers attached to the iSeries.

Typically customers who have LAN-based applications may use the PCL5e, PostScript, and PDF transform to take advantage of the IPDS printers and iSeries spool capabilities. This may reduce the number of servers that drive the same printer and enhance spooled file handling and security. This chapter includes the following topics:

- PCL and PDF/PostScript to AFP introduction
- Define the transform environment
- Sending PCL, PDF, or PostScript data to the iSeries

Table 24 shows the transforms that are part of the Infoprint Server product.

Table 24. PCL, PDF, and PostScript transform capabilities

Datastream conversion to	AFP	PDF
PCL 5e	Infoprint Server[1]	Infoprint Server[1, 2]
PostScript Level 3	Infoprint Server[1]	Infoprint Server[1, 2]
PDF	Infoprint Server[1]	n/a
1.Not text composed AFP or PDF formats, only full image. 2.Transformation from PCL or PostScript to PDF requires a double conversion. PCL and PostScript must be converted to AFP first; no direct conversion is provided.		

The following section describes the conversion process.

7.1 PCL and PDF/PostScript to AFP

PostScript Level 1 to AFP transform has been a part of image transform since V4R1 of OS/400. Infoprint Server for iSeries provides new transforms that support PCL5e and PDF/PostScript Levels 1, 2, and 3 to AFP. The new transforms are part of the Infoprint Server for iSeries. Refer to Chapter 1, “iSeries printing software to create, manage output” on page 1, for more information about software components.

The different transforms are integrated as a service required by an AFP print writer. The transform type is selected with the image transform parameter of the printer description. As soon as a spooled file is detected in the corresponding output queue, the transform process is invoked.

Figure 105 shows the transform process and its different components.

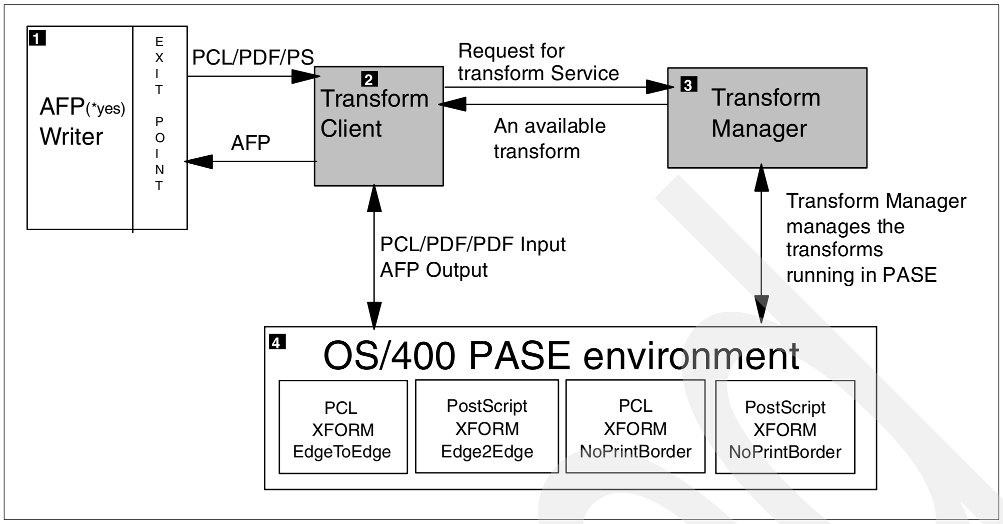


Figure 105. PCL, PDF/PostScript transform process

The different transforms are integrated as a service required by an AFP print writer. The transform type is selected with the image transform parameter of the printer description. As soon as a spooled file is detected in the corresponding output queue, the transform process is invoked. The following components are part of the process:

- The image transform has to be configured in the device description. Different image transform objects are selectable depending of the need. Section 7.2.3, “Creating a device description” on page 180, provides detailed information about the different objects.
- The image transform exit point is used to invoke the transform client. A transform client passes the request to the Transform Manager and routes the input and output data to and from the appropriate transform. Each print writer has its own associated transform client.
- A unique Transform Manager manages the different transforms regarding the request passed through the transform client and the number of transforms that are available. The number of available transforms may be defined in the transform configuration as shown in 7.2.2, “Setting up the configuration files” on page 177.
- Two different transform are available: a PCL to AFP and a PDF/PostScipt to AFP transform. Two different transform classes are available for each transform: edge-to-edge and no print border. All transforms run in the OS/400 PASE environment.

Figure 106 shows the new transforms in a more general iSeries environment. Parameters of the gray shaded objects provide specific information about the transform process. They are described in 7.2, “Defining the PCL, PostScript, and PDF transform environment” on page 176. This figure also the process flow of a PCL, PostScript, and PDF transform environment.

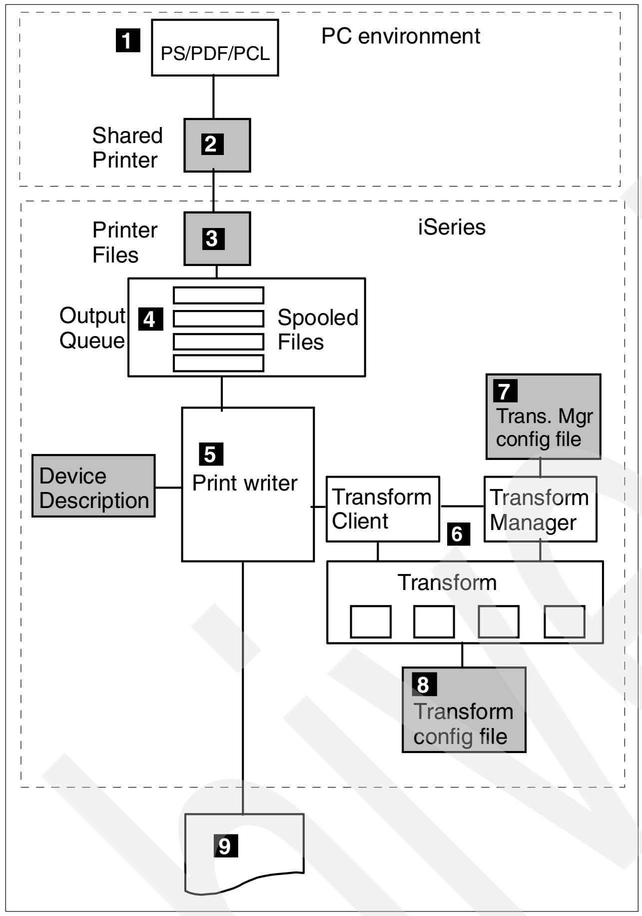


Figure 106. PCL, PostScript, and PDF transform environment

The process flow is outlined here:

1. A Windows-based application submits a PostScript, PDF, or PCL print job to a shared printer. These are available as any other printer through the Windows printer menu.
2. A Windows printer definition, as explained in 7.3.1, “Sharing the printer with NetServer” on page 188, or 7.3.3, “Printing your file with LPD” on page 194, must be used to submit the print job to the iSeries
3. A printer file is associated with each shared printer. Specific printer file attributes are used to define some of the print job options. Note that a unique printer file is associated to the Windows printer definition. See 7.2.4, “Creating a printer file” on page 182, for additional information.
4. The spooled files are placed in the iSeries output queue. The spooled files are in PCL5e, PostScript, or PDF formats. See the following note regarding drivers.
5. As soon as a valid spooled file is ready, the writer notifies the transform client and starts the conversion. The transform process is described in 7.1, “PCL and PDF/PostScript to AFP” on page 173.
6. After the transform process is invoked, the Transform Manager sniffs the data type and invokes the corresponding transform.

7. The transform configuration type determines how many transforms are available. See 7.2.2, “Setting up the configuration files” on page 177, for more information.
8. A configuration file is associated with each transform type. The configuration file parameters are used as the default setting for the transform. See 7.2.2.2, “Setting up the transform configuration file” on page 180.
9. The quality of the transform depends on the correct parameters set in the different descriptions and files. Section 7.2, “Defining the PCL, PostScript, and PDF transform environment” on page 176, discusses these parameters.

Supported Windows drivers

The Infoprint Server Transform Manager supports PostScript Level 3 and PCL5e drivers. There is no specific driver to recommend in this case. In the test case for this redbook, we used the Infoprint 40 PostScript driver. Some drivers use hardware specific features that are not supported by any transforms.

7.2 Defining the PCL, PostScript, and PDF transform environment

Infoprint Server enhances the conversion capabilities from Image Print Transform.

Imagine for a moment that you want to send your print requests directly as a print job from your Windows workstation to an iSeries attached printer. This section describes what has to be configured to use one of the transforms. The following iSeries objects have to be created or modified to use the transform. These objects are:

- Create the work directory
- Setup the Transform Manager configuration file
- Setup the transform configuration files
- Create a printer device description
- Create a printer file
- Start and stop the Transform Manager
- Transform Manager Exit Point
- Fonts

7.2.1 Creating the work directory

You must create the work directory with the proper authorities in order for Transform Manager to work properly. Create this path on your system with the specified owner and permissions as listed in Table 25:

```
/QOpenSys/QIBM/UserData/InfoprintServer/Transforms/work
```

Table 25. Directories to create for Transform Manager

Directory	Owner	Permissions
Infoprint Server	QSYS	755
Transforms	QSYS	755
work	QSPLJOB	700

To change the directory information, follow these steps:

1. From the command line, call:

   ```
   qp2term
   ```

2. Specify `cd` to change to the directory that contains the directory you want to edit. For example, the following command allows you to edit the Infoprint Server directory:

   ```
   cd /QOpenSys/QIBM/UserData/
   ```

3. Change the directory owner by specifying:

   ```
   chown new_owner directory
   ```

   For example, the following statement makes QSYS the owner of Infoprint Server:

   ```
   chown QSYS InfoprintServer
   ```

4. Change the directory permissions by specifying:

   ```
   chmod new_permissions directory
   ```

   For example, the following example makes 755 the number of permissions for InfoprintServer:

   ```
   chmod 755 InfoprintServer
   ```

Note

There cannot be anything in the work directory.

7.2.2 Setting up the configuration files

This section explain the relationship between the configuration files of the transform and Transform Manager. Two configuration files are available:

- **Transform Manager configuration file**: Allows the customer to tailor the number and type of transforms that will be available based on the estimated transform needs.
- **Transform configuration file:** Allows you to add specific functions available for the different transforms. These functions are mostly common to the transforms available with Infoprint Manager. Chapter 8, “PostScript to AFP using a different input paper tray” on page 203, provides an example of the specific function.

Before you can start with this section, you must read the transform process described in 7.1, “PCL and PDF/PostScript to AFP” on page 173.

Figure 107 shows the relationship between the different components.

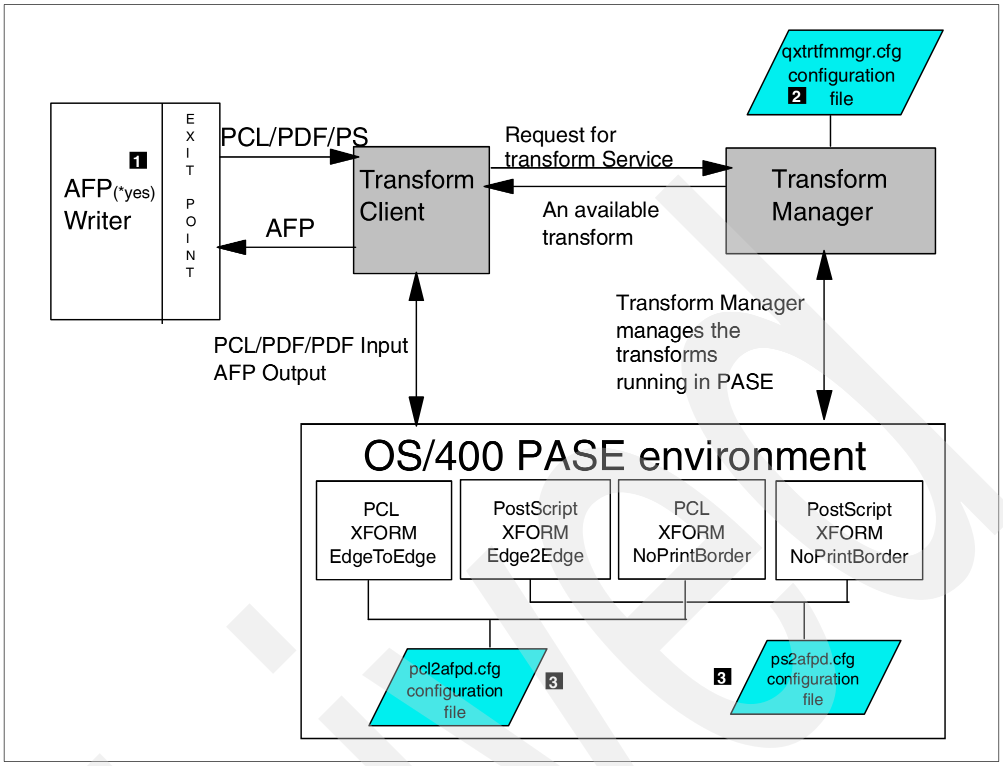


Figure 107. Transforming configuration files

The process flow shown in Figure 107 is explained here:

1. The device description must be configured as AFP(*YES), and the image transform object must be selected. Section 7.2.3, “Creating a device description” on page 180, provides detailed information about the different objects.
2. In addition to what we mentioned in the introduction of this chapter, the transform configuration file is used to define the number of transforms running.
3. Each transform type PCL or PostScript/PDF has a different configuration file.

7.2.2.1 The Transform Manager configuration file

The Transform Manager configuration file, qxtrtfmmgr.cfg, allows the customer to tailor the type of transforms that are available base on the data types that will be printed. It also allows the customer to determine the number of each type of transform to initialize when the Transform Manager is started.

Two different transforms are available:

- **PostScript to AFP Transform (PS2AFP)**: This transform is used to convert PostScript and PDF data to AFP.
- **PCL to AFP Transform (PCL2AFP)**: This transform is used to convert PCL5 data to AFP.

Two classes of image transform are available for the PS2AFP and for the PCL2AFP Transform:

- **EdgeToEdge**: For use with printers that support edge-to-edge printing.
- **NoPrintBorder** (.167 inch margin): For use with printers that have unprintable regions. All output produced by a transform of this class has a .167 inch margin on all four sides.

Refer to your printer documentation and printer configuration for additional information about EdgeToEdge printing.

Use the following steps to help you to create a new configuration file for one of the transforms. Before you start, be sure to make a copy of the configuration file as backup.

1. Use the CPY command or Operations Navigator to copy this file from the ProdData path to the UserData path. Using Windows or a qp2term session to copy the file alters the CCSID of the file and causes Transform Manager to fail to start.
2. Make a copy of the IBM-provided configuration file. To do this, copy the IBM supplied configuration file qxtrtfmmgr.cfg in /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ to /QOpenSys/QIBM/UserData/InfoprintServer/Transforms/.
3. Change the values for the minimum active and maximum active keywords for each transform type. *Do not* remove any lines from the file. If no transform of a given type is needed, set the minimum active value to 0 and leave the maximum active value as 1.
   - **The IBM-supplied default:** /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms
   - **The customer modified version**: /QOpenSys/QIBM/UserData/InfoprintServer/Transforms

CCSID for the configuration file

Use the CPY command or Operations Navigator as described in step 1. Using Windows Explorer or qp2term changes the CCSID of the file. The CCSID of the file must be 037.

The following configuration file shows how to start one transform of each type:

```
PS Transforms:
image config type = EdgeToEdge
minimum active = 1
maximum active = 1

PCL Transforms:
image config type = EdgeToEdge
minimum active = 1
maximum active = 1

PS Transforms:
image config type = NoPrintBorder
minimum active = 1
maximum active = 1
```

```
PCL Transforms:
image config type = NoPrintBorder
minimum active = 1
maximum active = 1
```

Consider this example. One PCL transform starts and a second one may be started automatically if needed. If three jobs are waiting in the queue, the third one will have to wait until one of the transforms is finished:

```
PCL Transforms:
image config type = NoPrintBorder
minimum active = 1   <----- change this line; values 0 to 32767 are valid
maximum active = 2   <----- change this line; values 1 to 32767 are valid
```

7.2.2.2 Setting up the transform configuration file

The PS2AFP and PCL2AFP transforms use configuration files with default parameters. This file may be modified to take advantage of specific functions. If incorrect parameters are specified, the transform startup fails.

You must first edit the PS2AFP.cfg configuration file. The following example explains how to modify the configuration file on the iSeries server. The configuration files for PCL and PostScript transforms are located in:

- /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp
- /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/pcl2afp

Add any new statement at the end of the configuration file. The file may be edited using any standard PC-based editor (remember to make a copy of the original configuration file before you start).

After you place this new line in the configuration file, you should restart Transform Manager. The command is effective after the new start. The system environment may overwrite the configuration file at each new start.

7.2.3 Creating a device description

You need to create or modify the device description for your printer. This example shows how to create a device description for an IPDS printer. Figure 108 shows how to create an IPDS LAN attached device description.

```
                  Create Device Desc (Printer) (CRTDEVPRT)

Type choices, press Enter.

Device description . . . . . . . > IP20RED   Name
Device class . . . . . . . . . . > *LAN      *LCL, *RMT, *VRT, *SNPT, *LAN
Device type  . . . . . . . . . . > *IPDS     3287, 3812, 4019, 4201...
Device model . . . . . . . . . . > 0          0, 1, 2, 3, 4, 10, 13, 301...
LAN attachment . . . . . . . . . > *IP       *LEXLINK, *IP, *USRDFN
Advanced function printing . . . > *YES      *NO, *YES
Port number  . . . . . . . . . . > 5001      0-65535
Online at IPL  . . . . . . . . .   *YES      *YES, *NO
Font:
  Identifier . . . . . . . . . . > 011       3, 5, 11, 12, 13, 18, 19...
  Point size . . . . . . . . . .   *NONE     000.1-999.9, *NONE
Form feed  . . . . . . . . . . .   *TYPE     *TYPE, *CONT, *CONT2, *CUT...
Separator drawer . . . . . . . .   *FILE     1-255, *FILE
Separator program  . . . . . . .   *NONE     Name, *NONE
  Library  . . . . . . . . . . .             Name, *LIBL, *CURLIB
Printer error message  . . . . .   *INQ      *INQ, *INFO
                                                                    More..
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 108. Create Device Description display

For this example, the following parameters are used:

- **Name**: The name you want to use for this device.
- **Device class**: This parameter is used to define how the device is attached.
- **Device type**: Define the device type, in this example, IPDS.
- **Device model**: Use 0 for any IPDS attached printer.
- **LAN attachment**: Define the attachment type.
- **Port number**: The default port for an IBM IPDS printer is 5001. This may vary if printers from other manufacturers are used. Please refer to the printer documentation in this case.
- **Advanced function printing**: Must be set to `*YES`.

Figure 109 shows the second screen of the Create Device Printer command. The Image configuration object parameter must be defined to use the transform.

```
                   Create Device Desc (Printer) (CRTDEVPRT)

Type choices, press Enter.

Message queue  . . . . . . . . .   *CTLD         Name, *CTLD, *SYSOPR, QSYSOPR
  Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
Activation timer . . . . . . . .   170           1-2550, *NOMAX
Image configuration  . . . . . . > *IMGC06        *NONE, *IMGA01, *IMGA02...
Maximum pending requests . . . .   6             1-31
Print while converting . . . . .   *YES          *NO, *YES
Print request timer  . . . . . .   *NOMAX        1-3600, *NOMAX
Form definition  . . . . . . . .   F1C10110      Name
  Library  . . . . . . . . . . .     *LIBL       Name, *LIBL, *CURLIB
Remote location:
  Name or address  . . . . . . . > '9.99.177.225'



User-defined options . . . . . .   *NONE         Character value, *NONE
               + for more values
                                                                       More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 109. Setting the image configuration object

The following different formatting options may be defined with IBM IPDS printers:

***IMGC01** IPDS 240-dpi printer
***IMGC02** IPDS 300-dpi printer
***IMGC03** IPDS 600-dpi printer
***IMGC04** IPDS 1200-dpi printer
***IMGC05** IPDS 240-dpi printer with no-print border
***IMGC06** IPDS 300-dpi printer with no-print border
***IMGC07** IPDS 600-dpi printer with no-print border
***IMGC08** IPDS 1200-dpi printer with no-print border
***IMGC09** IPDS 240-dpi printer (IM/1 image only)
***IMGC10** IPDS 240-dpi printer with no-print border (IM/1 image only)
***IMGC11** IPDS 240-dpi printer (CCITT G4 compression)

We recommend that you use an option that matches your printer hardware. Any other choice of the resolution, for example, causes a considerably longer conversion time and affects the print quality.

A PSF configuration object may be created to optimize the IPDS print driver. The PSF configuration object is described in Chapter 10, “PSF configuration object and the iSeries server” on page 221.

7.2.4 Creating a printer file

The printer file is used to set attributes dedicated to a shared printer. Only a few attributes of the printer file are supported for this purpose. Only one printer file is associated to one shared printer. These attributes are:

- Page size
- Drawer
- Duplex

- Form definition (formdef)
- Copies

Most customer situations require you to print with different attributes on the same shared printer. NetServer allows one printer to be associated with several shared destinations. You can have different printer files associated with different shared printers and use different attributes. Table 26 shows some typical printer file attributes.

Table 26. Printer files example

Name	Page size	Duplex	Copies	Drawer	Formdef
P01A4S11	297 x 210 mm	Simplex	1	1	*
P01A4D11	297 x 210 mm	Duplex	1	1	*
P01LES11	11 x 8.5 inch	Simplex	1	1	*
P01LED11	11 x 8.5 inch	Duplex	1	1	*
***** Specify FORMDF to access AFP functions such as stapling.					

Each of the attibutes is explained here:

- **Page size:** Specifies the length and width of the printer page used by this device file. We recommend that you use the *UOM parameter to set the measurement unit as *cm or *inch to prevent size mismatch.

  Do not use this parameter to specify paper orientation. The orientation information is carried within the PostScript, PDF, or PCL data stream.

  – Letter 11 x 8.5 for Portrait and Landscape (correct)
  – Letter 8.5 x 11 for Landscape (wrong)
  – A4 297 x 210 for Portrait and Landscape (correct)
  – A4 210 x 297 for Landscape (wrong)

- **Duplex:** Specifies whether output is printed on one side or two sides of the paper.
- **Copies:** Number of copies
- **Drawer:** Specifies the source drawer used when cut sheets are fed into the printer (specified by FORMFEED(*AUTOCUT)).
- **Formdef:** A form definition is a resource object that defines the characteristics of the form. See Chapter 12, “Using form definitions and page definitions” on page 255, for additional information about form definition.

Figure 110 shows how to use a different printer file for each shared printer. This allows different job attributes for each shared printer.

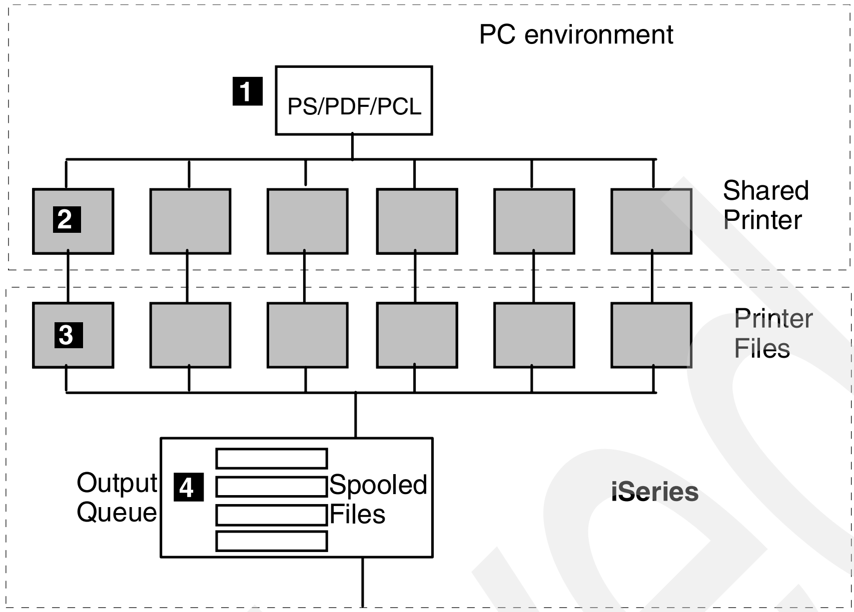


Figure 110. Shared printer and printer files

The process in Figure 110 is explained here:

1. A Windows-based application submits a PostScript, PDF, or PCL print job to a shared printer. These jobs are available on any other printer through a Windows printer menu.
2. You must use a Windows printer definition as explained in 7.3.1, “Sharing the printer with NetServer” on page 188, or 7.3.3, “Printing your file with LPD” on page 194, to submit the print job to the iSeries.
3. A printer file is associated with each shared printer. Specific printer file attributes are used to defined some of the print job options. Note that a unique printer file is associated with the windows printer definition. Only the following attributes are supported by the transform:
   - PAGESIZE
   - DUPLEX
   - FORMDF
   - COPIES

   You need to create a different printer file for each of the different needs you have.
4. The spooled files with the specific attributes are placed in the iSeries output queue. The transform process is invoked as soon as the spooled file status is ready.

7.2.5 Starting and stopping the transform

Enter STRTFMMGR on the command prompt to start Transform Manager. This starts the minimum number of each type of transform specified. See 7.2.2.2, “Setting up the transform configuration file” on page 180, for information about specifying the minimum number of transforms. You can only run one Transform Manager job at a time. If a Transform Manager is already running and you issue this command, the new job is not started, and a diagnostic message is sent to the job log and the screen.

Enter ENDTFMMGR on the command prompt to end Transform Manager. This command ends all transform jobs started by Transform Manager and ends any active jobs from printer writers.

7.2.6 Exit point

Verify the QIBM_QIMG_TRANSFORMS exit point setup to assure no other programs are registered before Infoprint Server. The exit point is set during installation so do not change it. If it is not set up correctly, the Infoprint Server PCL, PDF, and PostScript transforms will not be invoked.

Use the WRKREGINF command as follows:

```
WRKREGINF EXITPNT(QIBM_QIMG_TRANSFORMS) FORMAT(XFRM0100)
```

Choose option 8 (Work with exit programs). The Work with Exit Programs display (Figure 111) appears.

```
                              Work with Exit Programs

Exit point:   QIBM_QIMG_TRANSFORMS     Format:   XFRM0100

Type options, press Enter.
  1=Add   4=Remove   5=Display   10=Replace

              Exit
            Program     Exit
Opt          Number     Program        Library

               5380     QXTRCLIENT     QIPS
               5381     QXTRCLIENT     QIPS
               5382     QXTRCLIENT     QIPS





                                                                        Bottom
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel
```

Figure 111. Work with Exit Programs

The exit program number determines the order in which the transform is used. For example, if you have multiple PS to AFPDS transforms defined, the one with the lower exit program number is used. The default value for each transform type is listed here:

- 5380 QXTRCLIENT QIPS for the PCL to AFPDS transform
- 5381 QXTRCLIENT QIPS for the PostScript to AFPDS transform
- 5382 QXTRCLIENT QIPS for the PDF to AFPDS transform

7.2.7 Font substitution for PostScript and PDF transform to AFP

The Infoprint Server transform uses the Adobe Type 1 font format to create a high quality image. The Adobe type 1 fonts are rasterized and included in the image. The quality depends on the relationship defined by the device description in 7.2.3, “Creating a device description” on page 180.

The PostScript Type 1 fonts used by the Transform are located in the path /QOpenSys/QIBM/UserData/InfoprintServer/Transforms/ps2afp/fonts. You can access the fonts using Operations Navigator as shown in Figure 112.

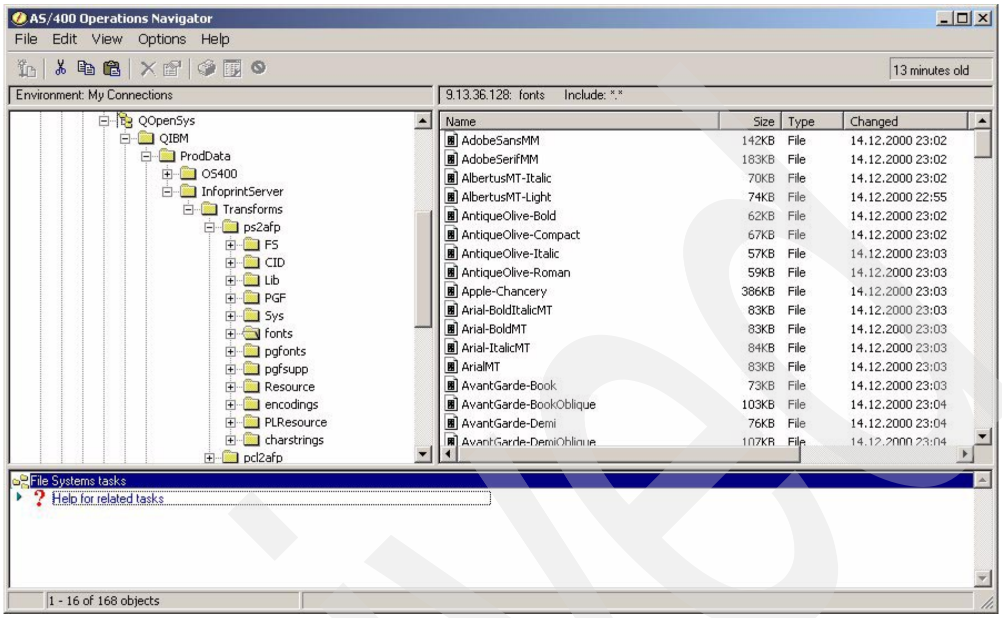


Figure 112. PostScript type 1 fonts

The Transform needs two thing to work with a new font:

- The Adobe type 1 font
- An entry added in the font.map file that describes the font

Note that the fonts do not have an extension. The font used by the transform corresponds to the *.pfb files of the Adobe type 1 font from your PC. Make a copy of the original fonts.map file before you change anything. The Transform will not start if this file is not correct.

To substitute an alternative font for a specific font, complete the following steps:

1. A subdirectory ps2afp need to be created if it isn't there before you can create a fonts.map file. See the complete path in the next step.
2. Create a fonts.map file to the UserData path /QOpenSys/QIBM/UserData/InfoprintServer/Transforms/ps2afp/fonts.

   You can also copy a statement from the IBM provided file located in /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/Transforms/ps2afp/fonts as an example.

   > **Important**
   >
   > Do not modify this file.

3. You can place a copy in the user directory but avoid duplicate entries between the IBM-provided file and the user file.
4. Use an ASCII text editor to open the fonts.map file.

5. Add a new line to the file to include the font name and the path and file name of the font file you want to use, for example:

```
Helvetica   /QOpenSys/QIBM/UserData/InfoprintServer
/Transforms/ps2afp/fonts/Helvetica
```

6. Save the new fonts.map file.
7. Copy the new font (as example Helvetica.pfb) in the same path without extension.

When a transform is started, Transform Manager checks for fonts.map in the UserData path. If it is not found, only the fonts.map in the ProdData path is used. If it is found, the file in the UserData directory is checked for a font first. If the font is not mapped there, the fonts.map file in the ProdData path is searched.

Or, when a font requested within a PostScript data stream is not available on the iSeries, and you do not have the corresponding Adobe type 1 font, a font substitution can be defined if there is a similar font available. *Font substitution* is the mapping of a font name to a font that is available and similar (in terms of its rasterization properties) to the font file being replaced. You can also specify font substitution if existing font mapping is producing undesirable output.

To define a font substitution, complete the following steps:

1. Create a fonts.map file to the UserData path /QOpenSys/QIBM/UserData/InfoprintServer/Transforms/ps2afp/fonts.

   You can also copy a statement from the IBM provided file located in /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/Transforms/ps2afp/fonts, for example.

> **Important**
>
> Do not modify this file.

2. You can place a copy in the user directory but avoid duplicate entries between the IBM-provided file and the user file.
3. Use an ASCII text editor to open the psfonts.map file.
4. Add a new line to the file to include the font name and the path and file name of the font file you want to use, for example:

```
Univers  /QOpenSys/QIBM/UserData/InfoprintServer
/Transforms/ps2afp/fonts/Helvetica
```

5. Save the new psfonts.map file.
6. Copy the new font (as example Helvetica.pfb) in the same path without the extension.

Only a system administrator should change fonts.map. If there are incorrect entries in fonts.map, Transform Manager will not start.

The following fonts are examples of the fonts provided with the AFP font collection and Inforpint Server and available in the psfonts.map file:

Courier, Helvetica, Times New Roman

Older transform user

Previous releases of the PostScript (PostScript Level 1) to AFP transform do not use the same IFS folder structure. These fonts are not accessible to the new PostScript level 3 transform. You need to copy these fonts into the new folder.

7.3 Sending PCL or PostScript/PDF data to the iSeries

Before you can use the PostScript, PDF or PCL to AFP transform, you need to understand the concept and set up the environment. Different access methods allow you to transfer print data to the iSeries output queue.

Imagine that you want to send your print requests directly as a print job from your Windows workstation to an iSeries attached printer. This section describes how you can send PCL, PostScript, or PDF print data to the iSeries. The following methods are described:

- Shared printer with NetServer
- Use LPD (LAN print daemon)

7.3.1 Sharing the printer with NetServer

You need to perform two steps to place print data in an output queue on the iSeries. First you have to define the printer as a shared printer and create the connection with your workstation.

1. Select the Client Access icon on your desktop and select Operations Navigator as shown in Figure 113.

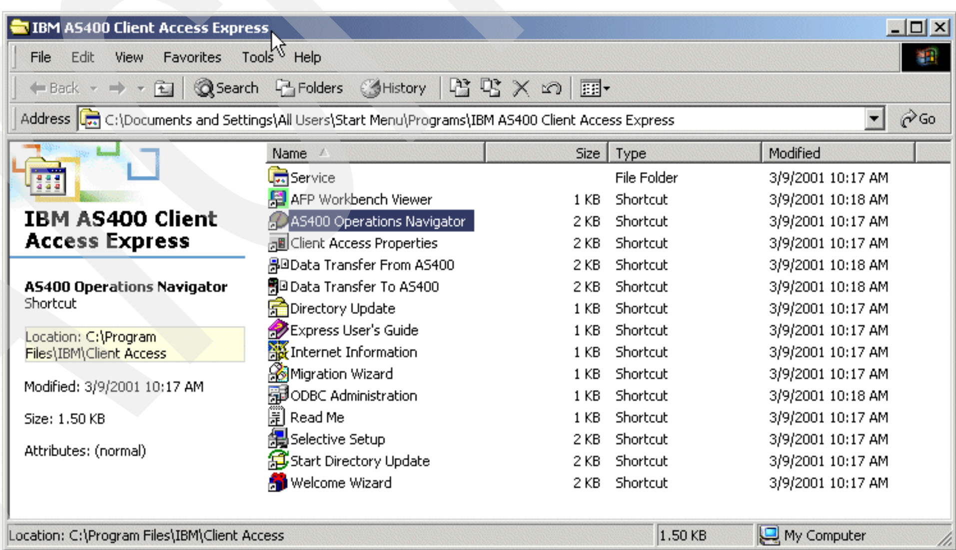


Figure 113. Starting Operations Navigator

2. The Operations Navigator window shown in Figure 114 may differ depending on your environment. Select the system where the printer is defined or expand the connection branch.

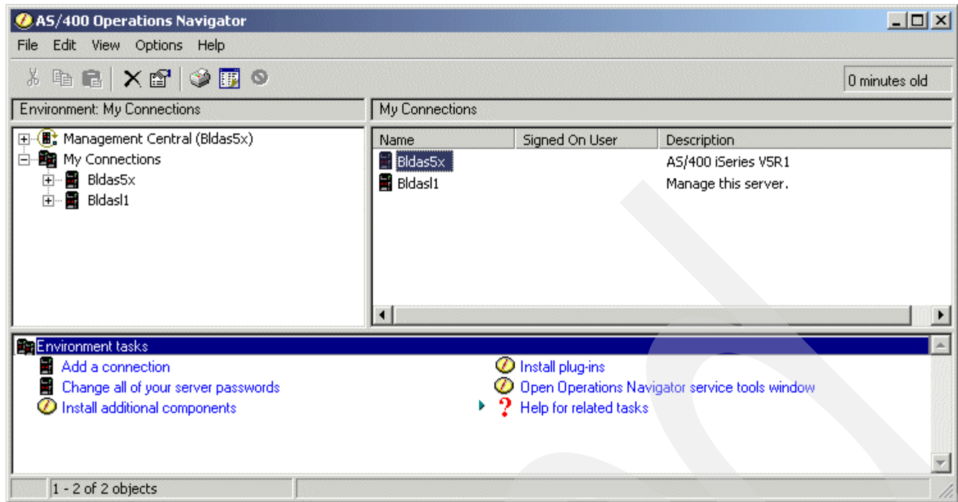


Figure 114. Selecting the system where the printer is defined

3. Select **Basic Operations** in the Operations Navigator window or expand the branch (Figure 115).

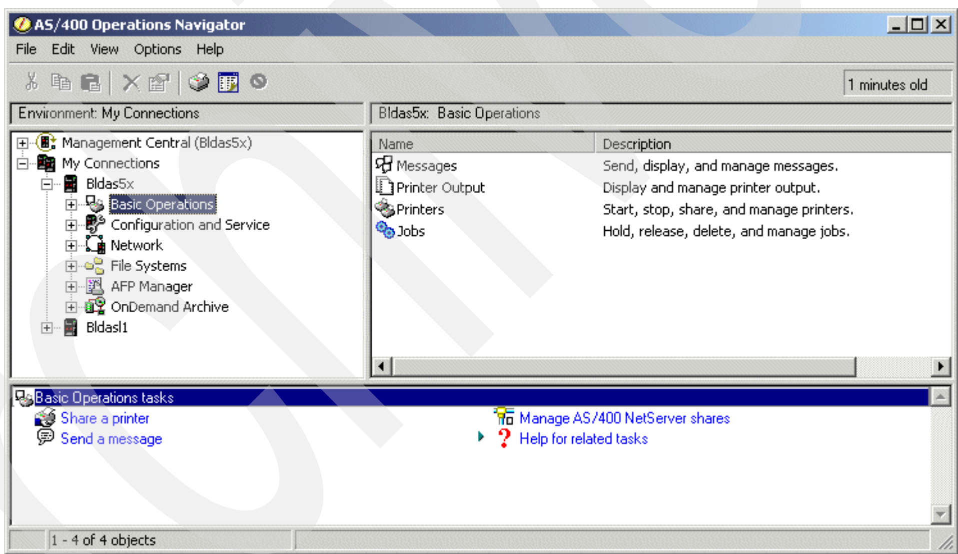


Figure 115. Selecting Basic Operations

4. Select **Printers** to see the list of the available printers on the iSeries (Figure 116).

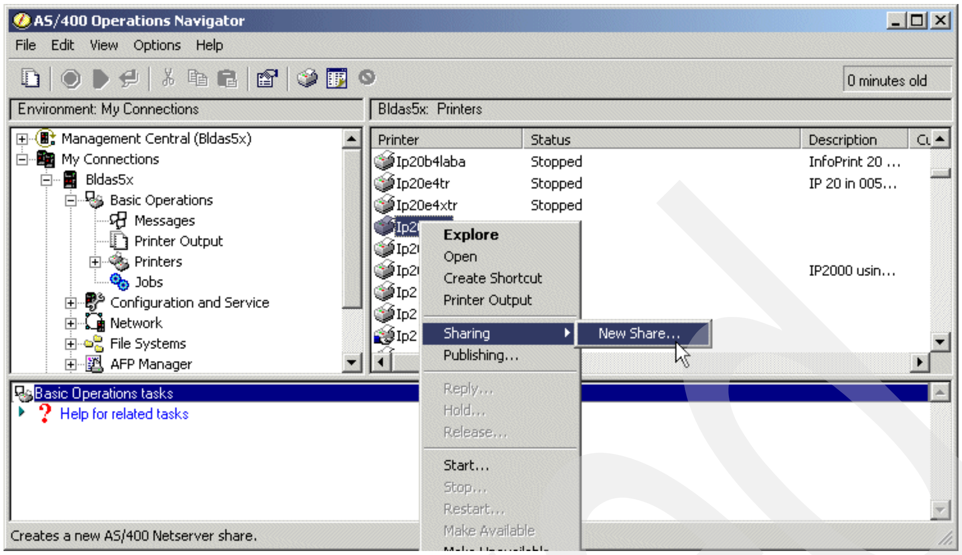


Figure 116. List of printers

5. After the printer list is displayed, select the printer you want to share. Right-click and select **Sharing-> New Share** to open the NetServer Print share window as shown in Figure 117.

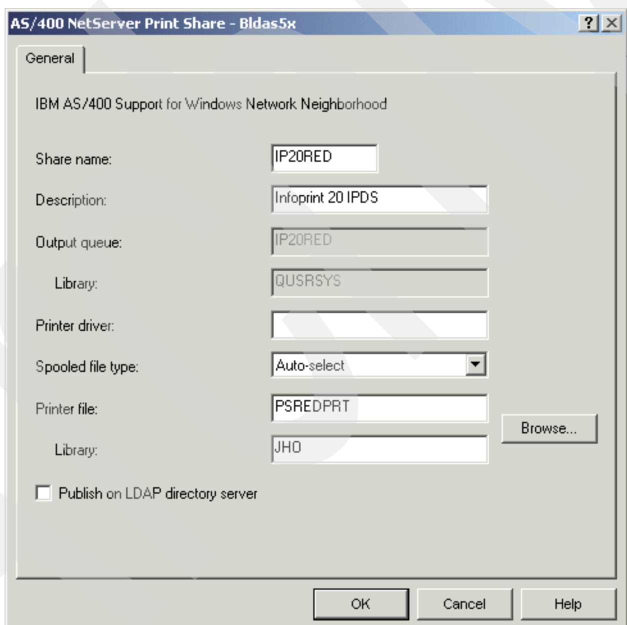


Figure 117. NetServer Print Share

On this window, you see the following fields:

- **Share name**: Specifies the name of the shared printer. The print share name cannot be changed for an existing print share.
- **Output queue**: Specifies the output queue, or you can click Browse to display a list of output queues and libraries.
- **Output queue library**: Specifies the output queue library, or you can click Browse to display a list of output queues and libraries.

- **Printer driver**: Specifies the printer driver. This name should match the name of a printer driver known to the client's operating system. For example, if you will send print jobs to an IBM 4019 laser printer, then you should specify IBM 4019 Laser Printer in this field.
- **Spooled file type**: Lists the type of spooled files that are created for this printer share. Possible values with the transform are:
  - User ASCII for PCL, PostScript, and PDF data
  - Advanced Function Printing (AFP) when the APF print driver is used
  - Auto-select

  Auto-select specifies that AS/400 NetServer will determine the associated data type automatically and create the spooled file with that type. Auto-select is only available if you have Version 4 Release 4 or higher OS/400 installed.

  > **Note**
  >
  > The spooled file type you specify here will override the printer device type you specify in the Printer file field.

- **Printer file:** Specify the name of the file used to define the printer attributes. The printer file that you specify here will be used for all print jobs spooled using this printer share.

  > **Note**
  >
  > The spooled file type you specify in the Spooled file type field will override the printer device type you specify here.

  This function is only available at V4R4 or later.
- **Printer file library**: Specifies the printer file library and provides a space for you to type the name of a new printer file library. You can click Browse to display a list of printer files and libraries on the iSeries.

  > **Note**
  >
  > This function is only available at V4R4 or later.

7.3.2 Creating your NetServer printer

In addition to the shared printer definition, you need to activate the connection between the client (PC) and the iSeries.

1. Select the Network icon on your workspace. Right-click and select **Search for Computer** in the menu. A search window appears as shown in Figure 118.

Remember

Ask for the pre-installation print driver and maintain driver level like:

- Install
- Update
- Maintain
- Automatic-distribution

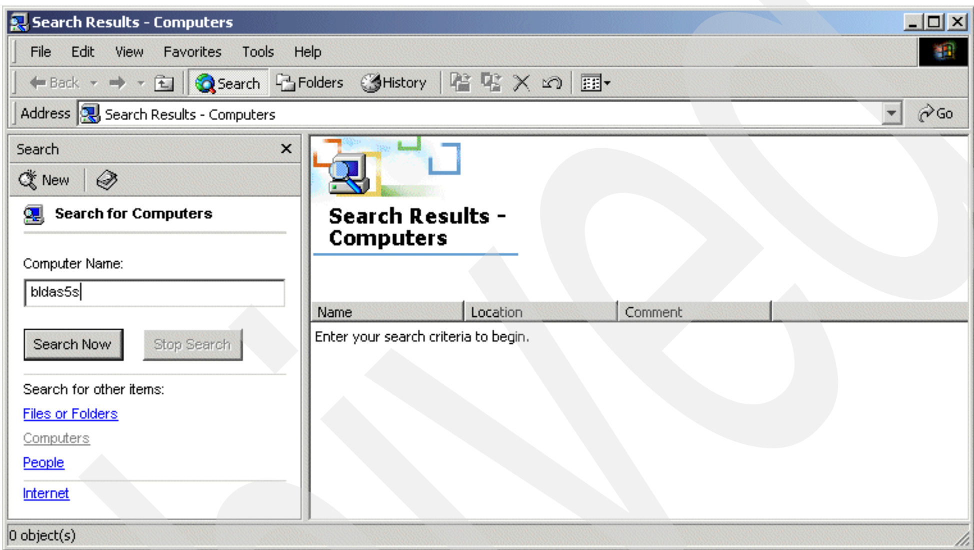


Figure 118. Searching for a computer

2. Enter the name of your iSeries server in the Computer name field. Then click **Search Now**.

   As soon as the system is found, its name appears in the Search Results part of the window as shown in Figure 119.

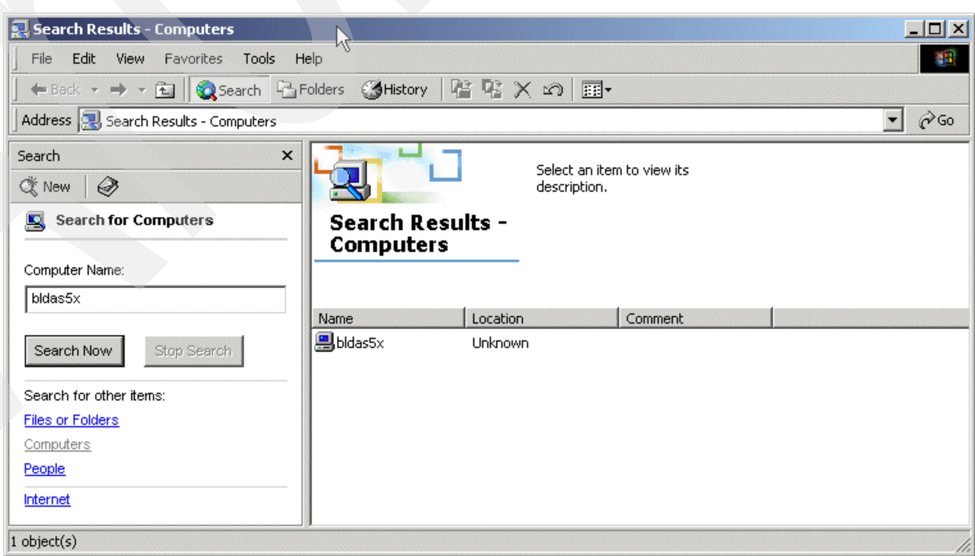


Figure 119. Search results

3. Double-click the system to select it. A list of the available components appears. It may take some time before the list appears depending on your system environment.
4. The share printer we defined must be in the component list as shown in Figure 120. If not, check your share printer definition.

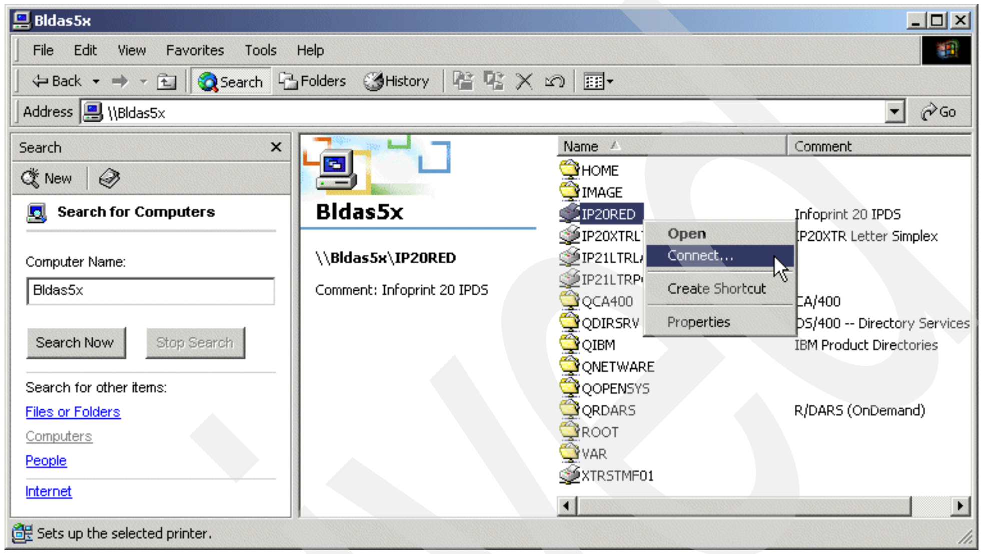


Figure 120. Components list

5. Select the printer. Then right-click and select **Connect** to start the connection between the share printer and the iSeries server.
6. In case of first installation, the printer driver may not be available on the system as indicated by the message shown in Figure 121. In this case, you have to follow the instructions for your operating system.

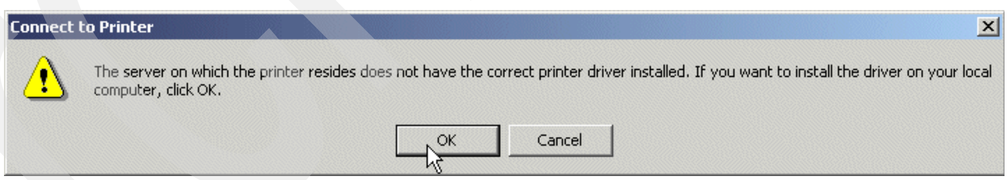


Figure 121. Windows driver requirement

You can check if your connection is active. Type the `net use` command on a prompt as shown here:

```
C:\net use
   New connections will be remembered.


   Status       Local     Remote                    Network

   -------------------------------------------------------------------------------
   ----
   OK           LPT2      \\bldas5x\ip20red         Microsoft Windows Network
   OK           LPT3      \\bldas11\ip20red         Microsoft Windows Network
   OK                     \\bldas5x\IPC$            Microsoft Windows Network
   OK                     \\bldas11\IPC$            Microsoft Windows Network
```

7.3.3 Printing your file with LPD

The LPD command may be used to send print data to the iSeries. You can use the LPD command from any platform to the iSeries. This has the advantage of being quick but does not provide any error recovery. This section provides a few tips on how to send a PCL, PostScript, or PDF spooled files to the iSeries to take advantage of the Infoprint Server transforms.

The LPR command may differ depending of the platform and is not available on all Windows-based operating systems. Windows 95 and 98 do not include LPR/LPD services. You can find the LPD/PPR command on the Web. IBM Printing Systems has a free LPD/LPR command on this Web site:

`http://www.ibm.com/printers`

Checking your printer file

Before you can send any spooled file to the iSeries, check your TCP/IP environment. If TCP/IP is not a common protocol for you, you have to set and define this environment before you can start.

From a printing point of view, the LPD/LPR command is linked to a printer file. The printer file related to the LPD command is QPTMPPRT. Please check the following parameter of the printer file:

```
Device type must be *USRASCII
```

Since only one printer file is used by the LPD command, this can affect other users, especially if you use this command to send print jobs from one iSeries to an other one.

7.3.3.1 Setting your LPD command

This simple command shows you how to send the print job “cedi.prn”:

```
lpr -S bldas5x -P ip20red cedi.prn
```

The command variables are further explained in this example:

```
lpr -S system-name -P outqueue filename.ext
```

lpr The command used on a command window

-S The system name

system name The iSeries name, IP address, or name

-P Indicates that the queue name is the next parameter

queue name Name of the iSeries output queue or printer

file name The file you were sent

We can now send a PCL, PostScript, or PDF file to an output queue on the iSeries. The Infoprint Server recognizes the data stream and invokes the adequate transform to print on the IPDS printer. Note that the iSeries spooled file type is *userascii.

7.3.4 Setting the LPD as a printer port under windows

This example uses an IPDS printer. The printer device description, output queue, and PSF configuration object are requested to drive a LAN-attached IPDS printer.

1. Select the **Add Printer** icon on the desktop. Or using the menu, click **Start-> Setting** to access to the Add Printer Wizard.
2. After the welcome window appears, click **Next** to access to the next panel as shown in Figure 122.
3. Select **Local printer** but do not activate the Automatically detect and install my Plug and Play printer option.

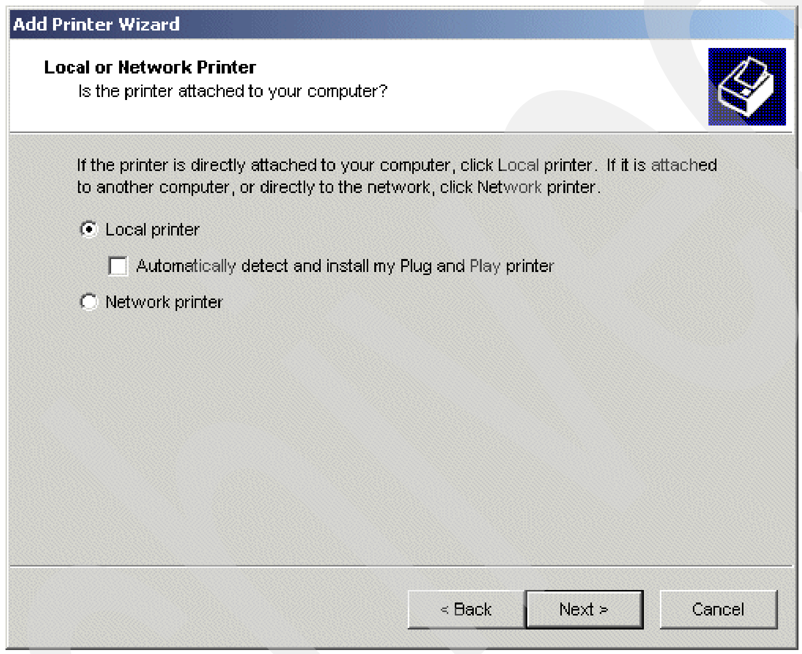


Figure 122. Selecting a local or network printer

4. Click **Next** and the Select the Printer Port panel appears (Figure 123). Select **Create a new port** and choose **LPR Port**.

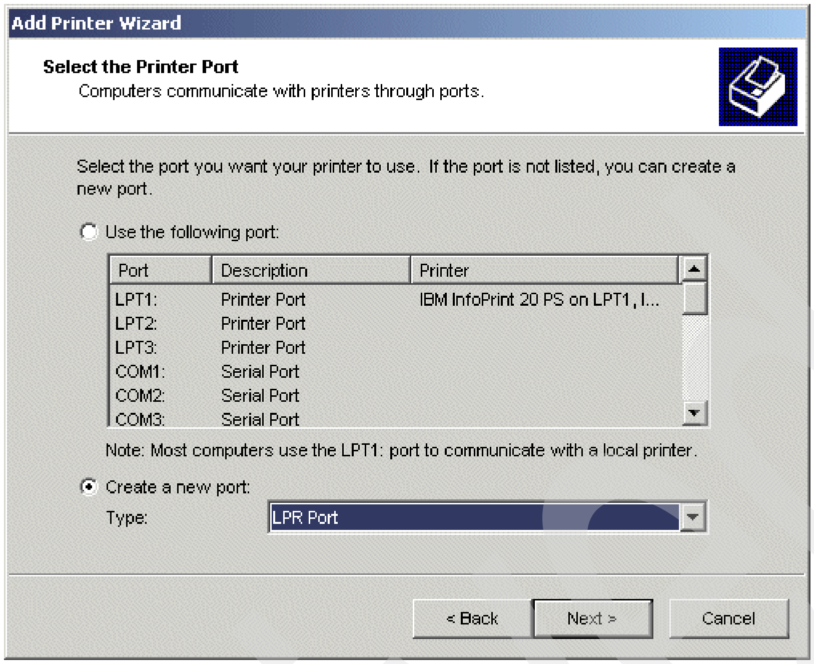


Figure 123. Selecting the printer port

5. Click **Next** to continue to define your LPR port as shown in Figure 124.
6. The iSeries device description and output queue have to be created before this operation. In this example, an IPDS printer “ip20red” is used.

   Type the iSeries server name or address in the first field and the iSeries output queue in second one. Then click **OK** to continue.

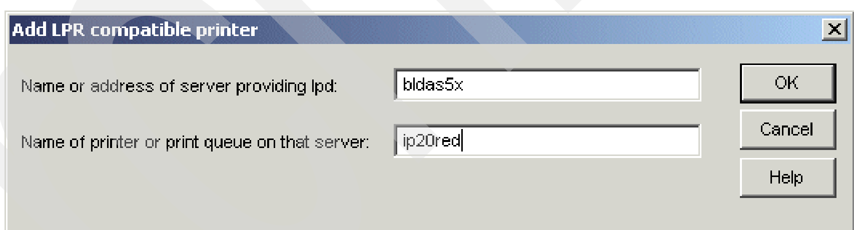


Figure 124. Adding an LPR compatible printer

7. Select a printer driver as shown in Figure 125.

   In this example, the IBM Infoprint 20 PostScript driver is used. You can choose any PostScript, PCL, or PDF driver. Because these kinds of drivers are mostly hardware dependent, the syntax used to create the print data is different. This influences the transform, printed paper, and specific hardware-related options like input paper selection, duplex, and output selection.

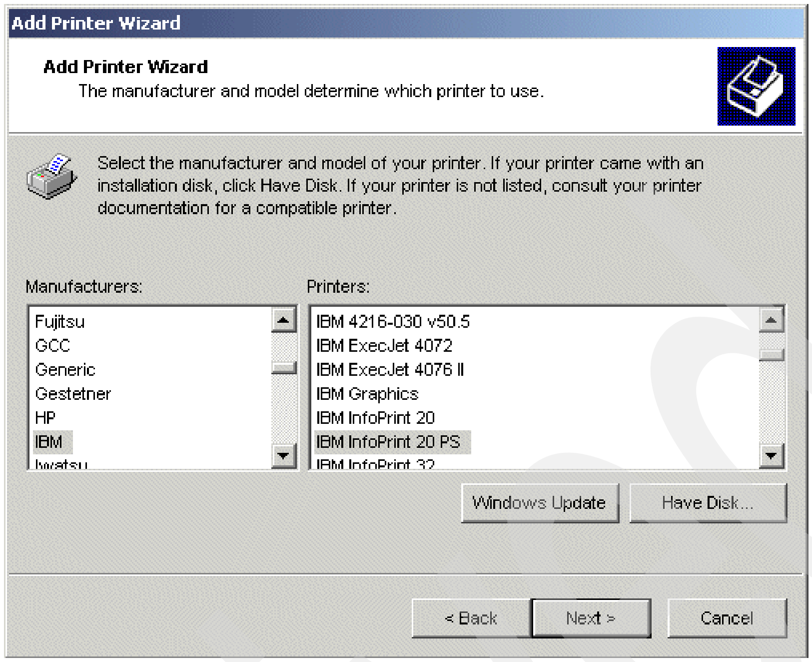


Figure 125. Selecting the print manufacturer and model for LPR port

Under certain circumstances, the system asks you to replace or leave an existing print driver shown in Figure 126.

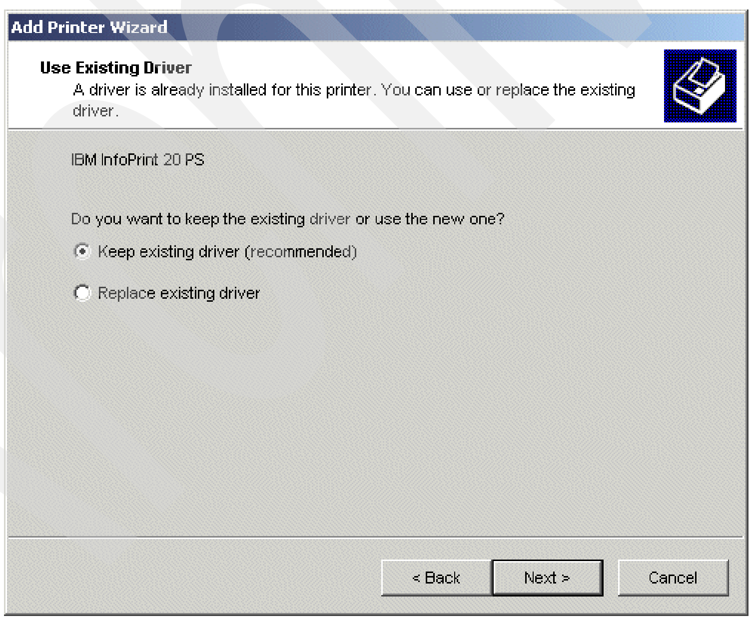


Figure 126. Use Existing Driver

We recommend that you select **Keep existing driver** at this time. Confirm by clicking **Next**.

8. Proceed to the next window to name your printer as shown in Figure 127. Give a name to the printer. The name appears under the printer icon after the printer is created. If you want use this new printer as the default for your

Windows-based application, select **Yes**. In this case, all our documents are printed on this printer.

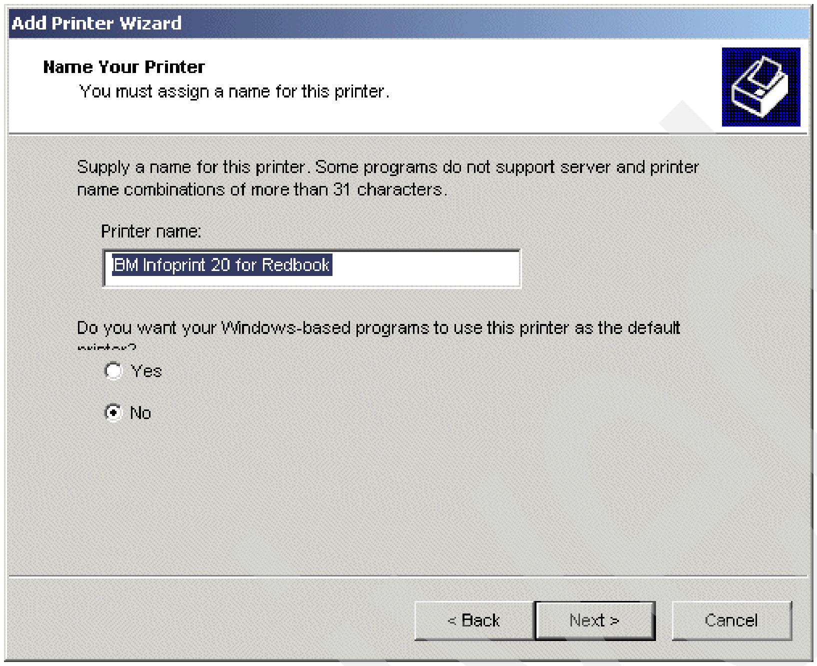


Figure 127. Naming your printer

9. Click **Next** to continue the configuration.

10. At this point on the Printer Sharing panel (Figure 128), you can share this Windows printer definition. However, this printer definition is already a kind of shared printer. It is really connected to the iSeries. Your spooled files are sent to the iSeries output queue.

    Setting the Shared option to “Yes” makes your print definition accessible to other Windows users. This causes additional overhead and workload on your own workstation and on the network. We recommend you select the **Do not share this printer** option. Other Windows users may have to create the same printer description as described in this section.

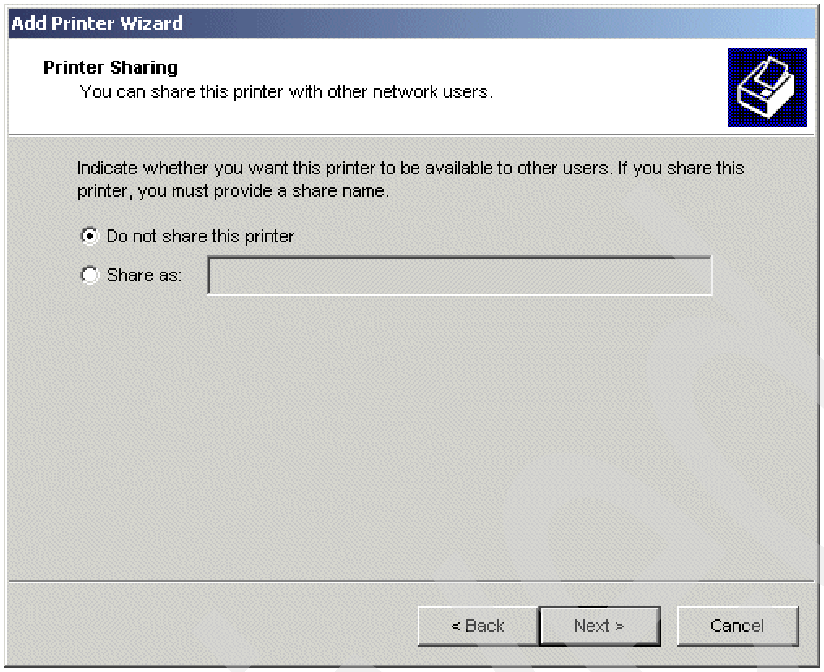


Figure 128. Printer sharing

11.Click **Next** to continue with the Print Test Page panel shown in Figure 129.

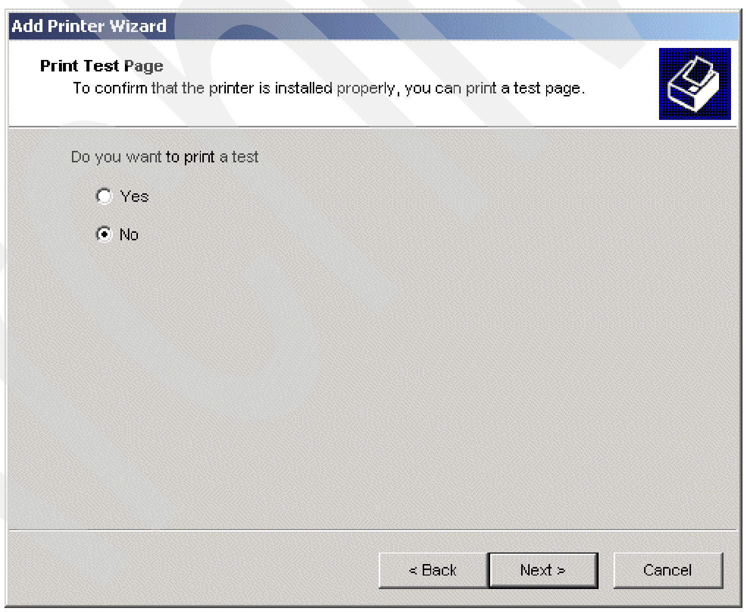


Figure 129. Printing a test page

12.If you decide to print a test page a PostScript test page (by selecting **Yes**), it is sent to print. You can only select this option if your Transform Manager environment was previously configured. Section 7.2, “Defining the PCL, PostScript, and PDF transform environment” on page 176, describes how to set up the transform on the iSeries. Select **No** if the Transform Manager configuration was not completed, and then continue by clicking **Next**.

13.The completion window appears as shown in Figure 130. It summarizes the selected options. Click **Finish** to complete the definition.

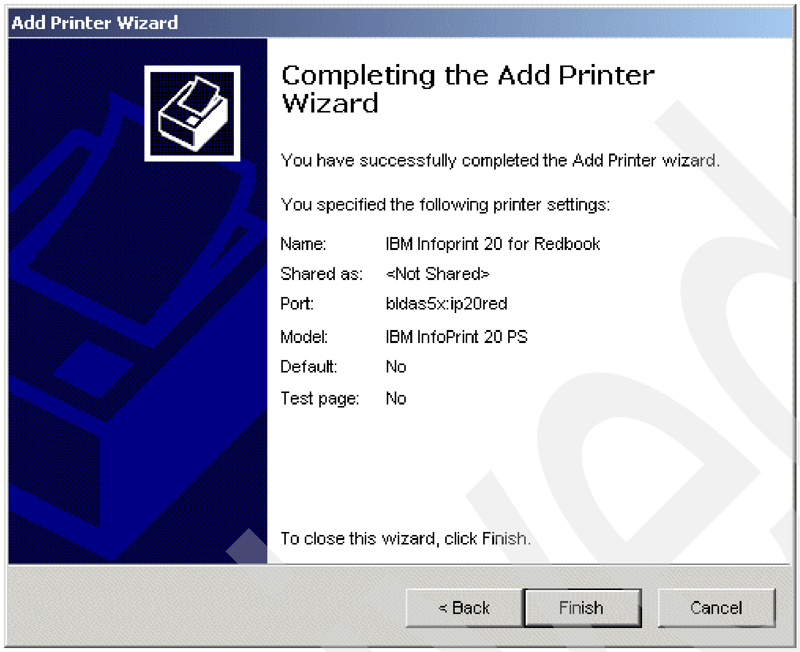


Figure 130. Completing the Add Printer Wizard

You can now open the application and print PostScript document to an iSeries IPDS printer. The Infoprint Server can sniff the data stream in invoke the PostScript to AFP transform.

Figure 131 shows the PostScript spooled on the iSeries server.

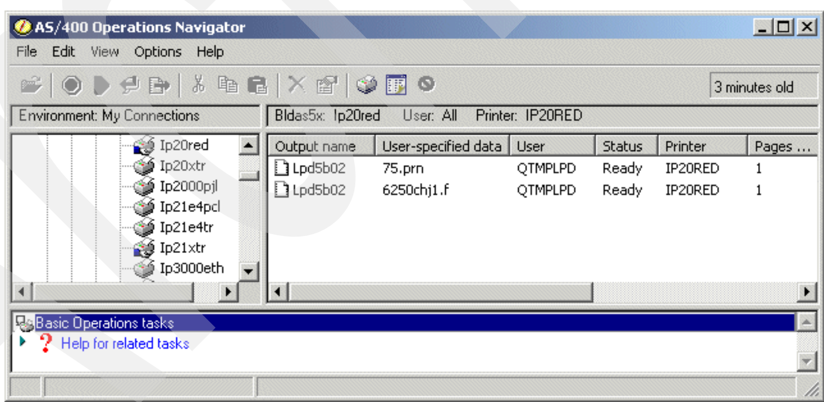


Figure 131. Spooled file in Operations Navigator

Operations Navigator has been used in this example to display the contents of the iSeries output queue. You will recognize two print jobs with following attributes:

- **Output name**: All print jobs you sent from the Windows workstation have the same output name. This name depends on the printer file used by the LPD command.
- **User-Specific data**: This parameter contains the name of the Windows print job and allows you to recognize the document.

- **User**: The user parameter does not contain information about the Windows user. The name “QTMPLPD” is the name of the job from the LPD process.

7.3.4.1 What you need to know about Printer File and LPD/LPR

When we sent a spooled file from another system to the iSeries, we used the printer file QTMPLPD. We recommend that you make a copy of this printer file before you make any modifications. It is the only printer file used for all LPR jobs from any other system. Any change may affect other print job submissions.

Chapter 8. PostScript to AFP using a different input paper tray

This chapter explains how you can use the PostScript to AFP transform with different paper input trays. The PostScript to AFP transform is common across different platforms. However, this chapter is related to the Infoprint Server for iSeries. The same modifications and tasks may also be applied to other environments.

The PostScript to AFP transform is part of Infoprint Server for iSeries. The transform can determine which paper input trays have been used in the PostScript data stream. PostScript drivers are not consistent in the way they select paper input trays. This chapter explains which elements are relevant for the transformation, including:

- How PostScript drivers work
- Modifying the PS2AFP configuration file
- Creating a PostScript document using different paper input trays
- Understanding PostScript data stream tray selection
- Understanding transform paper input tray allocation
- Debugging the AFP data stream

We used the Infoprint 40 PostScript driver to demonstrate how to use tray mapping between the PostScript data stream and AFP.

8.1 How PostScript drivers work

PostScript drivers use different methods to select a paper input tray (depending on the driver). Two different methods were considered in our test, but only one was used successfully:

- Most of the PostScript drivers use a hard tray call in the data stream with the command:

  ```
  %%BeginFeature: *InputSlot Tray1
  ```

  This chapter explains how the transform recognizes and interprets this command.

- Some new drivers, like the Infoprint 2000 PostScript driver, *cannot* be used with the transform because it only generates proprietary `"KDKSlip:"` commands, which the ps2afp transform does not honor. This restriction applies to any drivers using the KDKSlip command.

The Infoprint 40 PostScript driver can be used to demonstrate how to use tray mapping between the PostScript data stream and AFP. The AFP data stream does not depend on a specific printer. You can also print the converted files to any IPDS/AFP printer. If an AFP bin (input paper tray) is missing, the *default AFP bin is invoked.

Figure 132 shows the relationship between the PostScript and the resulting AFP tray command.

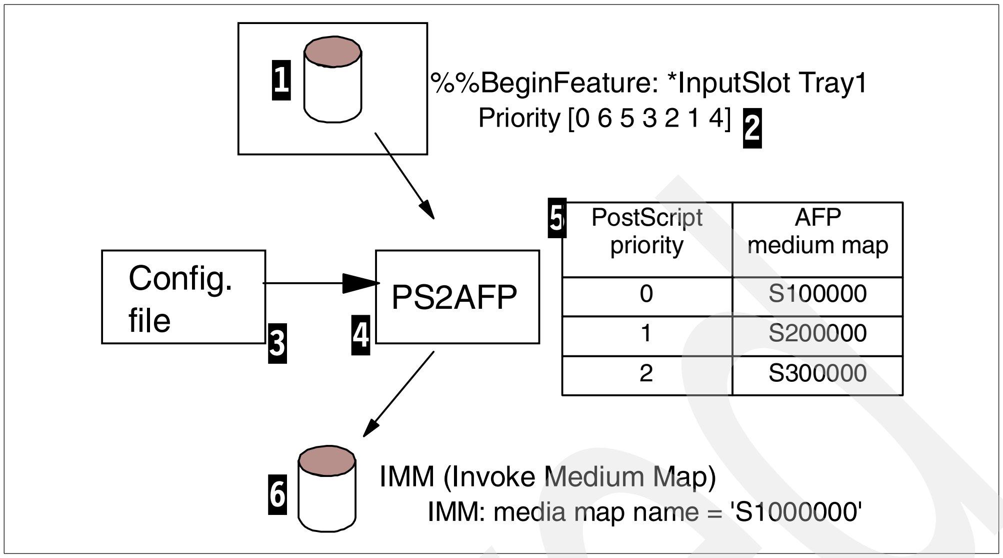


Figure 132. PostScript to AFP tray mapping relationship

The following list explains the process flow shown in Figure 132:

1. A PostScript spooled file using different paper input trays is sent to the iSeries server.
2. One or more paper input selections may be used in the PostScript file. Each statement is shown in the tray selection “*Inputslot Tray 1” and the priority that defines which physical tray is invoked. Depending of the driver, the priority may be different for other printers that are used in this example. Refer to your printer and driver manual, or use the method described in 8.3.2, “Paper input tray statement in PostScript” on page 207.
3. The PS2AFP configuration file must have the correct parameters. Refer to 8.2, “Modifying the PS2AFPD configuration file” on page 204.
4. As soon as a PostScript file is in the output queue, the Transform Manager invokes the appropriate transform. The Transform Manager can invoke the PCL, PDF, or PostScript transform depending on the data type. The transform (PS2AFP in this case) starts using the options defined in the configuration file.
5. An internal mapping table is used by the transform to define the relationship between the PostScript and the AFP statement.
6. A device independent AFP data stream is created. An internal AFP form definition (formdef) is created and Invoke Medium Map is placed on the page boundary to select the AFP bin (paper input tray).

8.2 Modifying the PS2AFPD configuration file

The PS2AFP transform uses a configuration file with default parameters. This file may be modified to take advantage of specific functions. If incorrect parameters are specified, the transform startup fails. All printers (all instances of the transform) will use the same unique settings.

You must first edit the PS2AFPD.cfg configuration file. The following example explains how to modify the configuration file on the iSeries server. The configuration file PS2AFPD.cfg is located in /QOpenSys/QIBM/ProdData/InfoprintServer/Transforms/ps2afp.

Add the following statement at the end of the configuration file. The file may be edited using any standard PC-based editor (remember to make a copy of the original configuration file before you start):

```
device_controls = plex,
input1=(Letter),input2=(Letter),input3=(Letter),input4=(Letter),input5=(Letter
),input6=(Letter),input7=(Letter)
```

This command allows you to determine how many trays the transform can recognize. Note that the paper must be the same for each input statement. If a different paper is allocated to an input tray, the statement is ignored by the transform and the default paper input tray is applied.

The following command is set for the A4 paper format instead of letter:

```
device_controls = plex,
input1=(A4),input2=(A4),input3=(A4),input4=(A4),input5=(A4),input6=(A4),input7
=(A4)
```

Note

This example shows several different input trays. If a tray does not exist on the physical printer, the default tray will be used.

After you place this new line in the configuration file, you should restart the Transform Manager. The command will be effective after the new start. Some system environment variables may overwrite the configuration file at each new start. In this case, follow the system documentation to place the command in the right place.

8.3 Creating a PostScript document using different paper input trays

You may use software that can allocate different input trays in the same document. Some software does not allow different selections on the page level, but only allows one selection for the first page and for the following pages.

For this example, Lotus WordPro was used. The only way to allocate different trays for different pages in the same document is to define different divisions. You can set a different paper input tray for each division in the WordPro document.

The following example shows how to recognize and interpret the PostScript command for paper input tray. For this example, the Infoprint 40 PostScript driver was used.

8.3.1 PostScript driver setup

Your PostScript driver must have the correct feature set up. After you install the driver, you may verify whether the driver can select different paper input trays. You can install or re-install your driver for this test and add additional hardware features like the 2500 additional input tray. This allows you to create a PostScript data stream with different trays.

Figure 133 shows the Infoprint 40 Device Settings. To use more trays, you must first install an additional option.

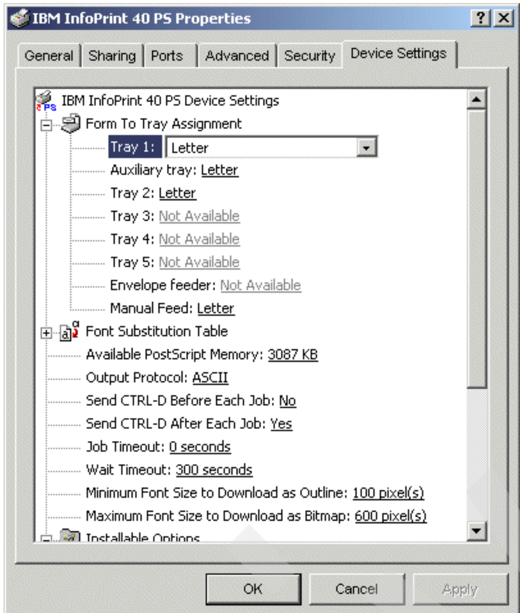


Figure 133. Infoprint 40 settings

Scroll down to see the device options as shown in Figure 134. You can install an additional device option. The additional 2500 sheet input tray option is used in this example. Note that you do not require a physical printer to add an option as described above.

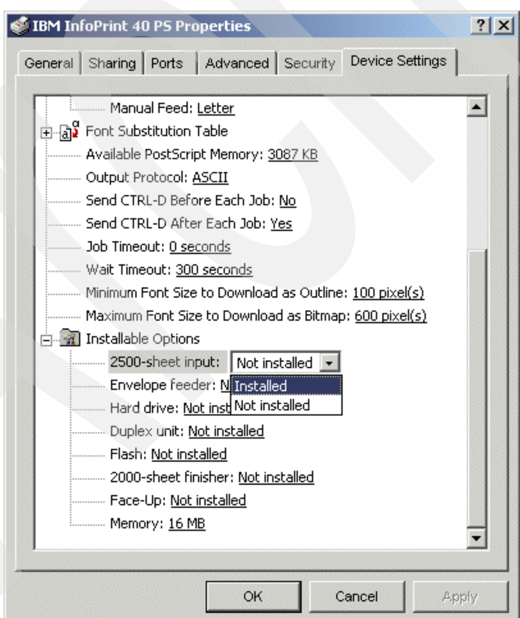


Figure 134. PostScript driver device options

The next step sets a paper size to each input tray as shown in Figure 135.

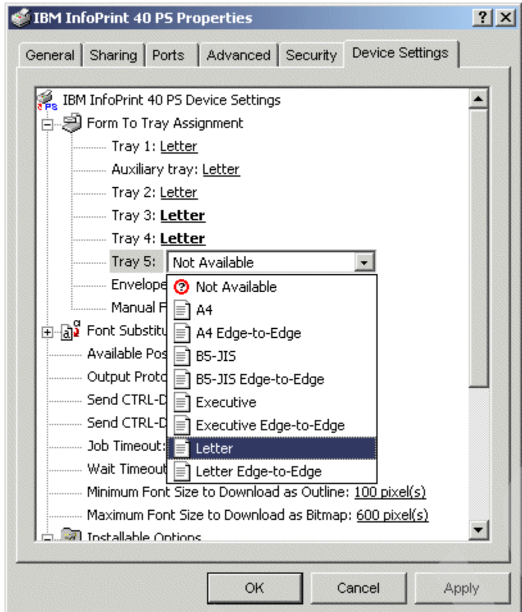


Figure 135. PostScript driver paper size

The paper size must match the paper size specified in the configuration file of the PostScript to AFP transform. Specify the same paper size for each input tray. You can now specify a paper input tray for each division of your document.

8.3.2 Paper input tray statement in PostScript

You can produce a PostScript file on your disk. The file may be used to check the PostScript statement or to determine how another driver allocates trays and tray priorities.

The following examples show how the Infoprint 40 PostScript driver works. Note that the important statement for the PostScript to AFP transform is the *priority*. Different drivers may use different priorities for the same paper input tray. Section 8.4, "Understanding the transform paper input tray allocation" on page 208, explains how the transform maps the PostScript statement in an AFP Invoke Medium Map (IMM) to allocate a different tray.

- **PostScript statement for paper input tray 1:**

```
%%BeginFeature: *InputSlot Tray1
 currentpagedevice /InputAttributes get 0 known
 {<<currentpagedevice /InputAttributes get 0 get dup null ne
    {/PageSize get /PageSize exch
    /InputAttributes <</Priority [0 6 5 3 2 1 4] >> >> setpagedevice}{pop
pop} ifelse} if
%%EndFeature
```

- **PostScript statement for paper input tray 2:**

```
%%BeginFeature: *InputSlot Tray2
 currentpagedevice /InputAttributes get 2 known
 {<<currentpagedevice /InputAttributes get 2 get dup null ne
    {/PageSize get /PageSize exch
    /InputAttributes <</Priority [2 6 5 3 0 1 4] >> >> setpagedevice}{pop
pop} ifelse} if
%%EndFeature
```

- **PostScript statement for paper input tray 3:**

```
%%BeginFeature: *InputSlot Tray3
currentpagedevice /InputAttributes get 3 known
 {<<currentpagedevice /InputAttributes get 3 get dup null ne
    {/PageSize get /PageSize exch
    /InputAttributes <</Priority [3 6 5 2 0 1 4] >> >> setpagedevice}{pop
pop} ifelse} if
%%EndFeature
```

Priority

Note that the first priority does not match the paper input tray number. The priorities are hardware dependent and may change from one driver to the next.

8.4 Understanding the transform paper input tray allocation

You can now understand how a PostScript driver works. Because each printer or printer family may have its own driver, the priority can be different for the same PostScript input paper tray.

The PostScript to AFP transform maps the first digit of the priority sequence to an AFP bin number. The value 1 is added to the first digit of the priority sequence to determine the AFP medium map or AFP paper input tray.

The following example is based on a customer requirement. This customer is required to use three different paper input trays on their AFP/IPDS printers (two different types) using the PostScript to AFP transform.

Table 27 shows how trays 1, 2, and 3 from the previous example are interpreted by the transform.

Table 27. Trays interpreted by the transform

PostScript input tray	PostScript priority	AFP medium map	AFP tray
1	`0 6 5 3 2 1 4`	S100000	1
2	`2 6 5 3 0 1 4`	S300000	3
3	`3 6 5 2 0 1 4`	S400000	4

In the first line of this example, you can see that PostScript input tray 1 is the same as the AFP because the priority is 0. PostScript input tray 2 is set as AFP input tray 3, and the PostScript input tray 3 is set as AFP input tray 4.

Table 28 shows all the capabilities of the Infoprint 40 PostScript driver and the AFP allocated tray from the transform.

Table 28. Infoprint 40 PostScript driver capabilities

AFP input tray	Medium map	PostScript tray	Priority
1	S1000000	1	`0 6 5 3 2 1 4`
2	S2000000	Auxiliary	`1 6 5 3 2 0 4`
3	S3000000	2	`2 6 5 3 0 1 4`

AFP input tray	Medium map	PostScript tray	Priority
4	S4000000	3	3 6 5 2 0 1 4
5	S5000000	Not available	Not available
6	S6000000	4	5 6 3 2 0 1 4
7	S0700000	5	6 5 3 2 0 1 4

Table 28 shows that the customer must use the input trays 1, auxiliary, and 2 on the PostScript driver if they want to use the AFP trays 1, 2, and 3.

8.4.1 Debugging the AFP data stream

If the correct paper input trays were not used on the printer, you may check the PostScript job again. If the PostScript job is correct, check the transform configuration file to see whether the configuration file contains the correct instruction described in 8.2, “Modifying the PS2AFPD configuration file” on page 204. If both are correct, check the AFP data stream as described in the following section. Two steps are required:

1. Generate the AFP data stream as a file.
2. Analyze the AFP data stream.

8.4.1.1 Generating the AFP data

The transform may not be invoked from the OS/400 command line or prompt. The following C program uses an API to create the streamed file. The PostScript file is used as input. The full qualified path must be defined for all three parameters as shown in the program header.

Disclaimer

This program example is provided "as is". It has not undergone any formal testing. It is not available for maintenance, support, or updates.

```
/***********************************************************************************************
******
* File Name: cvtps2afp.c                                                                      *
* Purpose: Uses the QimgCvtImg API to convert a PostScript stream file
*
*          to an AFP stream file.                                                             *
* Usage:                                                                                      *
* cvtps2afp inputFileName outputFileName imgConfigName
*
*    Where:                                                                                   *
*    inputFileName  - is the fully qualified path name of the input datastream
*
*    outputFileName - is the fully qualified path name of the output datastream
*
*    imgConfigName  - is the name of a valid image configuration object
*
* Example:                                                                                    *
* CALL PGM(CVTPS2AFP) PARM('/home/userid/inputFile.ps' '/home/userid/outputFile.afp'
'*IMGC05')  *
* CRTBNDC PGM(CVTPS2AFP) SRCSTMF('/home/<dirname>/cvtps2afp.c')
*
* SYSIFCOPT(*IFSIO) INCDIR('/QIBM/ProdData/ILEC/include/qsysinc')
*
* Note: The compiler’s default include path is                                                *
* /QIBM/ProdData/ILEC:/QIBM/ProdData/ILEC/include:/QIBM/include.
*
*                                                                                             *
***********************************************************************************************
*****/
```

```
#include <stdio.h>
#include <string.h>
#include <qusec.h>
#include <qimgcimg.h>
typedef struct input_Image_Structure {
    Qimg_IMGI0100_t inputImgStruct;
    char Obj_Nam[512];  /* variable data */
} input_Image_Structure_t;
typedef struct output_Image_Structure {
    Qimg_IMGO0100_t outputImgStruct;
    char Obj_Nam[512];  /* variable data */
} output_Image_Structure_t;
typedef struct ec{
    Qus_EC_t EC;
    char exceptionData[256];
} ec_t;
int main (int argc, char *argv[])
{
    char inputFileName[512];
    char outputFileName[512];
    char imgConfigName[11];
    char exceptionId[8] = { '\0'};
    Qimg_IMGC0100_t          controlStruct;
    input_Image_Structure_t  input;                   /* stream file input  */
    output_Image_Structure_t output;                  /* stream file output */
    ec_t                     err;
    if (argc < 4) {
        printf("Usage: cvtps2afp inputFileName outputFileName imgConfigName\n\n");
        printf("Where:\n\tinputFileName is the fully qualified path name of the input
datastream");
        printf("\n\toutputFileName is the fully qualified path name of the output
datastream");
        printf("\n\timgConfigName is the name of a valid image configuration object");
        printf("\n\nExample:\n\tcvtps2afp /home/username/test.ps /home/username/test.afp
*imgc06");
        exit(1);
    } /* end if */
    strcpy(inputFileName, argv[1]);
    strcpy(outputFileName, argv[2]);
    strcpy(imgConfigName, argv[3]);
    /************************************
    *                                  *
    *     QimgCvtImg Control Structure  *
    *                                  *
    ************************************/
    controlStruct.Struct_Len = sizeof(controlStruct);
    strcpy(controlStruct.Struct_Fmt, CONTROL_FMT_100);
    controlStruct.Operation = IMMED_IMAGE;
    memset(controlStruct.Append_Handle, 0, 32);
    controlStruct.Fb_Len = 0;
    strcpy(controlStruct.Fb_Struct_Fmt, FEEDBACK_FMT_100);
    controlStruct.Reverse = QIMG_NO;
    controlStruct.Color_Reduce = QIMG_SAME ;
    controlStruct.Resize = QIMG_KEEPSIZE;
    controlStruct.Stretch = QIMG_NO;
    controlStruct.H_Just = QIMG_HCENTER;
    controlStruct.V_Just = QIMG_VCENTER;
    controlStruct.Keep_Color = QIMG_YES;
    controlStruct.Keep_Quality = QIMG_YES;
    controlStruct.Cancel = QIMG_NO;
    controlStruct.Sev_Level = 99;
    /****************************************
    * QimgCvtImg Input Structure           *
    *                                      *
    ****************************************/
    memset(&input, 0, sizeof(input));             /* intialize */
   input.inputImgStruct.Struct_Len = sizeof(input.inputImgStruct) + strlen(inputFileName);
    strcpy(input.inputImgStruct.Struct_Fmt, STREAM_FIL_IN);
    input.inputImgStruct.Data_Strm_Fmt = QIMG_DS_PS;
    input.inputImgStruct.Photo_I = QIMG_PH_INOBJ;
    input.inputImgStruct.Res_Unit = QIMG_RU_INOBJ;
    input.inputImgStruct.H_Res = QIMG_RS_INOBJ;
    input.inputImgStruct.V_Res = QIMG_RS_INOBJ;
    input.inputImgStruct.Obj_Typ = 0;             /* steam file */
    input.inputImgStruct.Obj_Nam_Ofs = 68;
    input.inputImgStruct.Obj_Nam_Len = strlen(inputFileName);
    strcpy(input.Obj_Nam, inputFileName);
    /****************************************
```

```
    * QimgCvtImg Output Structure           *
    *                                       *
    ****************************************/
    output.outputImgStruct.Struct_Len = sizeof(output);
    strcpy(output.outputImgStruct.Struct_Fmt, STREAM_FIL_OUT);
    memset(output.outputImgStruct.Img_Cfg_Obj, 0x40,
sizeof(output.outputImgStruct.Img_Cfg_Obj));
    strncpy(output.outputImgStruct.Img_Cfg_Obj, imgConfigName, strlen(imgConfigName)); /*
must be padded with blanks */
    memset(output.outputImgStruct.Img_Cfg_Obj_Lib, 0,
sizeof(output.outputImgStruct.Img_Cfg_Obj_Lib) );
    output.outputImgStruct.Data_Strm_Fmt = QIMG_DS_AFPDS;
    output.outputImgStruct.Photo_I = QIMG_PH_DESTIMGCFG;
    output.outputImgStruct.Res_Unit = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.H_Res = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.V_Res = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.Siz_Unit = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.H_Siz = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.V_Siz = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.Cpr_Typ = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.Cpr_Qual = 0;
    output.outputImgStruct.Bit_Smp = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.Ppr_Siz = QIMG_PS_LETTER;
    output.outputImgStruct.Usr_Ppr_Siz_Unit = QIMG_RS_DESTIMGCFG;
    output.outputImgStruct.H_Usr_Ppr_Siz = 0;
    output.outputImgStruct.V_Usr_Ppr_Siz = 0;
    output.outputImgStruct.Ppr_Ori = QIMG_PO_DESTIMGCFG;
    output.outputImgStruct.Left_Border = QIMG_BD_DESTIMGCFG;
    output.outputImgStruct.Right_Border = QIMG_BD_DESTIMGCFG;
    output.outputImgStruct.Top_Border = QIMG_BD_DESTIMGCFG;
    output.outputImgStruct.Bottom_Border = QIMG_BD_DESTIMGCFG;
    output.outputImgStruct.Obj_Typ = QIMG_STR_OBJ_TYP;  /* stream file output */
    output.outputImgStruct.Obj_Nam_Ofs = 144;
  output.outputImgStruct.Obj_Nam_Len = strlen(outputFileName);
    output.outputImgStruct.Reserved1 = 0;              /* RESERVED must be set to 0x00 */
    strcpy(output.Obj_Nam, outputFileName);
    memset(&err, 0, sizeof(err));
    err.EC.Bytes_Provided = sizeof(err);
    QimgCvtImg (&controlStruct,       /* Control structure              */
                &input,               /* Input Image structure          */
                (void *)NULL,         /* Main storage input image       */
                &output,              /* Output image structure         */
                (void *)NULL,         /* Main storage output image      */
                (void *)NULL,         /* Feedback structure             */
                &err        );        /* Error code                     */
    if (err.EC.Bytes_Available == 0) {
        printf("\n\nNo error detected");
    } else {
        printf("\nError detected!");
        strncpy(exceptionId, err.EC.Exception_Id, sizeof(err.EC.Exception_Id));
        printf("\nException Id: %s", exceptionId);
        printf("\nException Data: %s", err.exceptionData);
    } /* end if */
}   /* end main */
```

8.4.1.2 Checking the AFP data stream with the AFPdmp program

You must analyze the AFP data stream. You can use the AFPdmp program to check the AFP structured fields. This program may be downloaded from the IBM Printing Systems Web page at: `http://www.ibm.com/printers`

Navigate to **Support-> Drivers, downloads, fixes and updates**. Under Unsupported Downloads, click the **Tools** link. Different versions of the AFPdmp program are available for your particular operating system. In this section, we used the AFPdmp program for OS/2 and Windows. Download the program in your work directory. Then, copy the AFP streamed file from the IFS to your PC disk.

Open a command prompt and type:

```
afpdmp infile.afp > outfile.txt -d
```

This command includes the following components:

afpdmp The name of the dump program

infile.afp The name of my AFP file (produced with the transform)

> Indicator to write the dump in a file

outfile.txt The file that contains the AFP dump

-d Option to indicate that the detail of the AFP data stream is to be provided

A list of all options of the AFPdmp program can be displayed with the command:

```
afpdmp ?
```

8.4.1.3 Finding the paper input tray statements

The Invoke Medium Map (IMM) structured field is used in AFP to call an input paper tray. You must find these IMM structured fields in the AFP data stream. Open the AFP dump file with an editor. Search for IMM with the find option. Repeat this operation to find all IMM structured fields in the data stream.

You can use the same PostScript example provided in 8.4, “Understanding the transform paper input tray allocation” on page 208.

If the conversion is correct, you must find three IMM structured fields in the AFP data stream as shown here (these statements are placed before the beginning of the page that requires a paper from a different input paper tray):

```
.386  - IMM (Invoke Medium Map)                        SFI 0010 D3ABCC 000000
        IMM: media map name = 'S1000000'

.403  - IMM (Invoke Medium Map)                        SFI 0010 D3ABCC 000000
        IMM: media map name = 'S2000000'

.420  - IMM (Invoke Medium Map)                        SFI 0010 D3ABCC 000000
        IMM: media map name = 'S3000000'
```

A short check using Table 29 for the Infoprint 40 PostScript driver shows that all trays were invoked correctly.

With this example, you can see that:

- Tray 1 (AFP) matches the IMM S1000000 (PostScript 1 priority 0).
- Tray 2 (AFP) matches the IMM S2000000 (PostScript auxiliary priority 1).
- Tray 3 (AFP) matches the IMM S3000000 (PostScript 2 priority 2).

Table 29. Infoprint 40 PostScript driver tray invocation

AFP input tray	Medium map	PostScript tray	Priority
1	S1000000	1	0 6 5 3 2 1 4
2	S2000000	Auxiliary	1 6 5 3 2 0 4
3	S3000000	2	2 6 5 3 0 1 4
4	S4000000	3	3 6 5 2 0 1 4
5	S5000000	Not available	Not available
6	S6000000	4	5 6 3 2 0 1 4
7	S0700000	5	6 5 3 2 0 1 4

Chapter 9. AFP Manager

This chapter provides an overview of AFP Manager included in Client Access Express V5R1. AFP Manager is a new graphical interface for managing and creating AFP resources on the iSeries server.

Many functions are detailed further in related chapters throughout this book. AFP Manager is used as a vehicle, other than green screens, for most of the new functions.

For more information, see:

- Chapter 10, “PSF configuration object and the iSeries server” on page 221
- Chapter 11, “Font mapping table” on page 237

An overview of AFP Manager is shown in Figure 136.

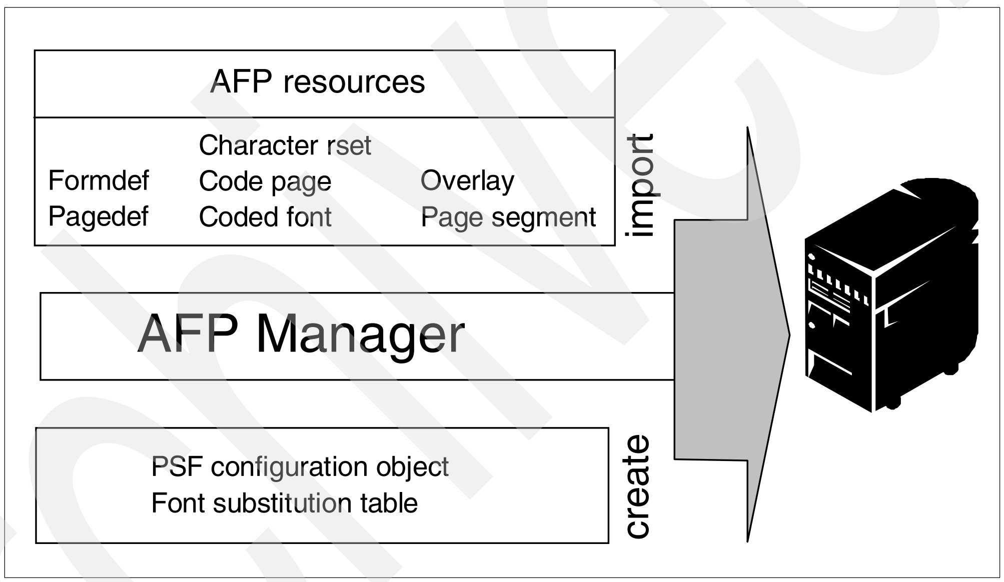


Figure 136. AFP Manager overview

Three different functions are available in AFP Manager:

- **Import AFP resources to the iSeries**: This function provides an easy way to transfer AFP resources from a PC, folder, or IFS to the iSeries server. Three different steps were required in the past. Now AFP Manager allows you to upload resources in only one step and create the iSeries AFP object in a library. The following resources are supported:
  - Code page
  - Coded font
  - Font character set
  - Form definition
  - Overlay
  - Page definition
  - Page segment

  Chapter 6, “Image file transform” on page 165, uses the import function to transfer page segment and overlays to the iSeries.

- **Graphical interface environment for creating and maintaining PSF configuration objects**: The PSF configuration object is used to manage Print Services Facility (PSF) functions on the iSeries server. New functions of InfoPrint Server for iSeries, such as PDF and e-mail, must be configured in the PSF configuration object under V5R1. Refer to Chapter 10, “PSF configuration object and the iSeries server” on page 221, for additional information.
- **Create and manage a Font substitution table**: This function is described in Chapter 11, “Font mapping table” on page 237.

The following sections describe how to install and start AFP Manager.

9.1 Importing AFP resources from a PC to the iSeries server

This section explains how to import AFP resources from a PC to the iSeries server. Figure 137 shows the AFP Manager window.

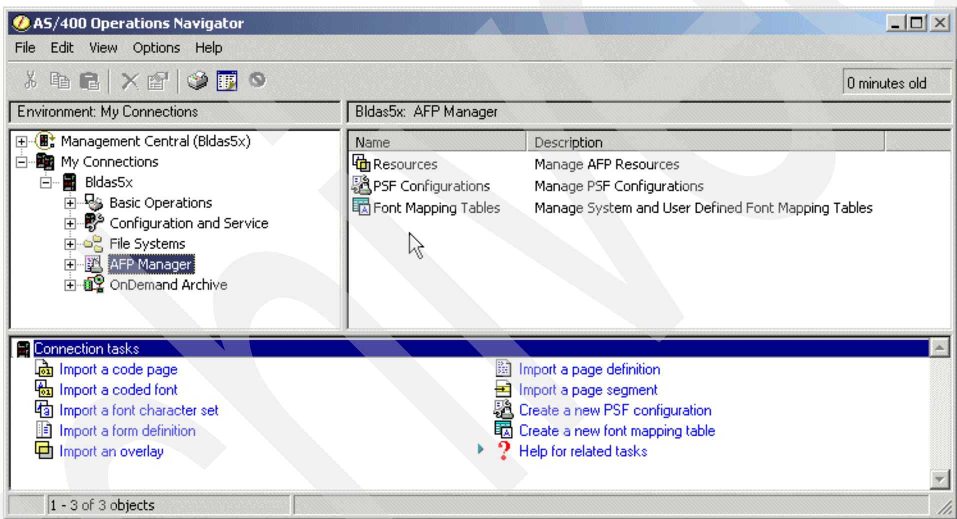


Figure 137. AFP Manager

The AFP Manager import function provides an upload function for AFP resources. Choose the corresponding import function for your resource. If your PC file does not match the requested format, no resource is created.

AFP Manager invokes the native command to upload the resource and create the iSeries object. The temporary files created for this process are deleted at the end of each upload request. Figure 138 shows the Import a Page Segment window.

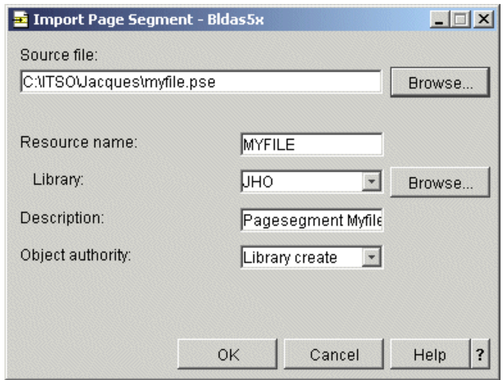


Figure 138. Import a Page Segment window

The fields in this window are explained in the following list:

- **Source file**: Specify the name of the file you want to import. Click **Browse...** to search for the file. The file must be in a supported format and correspond to the selected AFP resource.
- **Resource name**: Specify a name for the resource you want to create. The name cannot be more than eight characters long and must meet iSeries name requirements.
- **Library**: Specify the library where the new resource is located. You can store it in the current library or in a different library. To specify a different library, enter the name or click **Browse...** to search for a library. The library you specify must exist.
- **Description**: Enter a description of the resource you want to create. This parameter is recommended but not required. The description cannot contain more than 50 characters.
- **Library create**: When the library containing this object was created on the server using the Create Library (CRTLIB) command, a value was specified on the Create authority prompt (CRTAUT parameter). When you select Library create as the object authority, PSF/400 uses that value to determine the authority for this object. If the value specified on the Create authority prompt is changed, the new value does not affect any existing objects.

9.2 Installing AFP Manager

AFP Manager is a plug-in to Operations Navigator, a component of Client Access Express V5R1. You require AFP Manager to perform many of the tasks in this section. You can install AFP Manager using three methods:

- When you install Client Access Express, select **full install**.
- When you install Client Access Express, select **custom install**. From the Component Selection dialog, select **AFP Manager**.
- Add AFP Manager after installation.

The installation with the selective setup involves these steps:

1. Double-click the **Client Access Express** icon on your workspace to start. Figure 139 shows the Client Access Express window.

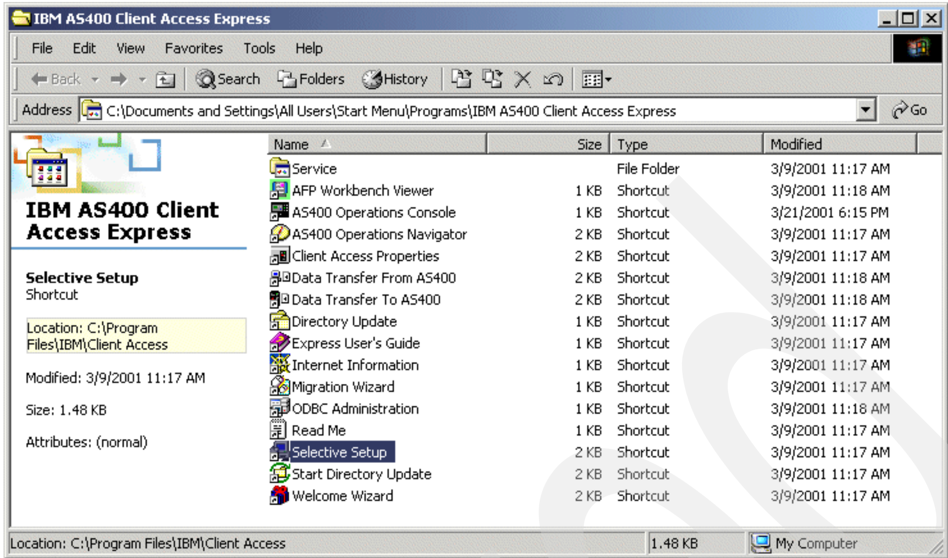


Figure 139. Client Access Express window

2. Select **Selective Setup** in the Client Access Express window. This command allows you to install components that were not initially installed. Figure 140 shows all installed and installable components. Use the rulers to scroll down through the different selections.

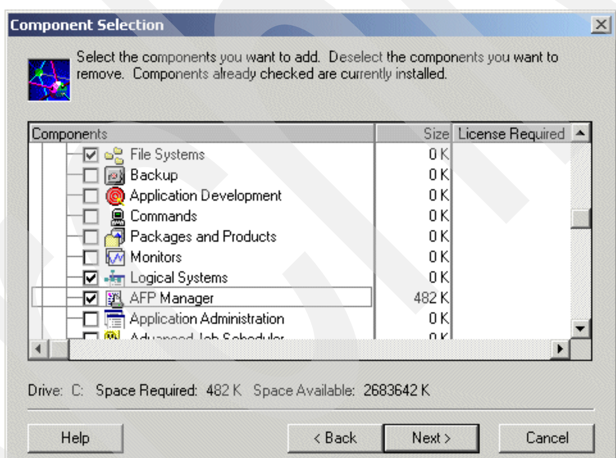


Figure 140. Component Selection

3. Select the AFP Manager component. The installation shield guides you through the installation. Complete the installation and start AFP Manager by following the instructions in 9.3, "Starting AFP Manager" on page 216.

9.3 Starting AFP Manager

This section explains how to start AFP Manager. Creating and managing the PSF configuration object is described in Chapter 10, "PSF configuration object and the iSeries server" on page 221.

1. Start Client Access Express. Select the corresponding icon on your desktop. The icon becomes available after you install AFP Manager.

The Client Access Express Window should appear like the example in Figure 139.

2. Double-click **AS400 Operations Navigator** to invoke it. The Operations Navigator window appears as shown in Figure 141.

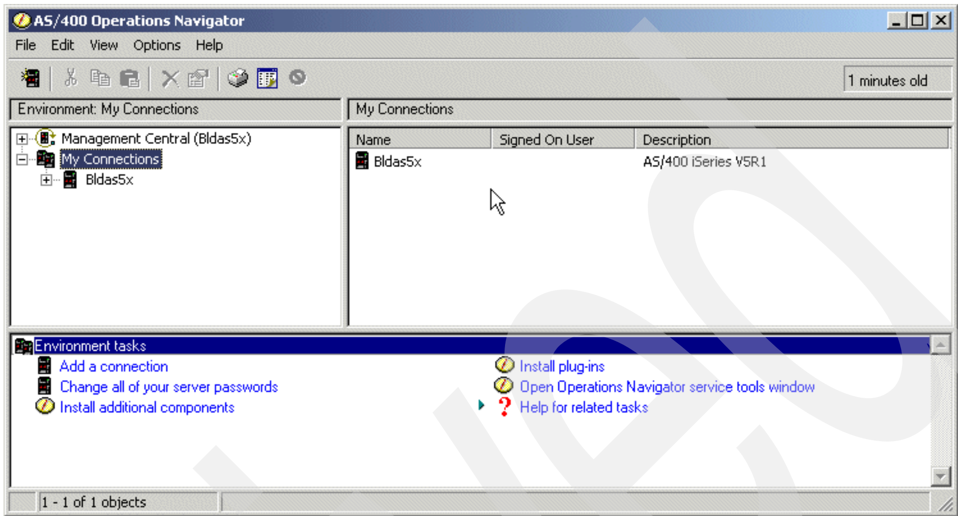


Figure 141. Operations Navigator window

3. Select the server connection with which you want to work. The number of systems shown in the windows depends on your environment.

   Figure 142 shows the Operations Navigator window with the available components.

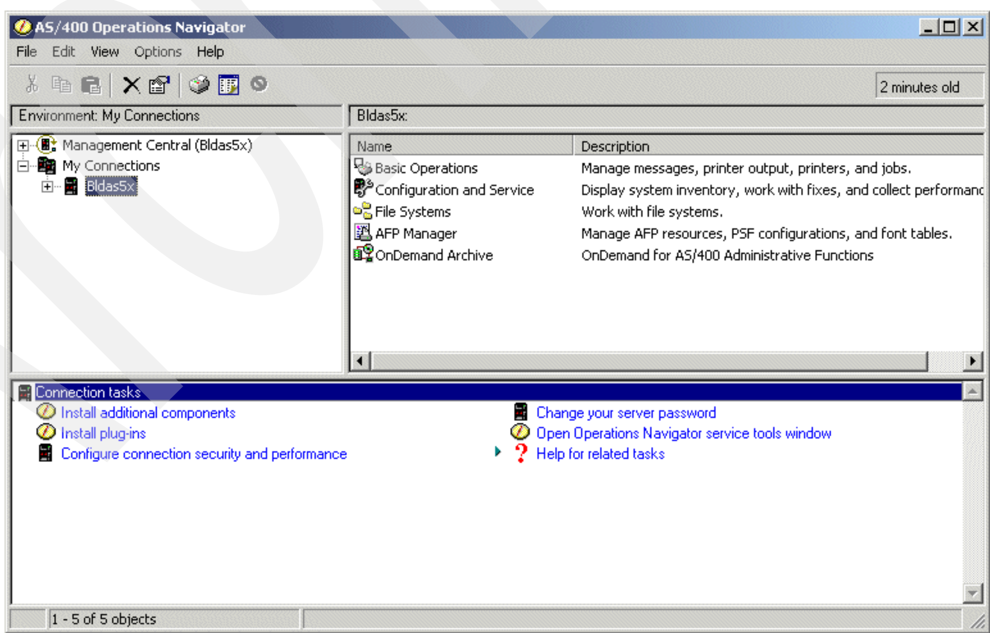


Figure 142. Selecting your system in Operations Navigator

4. Double-click **AFP Manager** to invoke it. Figure 143 shows AFP Manager.

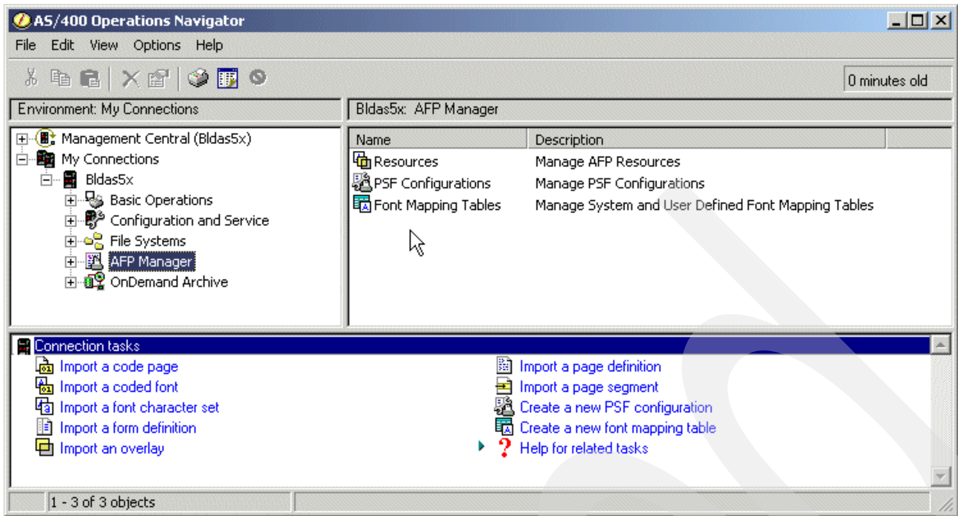


Figure 143. AFP Manager window

The AFP Manager provides the ability to import AFP resources from a PC to your iSeries server and to create specific AFP objects. The functions of the AFP Manager are used throughout this book or objects are created with the AFP Manager. Refer to the following sections for more information:

- Section 6.3.1, “Uploading your AFP resources with AFP Manager” on page 169
- Chapter 10, “PSF configuration object and the iSeries server” on page 221
- Chapter 11, “Font mapping table” on page 237

9.4 Displaying objects from other libraries

AFP Manager displays the user library by default. This section shows you how to select other libraries or use common iSeries Libraries lists:

1. Figure 144 shows the list of objects from the user library only. Select the **Options** menu and select the **Include** option.

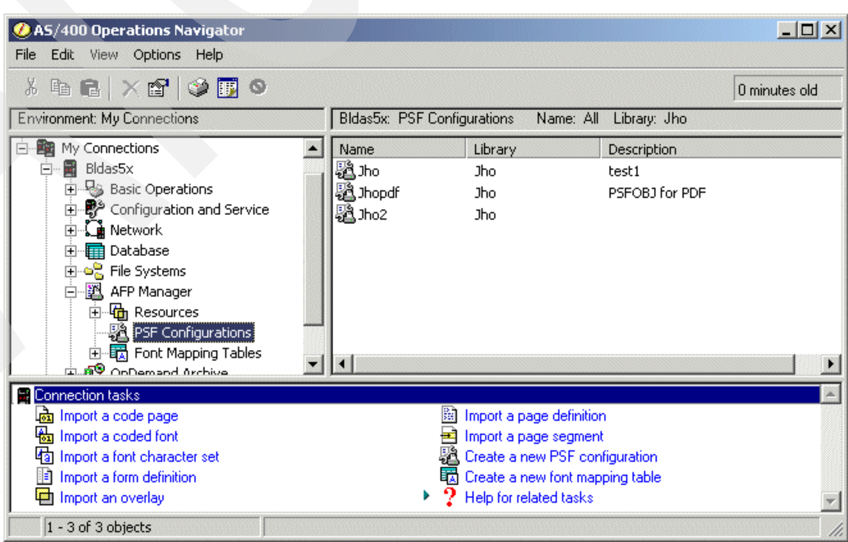


Figure 144. AFP Manager objects

2. The Include window appears as shown in Figure 145. Click the pull-down list for Library. Choose one selection and then confirm it by clicking **OK**.

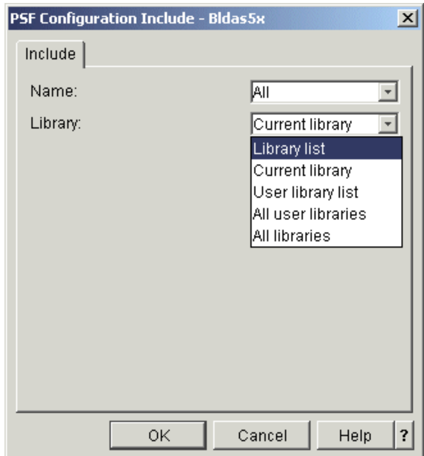


Figure 145. Include window

The new selection appears as shown in Figure 146.

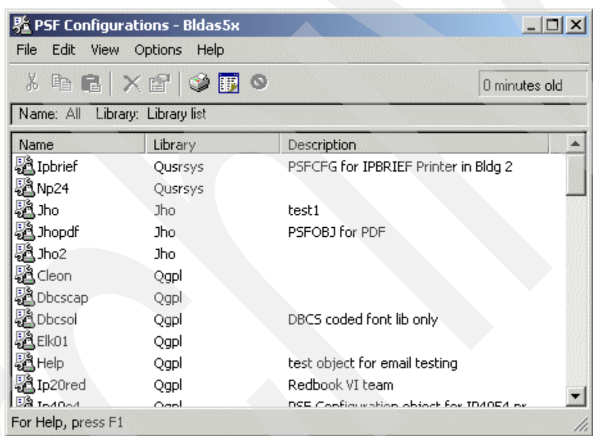


Figure 146. AFP Manager object list

Chapter 10. PSF configuration object and the iSeries server

This chapter describes the Print Service Facility (PSF) configuration object and the iSeries server. This object may be created when PSF/400 is used. Or it may be used for the AFP to PDF transform and the e-mail function of the Infoprint Server. The PSF configuration allows the use of specific options and capabilities from PSF. For more information about AFP software components and requirements, see Chapter 1, “iSeries printing software to create, manage output” on page 1.

The AFP Manager is used as a graphical interface to describe the related functions of the PSF configuration object. Information about installing and starting AFP Manager is provided in Chapter 9, “AFP Manager” on page 213.

10.1 Overview of the PSF configuration object

The PSF configuration object can be created in two different ways:

- Using the 5250 interface on the iSeries server (alternatively, AFP Manager may be used as part of Operations Navigator)
- Using AFP Manager described in 10.2, “Creating a PSF configuration object” on page 222

All functions are also available in the 5250 interface. The command language (CL) commands related to the PSF configuration object are:

CHGPSFCFG Change PSF Configuration
CRTPSFCFG Create PSF Configuration
DLTPSFCFG Delete PSF Configuration
DSPPSFCFG Display PSF Configuration
WRKPSFCFG Work with PSF Configurations

New functions have been implemented to take advantage of the Infoprint Server. The functions of the PSF configuration object can be grouped as shown in the following list:

- Resource related functions
- Communication, sharing, and recovery functions
- PDF transform functions
- Advanced PSF functions

PDF transform is part of the Infoprint Server but does not require PSF/400. You can create a PSF configuration object to use the PDF transform. See Chapter 4, “Using the Infoprint Server for iSeries PDF transform” on page 103, for more information.

Figure 147 shows the iSeries print process with PSF/400, the PSF configuration object, and the font substitution table.

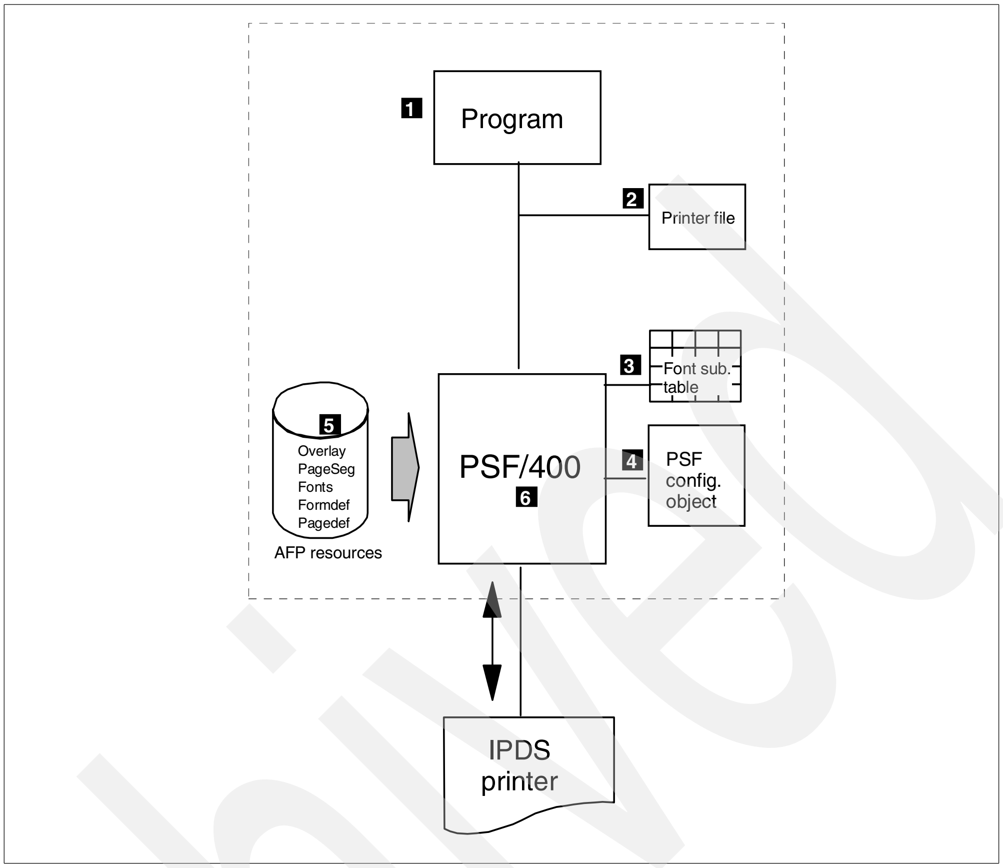


Figure 147. PSF configuration object

The numbers in the following list explain and correspond to those in Figure 147:

1. This is the print program or application that produces the data.
2. This is the printer file used by the program to format the spooled file.
3. A font substitution table is an object that defines how the substitution of missing font resources is performed. Chapter 11, “Font mapping table” on page 237, explains how to work with this object.
4. The PSF configuration object provides information about the printer, communication, and resource handling to PSF.
5. AFP resources are used to compose the print document. These resources may be overlays or electronic forms, a page segment or image, fonts, or form definitions (Formdef) and page definitions (Pagedef) as a formatting alternative to Data Description Specification (DDS).
6. PSF/400 composes the IPDS data stream and manages the dialogue with the printer.

10.2 Creating a PSF configuration object

The PSF configuration object may be used to set parameters required by a specific customer environment. These parameters are required to optimize printing functions and the dialog between PSF/400 and IPDS printers.

You should use AFP Manager to define and manage the PSF configuration object. Chapter 9, “AFP Manager” on page 213, guides you through installing and starting AFP Manager.

The 5250 interface and AFP Manager use the same default parameters. The parameters are not grouped the same way, but all functions and parameters are the same.

Figure 148 shows starting AFP Manager from Operations Navigator.

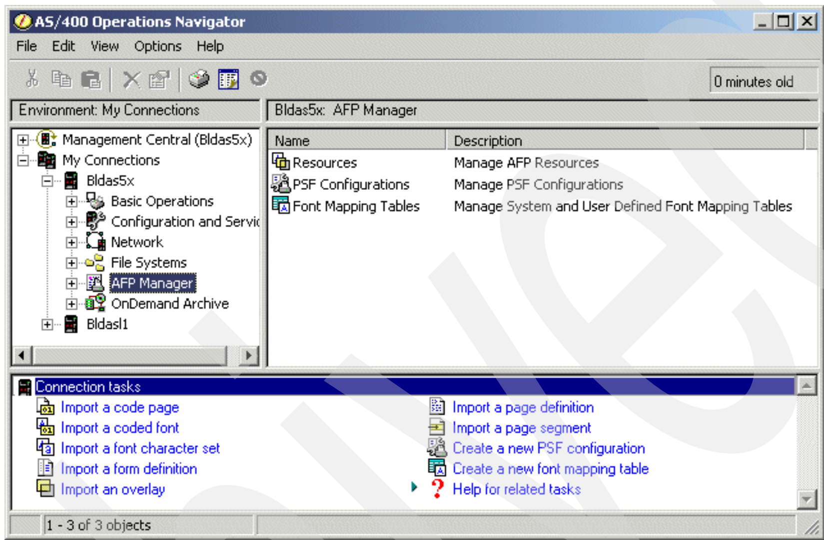


Figure 148. AFP Manager

You can now create or manage the PSF configuration objects. From the bottom panel shown in Figure 148, you can select **Create a new PSF configuration** toward the bottom. Or you can choose **PSF Configurations** in the top-right panel. Double-click **PSF Configurations** to view your PSF configurations objects.

AFP Manager provides the same default as the 5250 interface on the iSeries server. Figure 149 shows the General configuration tab.

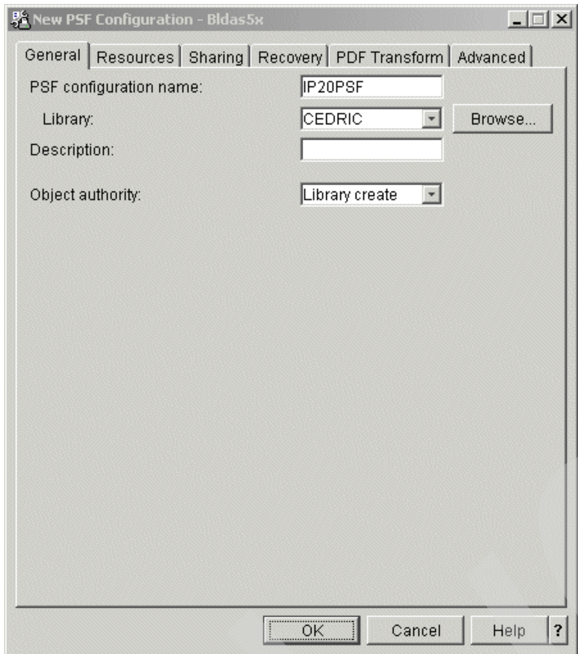


Figure 149. PSF configuration object: General tab

Specify a name for the PSF configuration object (it can be up to 10 characters long and must meet the iSeries name requirements):

- **Library**: Specify the library where the PSF configuration object is created. It can be created in the current library or a different library. To specify a different library, enter the name or click **Browse...** to search for a library. The library you specify must exist.
- **Description**: Specify a description of the PSF configuration object. This is optional (but recommended). The description cannot contain more than 50 characters.
- **Object authority**: Specify the authority you want to give to users and ensure it meets the following criteria:
  - They do not have specific authority to the object.
  - They are not on an authorization list.
  - Their group profile has no specific authority to the object.

> **Note**
>
> You cannot change the authority for a PSF configuration object after it has been created.

10.2.1 Resource configuration

Figure 150 shows the Resources configuration tab.

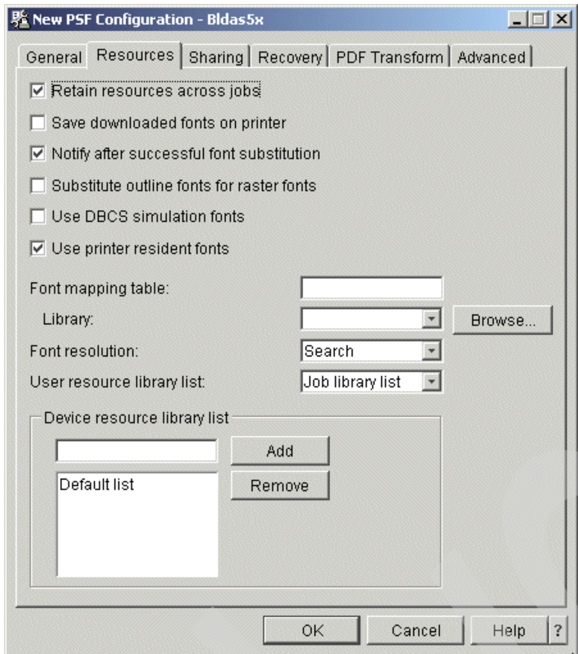


Figure 150. PSF configuration object: Resources tab

The options on this tab are explained here:

- **Retain resources across jobs**: Select this option if you want to retain page segments and overlays across printer output file boundaries in the printer. This allows different printer output files to use the same resource without downloading them multiple times. These resources are purged from the printer when the printer writer is ended.

  If this is not selected, page segments and overlays are deleted after every printer output file finishes printing.

  Note

  By default, code pages and font character sets are retained in the printer across printer output file boundaries.

- **Save downloaded fonts on printer**: Select this if you want to activate font capturing for eligible host downloaded fonts on the printer. Captured fonts are not purged from the printer when the printer writer is ended. This option may be very useful for reducing network traffic. If you plan to use font capture, we recommend that you have enough printer memory. This option may not reduce print performance for a single byte character set.

  Note

  Because improper use of font capture may cause unpredictable results, only system administrators should handle the font capture feature.

- **Notify after successful font substitution**: Select this if you want PSF/400 to issue messages indicating that a successful font substitution was performed.

- **Substitute outline fonts for raster fonts**: Select this if you want downloadable AFP raster font character sets to be replaced with equivalent downloadable outline fonts when possible. This allows you to use outline fonts to print applications that use raster fonts without changing the application or changing its appearance. Outline fonts are scalable so you don't have to store a different font character set for each point size on your system, and you don't have to send a different font to the printer for every change in point size. This increases your system storage space and enhances printing performance.

  You may want to pay attention to our customized fonts. Customized fonts may be replaced by an outline font without your specific modifications.

- **Use DBCS simulation fonts**: Select this if you want PSF/400 to use DBCS simulation fonts instead of old DBCS raster fonts on printers using this PSF configuration object. DBCS simulation fonts are outline fonts that are positioned like the old raster fonts. This allows you to use outline fonts to print applications that use the old DBCS raster fonts without changing the application or changing its appearance. Outline fonts are scalable so you don't have to store a different font character set for each point size on your system, and you don't have to send a different font to the printer for every change in point size. This increases your system storage space and enhances printing performance.

  If you select this, and your system does not have DBCS simulation fonts, PSF/400 searches for the simulation font. It then searches for the raster font. This extends printing time.

  Note

  The DBCS simulation fonts should only be used for an old application. Never mix the old DBCS raster fonts with the new DBCS Core raster or outline fonts since they are positioned differently.

- **Font mapping table**: Specify the name of a printer-resident to printer-resident (PPF) font table. This printer-resident font table is used by PSF/400 when printing to a printer that supports printer-resident fonts and the print job specifies a printer-resident font that is not supported by the printer.

  This parameter is optional. If you don't specify a PPFCS font table, PSF/400 maps fonts with its internal tables as necessary.

- **Font resolution**: Specify the resolution that PSF/400 should use to print the printer output file when the following conditions are met:
  - The printer supports multiple resolutions.
  - The printer is configured to report support of multiple resolutions.

    Either the printer output file does not specify the font metrics and resolution with which to print the file or the font is not available at that resolution.

    Click the drop-down list to select one of the following options:

    - **Search**: Search the library list for the first occurrence of a host font with a name match. The resolution of that font is used to print the file.
    - **240**: The font resolution used to print the file should be 240 pels per inch.

- **300**: The font resolution used to print the file should be 300 pels per inch.

> **Note**
>
> If the printer is configured to report support of either 240 pels per inch or 300 pels per inch only, this value is ignored.

- **User resource library list**: Specify the user resource library list to be used when searching for AFP resources. When an AFP resource is specified with a printer output file, PSF/400 first searches the user resource library list and then searches the device resource library list for the resource.
- **Device resource library list**: Specify the device resource library list to be used when searching for AFP resources. When an AFP resource is specified in a printer output file, PSF/400 first searches the user resource library list and then searches the device resource library list for the resource.

  You can use the default list or make your own list. To make your own list, specify a library and click **Add**. Your list is used instead of the default list. You can remove libraries from your custom list by selecting the library name and clicking **Delete**.

You can restore the default list by deleting all of the libraries from your list.

If you choose the default list, all of the following libraries (if present on the system) are searched for AFP resources, including:

- **QFNTCPL**
- **QFNT01 - QFNT19**
- **QFNT61 - QFNT69**

> **Note**
>
> If all the system libraries in this list have not been created, you can create libraries with the name of the missing system libraries. You can then place resources in these libraries that can subsequently be found by other users. Alternatively, the system administrator can create all of the missing system libraries with PUBLIC *USE authority.

10.2.2 Sharing options

Consider the sharing option if your printer must be used from different systems. As soon as a PSF uses a printer, no other system can print on the system. The print writer must be ended to finish a connection if no sharing is defined.

Figure 151 shows the Sharing tab on the PSF configuration object.

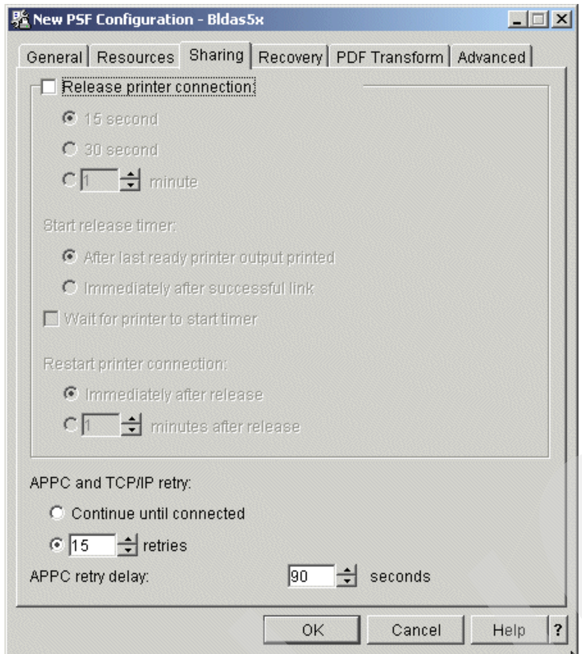


Figure 151. PSF configuration object: Sharing tab

The Sharing tab supports these options:

- **Release printer connection**: These options allow you to control printer release. After selecting Release printer connection, you can control:
  - When to start the release timer
  - The duration of the release timer
  - How long to wait after the printer is released before trying to establish a new session or dialog

  When the printer is released, PSF/400 does not terminate, but releases the connection with the printer. A printer output file that has started printing before the time is up finishes before the printer is released. Select the 15 seconds or 30 seconds option, or specify up to 1,440 minutes.

- **Start release timer**: These controls let you specify when to start the release timer. The value you specify under Release printer connection determines the length of time the writer keeps the printer before releasing the session. Releasing the session does not terminate PSF. It allows another PSF to start a session to the printer. You can choose one of the following options:
  - *After last ready printer output printed*: Select this to start the release timer after your last ready printer output file has been printed. If you select this, and a new printer output file becomes ready before the session is released, the timer is canceled. The timer starts again when there are no more ready printer output files.
  - *Immediately after successful link*: Select this to start the release timer immediately after a successful link to the printer has been made. Use this value when you want the writer to time-share the printer. If you select this, and a new printer output file becomes ready before the session is released, the file is printed but the timer is not canceled.
  - *Wait for printer to start timer*: Select this to wait for an indication from the printer before starting the release timer. This option works in conjunction

with one of the other two choices. The timer is started after the writer receives an indication from the printer to release the IPDS dialog and one of the above requirements is met.

> **Note**
>
> This option is only available on IBM printer 3130 (Release 2). If the writer detects that the printer is not capable of controlling the IPDS dialog, this option is ignored.

Consider these examples. If you select Immediately after successful link and 60 minute timer, the printer releases 60 minutes after a connection has been made. Any documents you start to print in that time are printed. The printer is released after 60 minutes or after it finishes printing any ready printer output files it has started (whichever is longer).

If you select After last ready printer output printed and 60 minute timer, the printer is released 60 minutes after the last ready printer output file prints. If some new printer output files become ready after the timer has started, but before the 60 minutes have passed, the new files are printed and the timer is stopped. After the last file has printed, the 60-minute timer is restarted.

> **Note**
>
> Only printers and devices attached to a server using APPC or TCP/IP support these options. The IBM 3130, Release 2, with a Twinaxial connection supports these options if Wait for printer to start timer is selected.

- **Restart printer connection**: These controls allow you to specify how long the printer writer should wait before trying to establish a new session or dialog after the printer is released. You can select one of the following options:
  - *Immediately after release*: Select this to try to establish a new session or dialog as soon as a printer output file is ready.
  - *Specified number of minutes after release*: Select this to specify a number of minutes to wait before the printer writer tries to establish a new session or dialog. Specify the number of minutes or click the arrows to select the number of minutes. You can specify, at most, 1,440 minutes.

  Consider this example. If you specify 60 minutes after release, the writer looks for a printer output file 60 minutes after the printer is released. If there is one ready, the writer tries to establish a session. If there is no output file ready, the writer keeps looking for one periodically. When one is found, the writer tries to establish a session with the printer.
- **APPC and TCP/IP retry**: These options allow you to control APPC and TCP/IP retries after PSF/400 receives notification that a session start request has failed:
  - To control the number of times PSF/400 tries to start a session, select one of the following options:
    - Continue until connected: Select this if you want to try to establish a connection until one is made.

- Specified number of retries: Select this to specify the number to times to try to establish a connection with the printer if the first try fails. Specify the number or click the arrows to select the number of retries. You can specify at most 99 retries.

Note

These parameters only apply to printers and devices attached to a server using APPC or TCP/IP.

– To control the amount of time you want PSF/400 to pause after it receives notification that a session start request has failed, specify the desired number of seconds after APPC retry delay. After the specified time has elapsed, another session start request is issued. You may specify between 0 and 999 seconds.

10.2.3 Recovery option

These functions allows PSF to recover an inactive session and determine which part of the job has proceeded. Figure 152 shows the different options available on the Recovery tab.

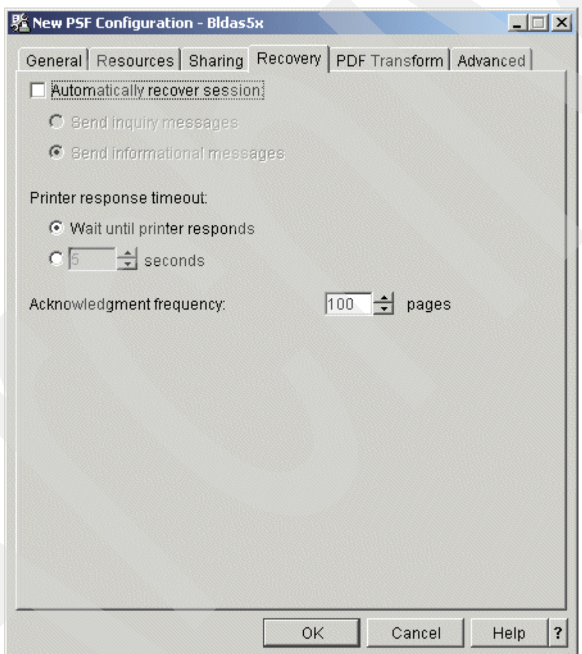


Figure 152. PSF configuration object: Recovery tab

The features on this tab are explained here:

- **Automatically recover session**: These options allow you to control session recovery. Select this option to have PSF try to resume printing when a session has been unexpectedly ended by a device. A message is sent to the message queue when PSF/400 performs automatic session recovery. You can choose one of the following options:

– *Send inquiry messages*: This message allows you to specify the page number from which a writer should begin printing the output file that was being processed when the session was ended.

– *Send informational messages*: This message informs you that PSF/400 is performing automatic session recovery.

> **Note**
>
> These parameters only apply to printers and devices attached to a server using APPC or TCP/IP.

If you are using an APPC-attached printer or device, you must use an APPC controller description and an APPC device description, both of which specify APPN(*YES) on your server. In addition, the APPC controller description must specify MINSWTSTS(*VRYONPND).

- **Printer response timer**: These options allow you to control how long to wait for a response from a TCP/IP-attached printer. You can choose one of the following options:
  - *Wait until printer responds*: Select this if you want the printer writer to wait for a response from the printer until one is received. If the writer does not receive a message, it is never ended. To avoid this, you may want to use the other option.
  - *Specified number of seconds*: Select this option to specify how long the printer writer should wait for a response from the printer. Specify the number, or click the arrows, to select the number of seconds. You may specify between 5 and 3,600 seconds. The writer is ended if the printer does not respond within the specified amount of time. If this happens, a message is sent to the message queue associated with the writer.
- **Acknowledgment frequency**: Specify how often you want PSF/400 to send IPDS acknowledgment requests to a printer. The printer responses contain information about the status of pages sent to the printer.

  Acknowledgment frequency is supported on all attachments. You may specify between 1 and 32,767 pages. Specify the number, or click the arrows, to select the number of pages.

  If a printer output file contains fewer pages than you specified, an acknowledgment is requested after the last page of the file is sent.

  When a connection with a printer is abnormally ended, PSF/400 may reprint pages because the printer could not return the status of some pages that were printed. If you send acknowledgment requests more often, the number of pages that may be reprinted can be decreased. For example, if you specified 100 as the acknowledgment frequency, and the job fails on page 156, PSF/400 does not know that the last 56 pages already printed and then printed again. In the same situation, if you specified 50 as the acknowledgment frequency, PSF/400 knows that the first 150 pages printed and only prints the last six pages again. However, if you request acknowledgments too often, such as one per page, you may notice slower performance.

10.2.4 PDF transform options

The Infoprint Server for iSeries provides a new transform capability. The AFP to PDF transform is part of the Infoprint Server. The options for the AFP to PDF transform have been added in the PSF configuration object, including:

- **Generate PDF**: Controls allow you to generate one or multiple PDF files and store the file or send it. Select Generate PDF if you want the printer output converted to PDF. You can do one of the following tasks with the PDF files:
  - *Store as printer output*: Select this if you want to store the PDF output as a printable file. You can specify the output queue on which to store the file.
  - *Store as stream file*: Select this if you want to store the PDF output as a stream file. This allows you to electronically mail the file or use the file in an Internet based application, such as a soft copy repository. You can specify the path where you want the PDF output to be stored.
  - *Send as electronic mail*: Select this if you want PSF/400 to send the PDF file as e-mail. You can specify who sends the e-mail and which e-mail mapping program to use.

  Along with the above options, you can choose to generate multiple PDF files. Select this if you want to split the PDF output into multiple PDF files at each start group boundary (DDS keyword STRPAGGRP) in the data stream when applicable. If you select this option, and there is at least one page group but some data is not in a page group, you receive an error and the PSF configuration object is not created. If you select this option, and there are no page groups, one PDF file is created.

> **Note**
>
> You must have Infoprint Server for OS/400 to generate PDF files.

Figure 153 shows the PDF Transform tab.

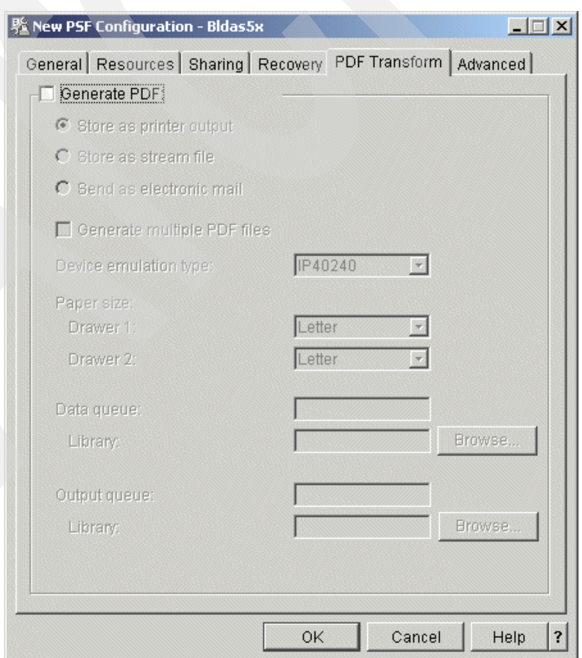


Figure 153. PSF configuration object: PDF Transform tab

The options on this page are explained here:

- **Device emulation type**: Specify the type of device that the virtual printer associated with the IPDS to PDF transform should emulate. Click the drop-down list to select one of the following options:

  IP40240 The virtual printer should emulate an IP40 printer configured at 240 pels per inch resolution.

  IP40300 The virtual printer should emulate an IP40 printer configured at 300 pels per inch resolution.

  4028 The virtual printer should emulate a 4028 printer.

  3812 The virtual printer should emulate a 3812 printer.

- **Paper size**: Specify the size of paper in drawers one and two of the device associated with the IPDS to PDF transform. This should be the device specified in Device emulation type. Click the drop-down list and choose the option you need.
- **Data queue**: Specify the name of the data queue where PSF/400 logs the IPDS to PDF transformation completion notifications. This is optional. However, if you specify a data queue, you must specify the library in which it is contained. The data queue must exist when you use this PSF configuration object. If the data queue you specify does not exist at that time, you cannot call the writer that uses this PSF configuration object. If the data queue you specify is damaged or full, the completion notifications are not logged.
- **Library**: Specify the library where the data queue is located. Click **Browse...** to search for one. If the library you specify does not exist when you try to use the PSF configuration object, you cannot call the writer that calls this PSF configuration object.
- **Store as printer output**: Use this parameter to store PDF output as a printable file on the specified output queue. You must specify an output queue.
- **Output queue**: Specify the name of the output queue that should be used for storing the PDF output. The output queue must exist when this PSF configuration object is used. If the output queue does not exist when you try to use the PSF configuration object, you cannot call the writer that uses this PSF configuration object.
- **Library**: Specify the library where the output queue is located. Click **Browse...** to search for one. The library you specify must contain the output queue when this PSF configuration object is used. If the library does not exist when you try to use the PSF configuration object, you cannot call the writer that uses this PSF configuration object.

10.2.5 PSF configuration object advanced options

The following list explains the PSF configuration object advanced options:

- **Allow IPDS and SCS files to bypass conversion**: Before printing, SCS files are converted to IPDS files. Then they are converted to AFPDS and then back to IPDS. IPDS files are also converted to AFPDS and then converted back to IPDS. The final IPDS files have specific commands for your printer.

  You can choose to skip the conversion to AFPDS and back to IPDS. This is called *IPDS pass-through*. Select this option if you want to allow IPDS pass-through for eligible files configured with this PSF configuration object.

Not all SCS or IPDS printer output files are eligible for IPDS pass-through. Some files may contain special functions that require transform to AFPDS for correct printing. Specifying IPDS pass-through allows only those printer output files eligible for IPDS pass-through to bypass the extra transforms.

IPDS pass-through is not valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts cannot support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts that are downloaded to the printer. Distributed Print Function (DPF) is a function supported by the PSF for OS/2 print server that blocks the use of printer-resident fonts. Therefore, no DPF-attached printer supports IPDS pass-through.

Any printer that supports resident fonts can support IPDS pass-through. See the documentation provided with your printer to determine if your printer supports resident fonts (and therefore IPDS pass-through).

Note

Selecting this option speeds up printing. You should not skip the conversion unless you know that the original IPDS stream works with your printer.

Figure 154 shows the Advanced functions of the PSF configuration object.

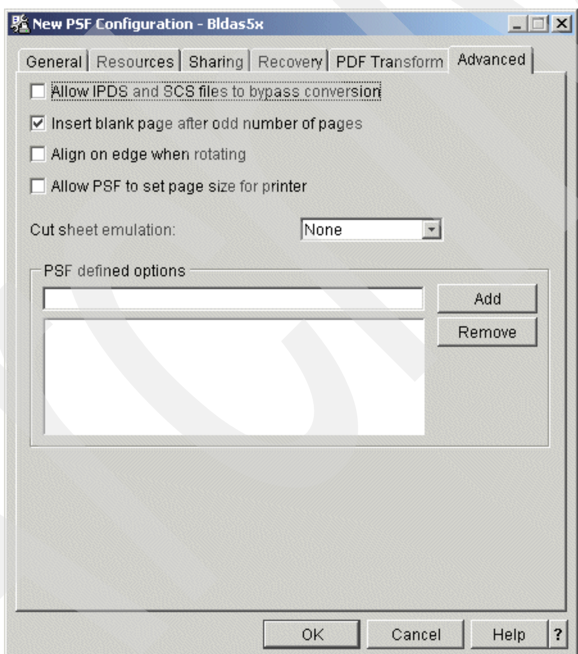


Figure 154. PSF configuration object: Advanced tab

These functions are explained in the following list:

- **Insert blank page after odd number of pages**: Select this if you want PSF/400 to issue a blank page after every separator page and printer output file copy that contains an odd number of pages. The blank pages assure that the first page of one file is not printed on the back of the last page of another file. If you are going to use bursting, you should select this option.

> **Note**
>
> This parameter only applies to continuous forms printers. See the documentation provided with your printer to determine if your printer supports this feature.

- **Align on edge when rotating**: When the page rotation value of a printer output file is *COR or *AUTO on the server, and the server rotates the output, the output is normally rotated 90 degrees.

  Select this option when rotating to spin the output 270 degrees instead of 90.

- **Allow PSF to set page size for printer**: Select this if you want PSF/400 to set the page size (forms) in the printer.

> **Note**
>
> This parameter does not apply to all printers. See the documentation provided with your printer to determine if your printer supports this feature.

- **Cut sheet emulation**: Specify how frequently PSF/400 checks the document page to determine if it fits on half the continuous forms page when using cut sheet emulation. If PSF/400 determines that the page does not fit properly, the cut sheet emulation mode is ended. Click the drop-down list to select one of the following options:
  - *None*: PSF/400 does not verify that the document page fits on half the continuous forms physical page.
  - *Check first page*: The first page of each copy group is checked to determine if the page fits on half the continuous forms page.
  - *Check front side*: Each front side page is checked to determine if the page fits on half the continuous forms page.

- **PSF defined options**: These parameters allow you to add and remove PSF defined options. One or more options may be made available between releases of OS/400. You can add and remove these options. The list shows your current PSF defined options. If the option you specify is not valid, PSF/400 ignores the option.

  To add an option, specify the value as defined by IBM and click **Add**. The new option appears in the list. You can have a maximum of six PSF defined options at a time.

  To delete an option, select it from the list and click **Remove**.

Chapter 11. Font mapping table

A font substitution is applied when a font resource is not found or does not match the printer characteristics. Font mapping tables allow you to manage the way font substitution is provided on the iSeries server. The new printer-resident to printer-resident font character set (PPFCS) font mapping table will help many customers resolve font substitution problems.

This chapter shows you how to create and manage font mapping tables with the new AFP Manager and how to work with new table type provided since V4R4M0. All these functions are available with the 5250 interface. To find commands related to the font mapping table on the 5250 interface, type:

```
GO CMDFNTTBL
```

Using font mapping tables requires you to have PSF/400 installed. PSF/400 manages font requirements from an AFP point of view.

11.1 Introduction to font mapping table types

The following section describe the different font mapping table types and how they are invoked in the print process.

All font mapping table types may be either:

- **System tables**: Define how substitution is applied on the iSeries. System tables cannot be modified by a user.
- **User tables**: Provide substitution for a device or a group of devices. The PSF configuration object allows you to define a specific font mapping table. These are searched before the existing system tables.

Font tables are used by the iSeries as soon as a font request cannot be completed. This can append when resources and fonts are not available or do not match the printer resolution.

The substitution types that are provided are described in the following sections.

11.1.1 Printer-resident to host-resident font character set (PHFCS)

A PHFCS font mapping table is used when the application references printer-resident fonts and the printer being used does not support them. Some of these printers are the 3827, 3825, 3820, and 3900 Model 1. PSF/400 must map the printer-resident font to a host-resident font and download the host-resident font to the printer.

When performing printer to host mapping, PSF/400 first looks in QUSRSYS for the appropriate table QPHFCS. If it does not find one, it uses the system table QSYSPHFCS. The system font mapping tables are supplied by PSF/400.

You can create, change, delete, open, and rename a PHFCS font mapping table. Only the user PHFCS font mapping table placed in QUSRSYS is active.

Internal font resources of a printer may be disabled using the “Use printer resident fonts” option of the PSF configuration object.

11.1.2 Host-resident to printer-resident font character set (HPFCS)

A HPFCS font mapping table is used when the application references host-resident fonts and the printer being used does not support them. Some of these printers are the 4224, 4234, 4230, and 64xx. PSF/400 must map the host-resident font to a printer-resident font and use the printer-resident font.

When performing host to printer mapping, PSF/400 first looks in QUSRSYS for the appropriate table (QPFCS, for example). If it does not find one, it uses a system table (QSYSPFCS, for example). The system font mapping tables are supplied by PSF/400. If you want to change them, you must create your own.

You can create, change, delete, open, and rename an HPFCS font mapping table. Only the user HPFCS font mapping table placed in QUSRSYS is active.

11.1.3 Printer-resident to host-resident code page font (PHCP)

A PHCP font mapping table is used when the application references printer-resident code pages and the printer being used does not support them. Some of these printers are the IBM 3827, IBM 3825, IBM 3820, and IBM 3900 Model 1. PSF/400 must map the printer-resident code page to a host-resident code page and then download the host-resident code page to the printer.

When performing printer to host mapping, PSF/400 first looks in QUSRSYS for the appropriate table (QPHCP, for example). If it does not find one, it uses a system table (QSYSPHCP, for example). The system font mapping tables are supplied by PSF/400. If you want to change them, you must create your own.

You can create, change, delete, open, and rename a PHCP font mapping table. Only the user PHCP font mapping table placed in QUSRSYS is active.

11.1.4 Host-resident to printer-resident code page (HPCP)

A HPCP font mapping table is used when the application references host-resident code pages and the printer being used does not support them. Some of these printers are the 4224, 4234, 4230, and 64**. PSF/400 must map the host-resident code page to a printer-resident code page and use the printer-resident code page.

When performing host to printer mapping, PSF/400 first looks in QUSRSYS for the appropriate table (QHPCP, for example). If it does not find one, it uses a system table (QSYSHPCP, for example). The system font mapping tables are supplied by PSF/400. If you want to change them, you must create your own.

You can create, change, delete, open, and rename an HPCP font mapping table. Only the user HPCP font mapping table placed in QUSRSYS is active.

11.1.5 Printer-resident to printer-resident font character set (PPFCS)

A PPFCS font mapping table is used *only if the printer-resident font* specified in the print job *is not supported by the printer*. In this situation, the printer-resident to printer-resident font mapping tables are searched in the following order:

- If a matching entry is found in the table and the substitute font is supported by the printer, then the specified substitute font is used.

- If a matching entry is not found or if the specified substitute font is not supported by the printer, then the system uses its internal system font mapping tables to perform the font substitution.

Several PPFCSes may be created. Each PPFCS font mapping table is linked to a printer or a group of printer using the Font Mapping Table parameter of the PSF configuration object.

With the default printer resident font substitution table, if a specific printer resident font (FGID) is requested in the application but is not supported by the designated printer, or cannot be downloaded from OS/400, it will be substituted for a different printer resident font. Appendix D of *OS/400 Printer Device Programming*, SC41-5713, documents the various font substitution tables.

For example, if you look at Table D2 in that appendix, if FGID 204 (Matrix Gothic, 13 CPI) was selected in the spooled file and the appointed printer is an IBM Infoprint 32, this font is substituted for FGID 223 (Courier, 15 CPI). Therefore, if this substituted font is not desirable, it can be mapped to a different FGID that is present on the printer, such as FGID 203 (Gothic Text, 13 CPI).

11.1.6 Font mapping table flow

Figure 155 shows the relationship between the different system elements and the font mapping table.

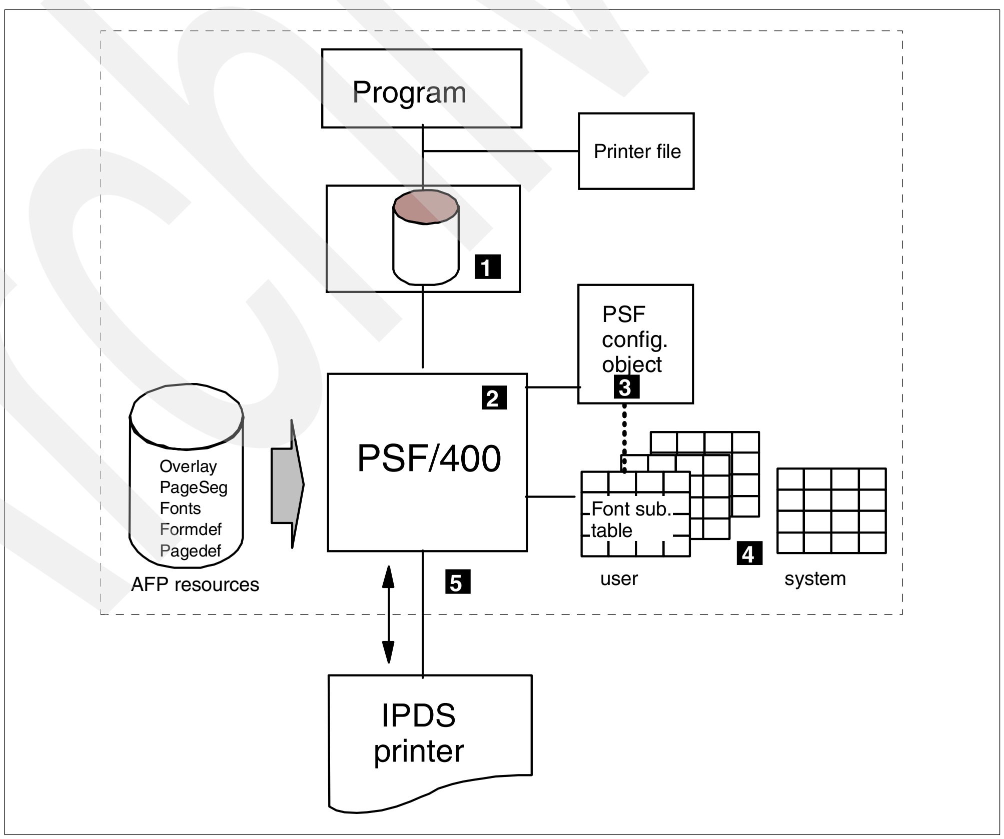


Figure 155. Font substitution table flow

The numbers in the following list correspond to and explain the numbers shown in Figure 155:

1. This is the print program or application that produces the data.
2. Spooled are placed in the output queue.
3. PSF/400 manages the print process regarding the parameter in the PSF configuration object. A font substitution is applied when a font resource is not found or does not match the printer characteristics. Font mapping tables allow you to manage the way font substitution is provided on the iSeries.
4. The PSF configuration object parameter “Font Table” is used to assign a printer-resident to printer-resident font character set substitution table to a printer or a group of printer.
5. A font substitution table is an object that defines how the substitution of missing font resources is performed.
   - System font substitution tables are used if no user table has been created.
   - The following user tables are placed in QUSRSYS. These tables may not be defined in the PSF configuration object and are the unique object that PSF takes under consideration:
     - Printer-resident to host-resident font character
     - Host-resident to printer-resident font character
     - Printer-resident to host-resident code page
     - Host-resident to printer-resident code page
     - Printer-resident to printer-resident font character set is the only table that can be defined in the “Font Table” parameter in the PSF configuration object. This table may be stored in a user library.
6. PSF/400 composes the IPDS data stream and manages the dialogue with the printer. If a font resource is not found or does not match the printer characteristic, a font substitution is applied. PSF/400 searches for a valid substitution in the following sequence:
   a. *User font substitution table*: PSF/400 search for a valid substitution regarding the substitution type and the substitution defined by the user. If a valid substitution is found, it is applied. If no valid substitution is found, the value of the system font substitution table is applied.
   b. *System font substitutions*: These are applied if no user table is found or no valid substitution has been defined.

11.2 Using AFP Manager to manage your font mapping table

AFP Manager may be used to display the system font substitution table or to create a user font substitution table. AFP Manager has been discussed in several parts of this book to manage, create, or import AFP resources. You can find additional information in Chapter 10, “AFP Manager” on page 195.

Start Operations Navigator and then AFP Manager. Select **Font Mapping Table** as shown in Figure 156 to manage font mapping tables.

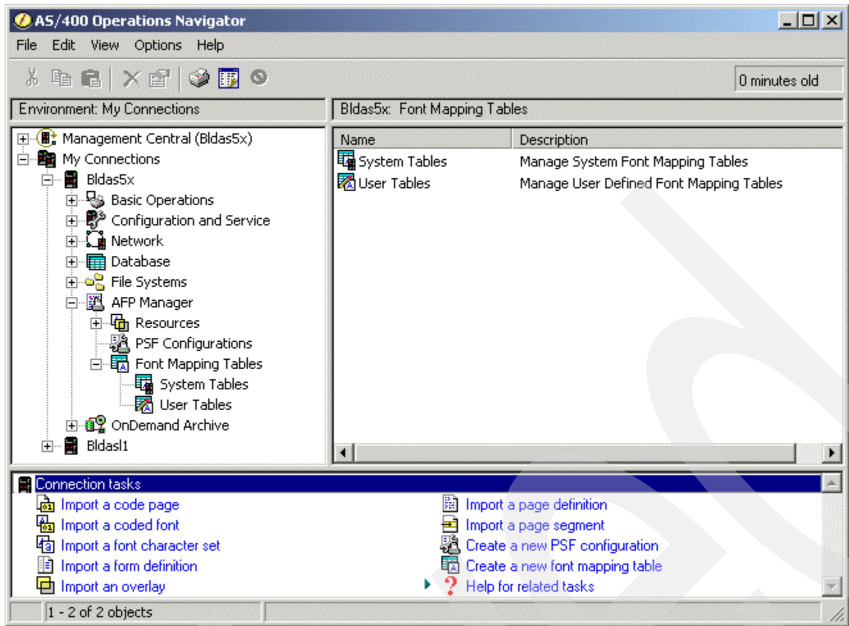


Figure 156. Font Mapping Table

A font mapping table is an object that specifies a font substitution that is used when a requested character set or code page is unavailable and a similar one is available.

- System tables are provided with the operating system. You cannot modify these tables.
- User tables may be used to provide alternate fonts substitution. Five types of font mapping tables are available:
  – Printer-resident to host-resident font character
  – Host-resident to printer-resident font character
  – Printer-resident to host-resident code page
  – Host-resident to printer-resident code page
  – Printer-resident to printer-resident font character set

Refer to 11.1, “Introduction to font mapping table types” on page 237, for more information about the different font mapping tables.

11.2.1 Displaying font substitution tables

You can display information from any system or user font substitution tables available in Operations Navigator. This information can help to understand how substitution is provided. We recommend that you select the “Notify after successful font substitution” parameter in the PSF configuration as shown in Figure 166 on page 250.

Figure 157 shows the list of system tables.

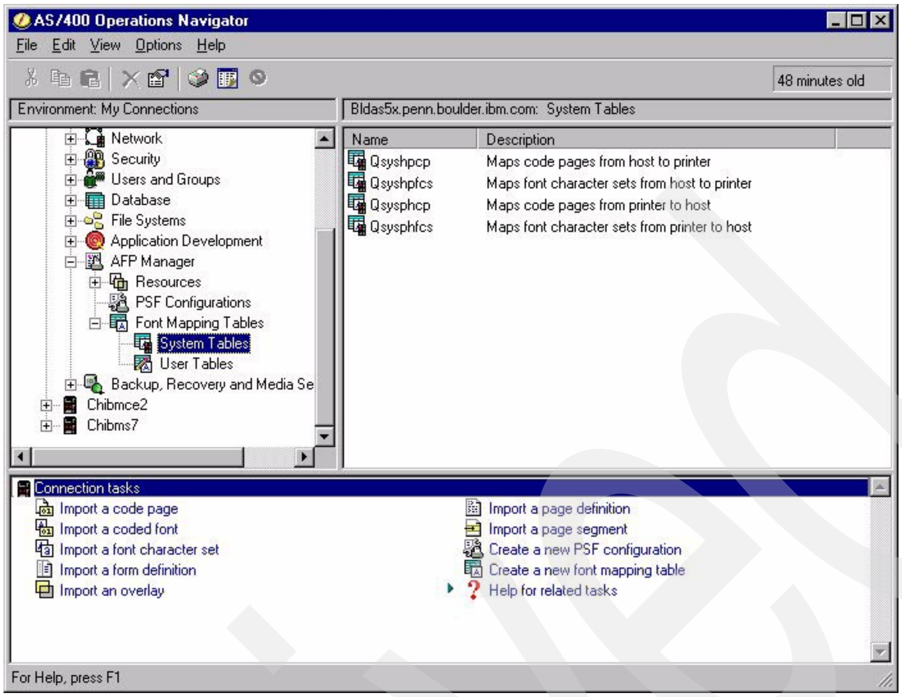
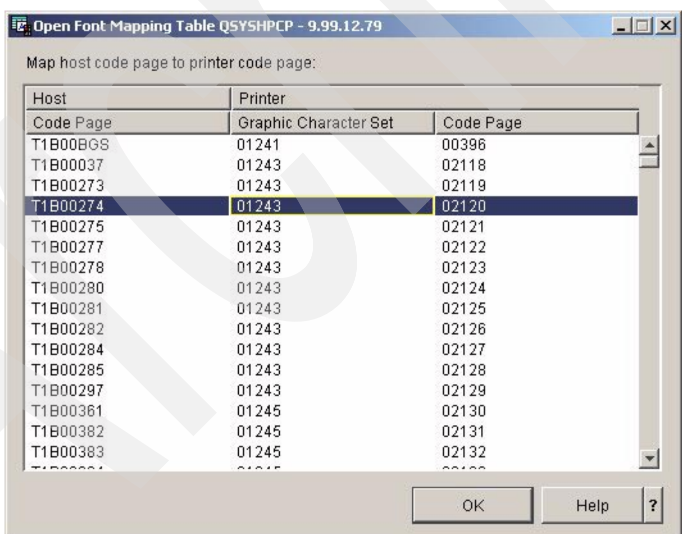


Figure 157. System font substitution tables

You can display the contents of any system or user font mapping table. Simply double-click one of the system tables to display the content. Figure 158 shows the QSYSHPCP table.



Host	Printer	
Code Page	Graphic Character Set	Code Page
T1B00BGS	01241	00396
T1B00037	01243	02118
T1B00273	01243	02119
T1B00274	01243	02120
T1B00275	01243	02121
T1B00277	01243	02122
T1B00278	01243	02123
T1B00280	01243	02124
T1B00281	01243	02125
T1B00282	01243	02126
T1B00284	01243	02127
T1B00285	01243	02128
T1B00297	01243	02129
T1B00361	01245	02130
T1B00382	01245	02131
T1B00383	01245	02132

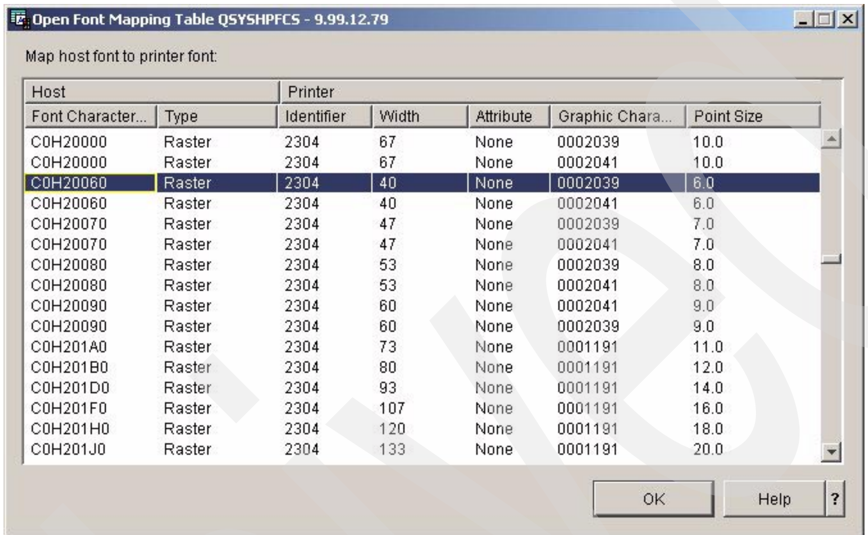
Figure 158. QSYSHPCP substitution table

This panel displays the current values for an HPCP font mapping table:

- **Host**:
  – *Code Page*: Displays the host-resident code page value.

- **Printer**:
  - *Graphic Character Set*: Displays the printer-resident graphic character set.
  - *Code Page*: Displays the printer-resident code page value.

The contents of each table may be different regarding the substitution type of each table. The contents of the host to printer-resident font character set may be displayed as shown in Figure 159.



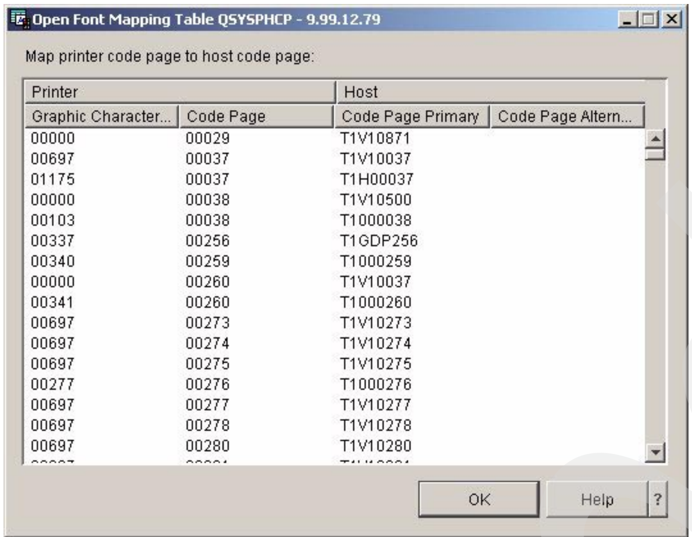
Host		Printer				
Font Character...	Type	Identifier	Width	Attribute	Graphic Chara...	Point Size
C0H20000	Raster	2304	67	None	0002039	10.0
C0H20000	Raster	2304	67	None	0002041	10.0
C0H20060	Raster	2304	40	None	0002039	6.0
C0H20060	Raster	2304	40	None	0002041	6.0
C0H20070	Raster	2304	47	None	0002039	7.0
C0H20070	Raster	2304	47	None	0002041	7.0
C0H20080	Raster	2304	53	None	0002039	8.0
C0H20080	Raster	2304	53	None	0002041	8.0
C0H20090	Raster	2304	60	None	0002041	9.0
C0H20090	Raster	2304	60	None	0002039	9.0
C0H201A0	Raster	2304	73	None	0001191	11.0
C0H201B0	Raster	2304	80	None	0001191	12.0
C0H201D0	Raster	2304	93	None	0001191	14.0
C0H201F0	Raster	2304	107	None	0001191	16.0
C0H201H0	Raster	2304	120	None	0001191	18.0
C0H201J0	Raster	2304	133	None	0001191	20.0

Figure 159. QSYHPFCS substitution table

This panel displays the current values for an HPFCS font mapping table:

- **Host**:
  - *Font Character Set*: Displays the font character set for the host-resident font.
  - *Type*: Displays the host font type.
- **Printer**:
  - *Identifier*: Displays the printer-resident font identifier.
  - *Width*: Displays a width for the printer-resident font.
  - *Attribute*: Displays the attribute associated with the printer-resident font.
  - *Graphic Character Set*: Displays the graphic character set to be associated with the printer-resident font.
  - *Point Size*: Displays the point size of the printer-resident font.

Figure 160 shows the content of a host-resident to printer-resident font character set.


Open Font Mapping Table QSYSPHCP - 9.99.12.79

Map printer code page to host code page:

Printer		Host	
Graphic Character...	Code Page	Code Page Primary	Code Page Altern...
00000	00029	T1V10871	
00697	00037	T1V10037	
01175	00037	T1H00037	
00000	00038	T1V10500	
00103	00038	T1000038	
00337	00256	T1GDP256	
00340	00259	T1000259	
00000	00260	T1V10037	
00341	00260	T1000260	
00697	00273	T1V10273	
00697	00274	T1V10274	
00697	00275	T1V10275	
00277	00276	T1000276	
00697	00277	T1V10277	
00697	00278	T1V10278	
00697	00280	T1V10280	

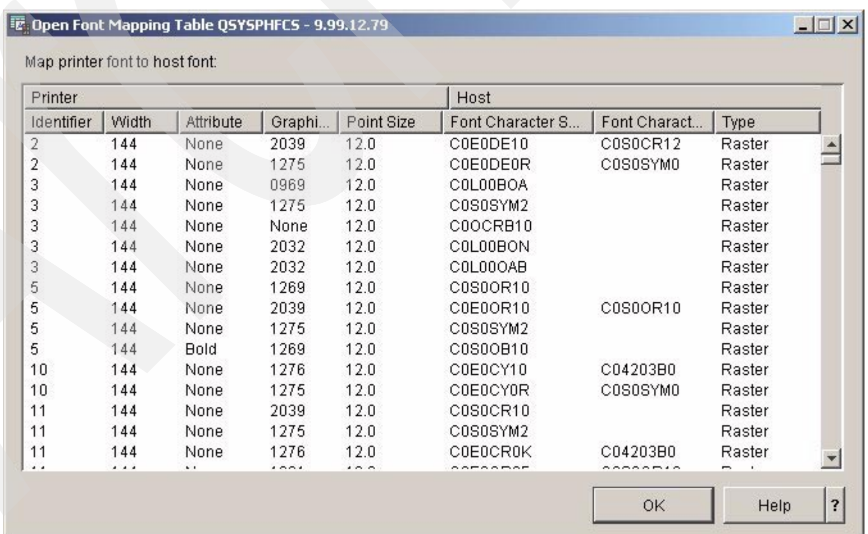
OK Help ?

Figure 160. QSYSPHCP substitution table

This panel displays the current values for the QSYSPHCP font mapping table:

- **Printer**:
  - *Graphic Character Set*: Displays the printer-resident graphic character set.
  - *Code Page*: Displays the printer-resident code page value.
- **Host**:
  - *Code Page Primary*: Displays the host-resident code page value that is the closest match to the printer-resident code page.

Figure 161 shows the content of a printer-resident to host-resident font character set.


Open Font Mapping Table QSYSPHFCS - 9.99.12.79

Map printer font to host font:

Printer					Host		
Identifier	Width	Attribute	Graphi...	Point Size	Font Character S...	Font Charact...	Type
2	144	None	2039	12.0	C0E0DE10	C0S0CR12	Raster
2	144	None	1275	12.0	C0E0DE0R	C0S0SYM0	Raster
3	144	None	0969	12.0	C0L00BOA		Raster
3	144	None	1275	12.0	C0S0SYM2		Raster
3	144	None	None	12.0	C0OCRB10		Raster
3	144	None	2032	12.0	C0L00BON		Raster
3	144	None	2032	12.0	C0L00OAB		Raster
5	144	None	1269	12.0	C0S0OR10		Raster
5	144	None	2039	12.0	C0E0OR10	C0S0OR10	Raster
5	144	None	1275	12.0	C0S0SYM2		Raster
5	144	Bold	1269	12.0	C0S0OB10		Raster
10	144	None	1276	12.0	C0E0CY10	C04203B0	Raster
10	144	None	1275	12.0	C0E0CY0R	C0S0SYM0	Raster
11	144	None	2039	12.0	C0S0CR10		Raster
11	144	None	1275	12.0	C0S0SYM2		Raster
11	144	None	1276	12.0	C0E0CR0K	C04203B0	Raster

OK Help ?

Figure 161. Display QSYPHFCS substitution table

This panel displays the current values for the font mapping table:

- **Printer**:
  - *Identifier*: Displays the printer-resident font identifier.
  - *Width*: Displays a width for the printer-resident font.
  - *Attribute*: Displays the attributes associated with the printer-resident font.
  - *Graphic Character Set*: Displays the graphic character set to be associated with the printer-resident font.
  - *Point Size*: Displays the point size of the printer-resident font.
- **Host**:
  - *Font Character Set*: Displays the font character set for the host-resident font.
  - *Type*: Displays the host font type.

11.2.2 Creating user font substitution tables

It is necessary to create one or more font tables, and then add, alter, or delete entries from them. Only one of each of the four font substitution cases previously described may be created using the Create Font Table (CRTFNTTBL) command. They are assigned a system-supplied name as follows:

- **PHFCS** (printer to host-resident font character set): This creates a table named QPHFCS in the QUSRSYS library, object type *FNTTBL.
- **PHCP** (printer to host-resident code page): This creates a table named QPHCP in the QUSRSYS library, object type *FNTTBL.
- **HPFCS** (host to printer-resident font character set): This creates a table named QHPFCS in the QUSRSYS library, object type *FNTTBL.
- **HPCP** (host to printer-resident code page): This creates a table named QHPCP in the QUSRSYS library, object type *FNTTBL.

One or several of the following user table type may be created and assigned to one or a group of printers. The Font Table parameter of the PSF configuration object is used to assign the table.

- **PPFCS:** Printer-resident to printer-resident font character set (PPFCS) font mapping table

  This creates a table named QPHCP in the QUSRSYS library, object type *FNTTBL.

Figure 162 shows the AFP Manager relationship to the font tables.

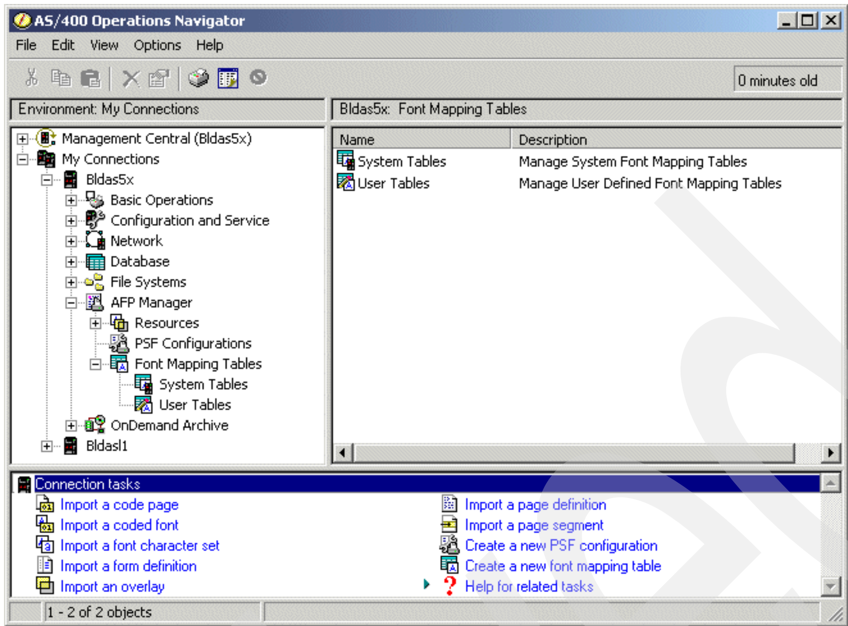


Figure 162. Creating a new font mapping table

To create a new font mapping table, follow these steps:

1. Use the option **Create a new font mapping table** under the Connection tasks window, or select **User Tables**, right-click, and select **New**.
2. The New Font Mapping Table display (Figure 163) appears. Specify a name for the new PPFCS font mapping table. The name cannot be more than 10 characters long and must meet OS/400 name requirements. This option is only available for PPFCS font mapping tables. If you create any other type of font mapping table with this tool, it is named by PSF/400.
3. Specify the library where you want the PPFCS font mapping table to be stored. You can store it in the current library or a different library.

   To specify a different library, enter the name or click **Browse**... to search for a library.
4. Specify a description for your font mapping table. This parameter is recommended but not required. The description cannot contain more than 50 characters.

Figure 163. New Font Mapping Table

5. Specify the object authority you want to give to users that meet all of the following criteria:

   – They do not have specific authority to the object.
   – They are not on an authorization list.
   – Their group profile has no specific authority to the object.

   Specify an authorization list name, or click the drop-down list to select one of the following choices:

   - **Library create**: When the library containing this object was created on the server using the Create Library (CRTLIB) command, a value was specified on the Create authority prompt (CRTAUT parameter). When you select Library create as the object authority, PSF/400 uses that value to determine the authority for this object. If the value specified on the Create authority prompt is changed, the new value does not affect any existing objects.
   - **Change**: The user can perform all operations on the object except:
     – Control its existence
     – Specify its security
     – Move or rename it
     – Change its owner
     – Add members to database files
     – Perform any operation limited to the owner
   - **All**: The user can perform all operations on the object except:
     – Perform any operation limited to the owner
     – Perform any operation controlled by an authorization list management authority
   - **Use**: The user can read the description of the object, place the contents of an entry in the object, locate the object in a library, and use the object as determined by the data authority that the user has to it.
   - **Exclude**: The user cannot access the object.

Table in QUSRSYS

Because customers may already have a table in QUSRSYS, you may not be able to create any of these tables:

- Printer-resident to host-resident font character
- Host-resident to printer-resident font character
- Printer-resident to host-resident code page
- Host-resident to printer-resident code page

In this case, copy this table in your library as a backup and delete the object in QUSRSYS.

11.2.3 Related OS/400 commands

The following native commands are available on OS/400. These commands match the capabilities from AFP Manager:

- Add Font Table Entry (ADDFNTTBLE)
- Change Font Table Entry (CHGFNTTBLE)
- Create Font Table (CRTFNTTBL)
- Delete Font Table (DLTFNTTBL)
- Display Font Table (DSPFNTTBL)
- Remove Font Table Entry (RMVFNTTBLE)

11.3 Example of creating user font substitution tables

The following sections contain examples of using the font substitution tables.

11.3.1 How the printer-resident to printer-resident mapping table works

After you create a printer-resident to printer-resident font substitution table, one of the following processes is performed:

- If the FGID specified in the application is supported by the designated printer, then this font will be selected. The printer-resident to printer-resident font substitution table is *not* searched.
- If the FGID specified in the application is not supported by the printer, then the printer-resident to printer-resident font substitution table *is* searched.
- If a matching entry is found in the printer-resident to printer-resident font substitution font substitution table and the entry is supported by the printer, then the specified substituted FGID in the printer resident font substitution table is used.

However, if a matching entry is *not* found in the printer-resident to printer-resident font substitution table or if the specified substituted font is not supported by the printer, then the system uses its internal font substitution tables to perform the font substitution. (See Table D3 in *OS/400 Printer Device Programming*, SC41-5713.)

11.3.1.1 Creating a printer-resident to printer-resident table

Initially the printer resident font listings should be printed to determine what fonts are available. Font listings printed from IBM printers normally include examples of the different font typestyles combined with their FGIDs and whether they are scalable or fixed pitch.

Figure 164 shows an example from an printer-resident to printer-resident font table.

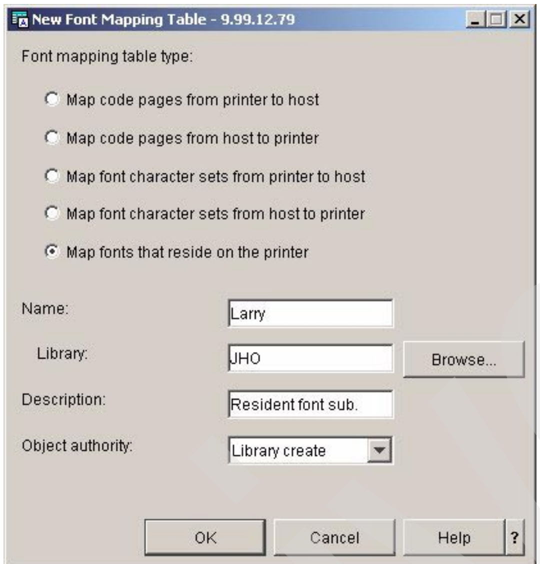


Figure 164. Creating a printer-resident to printer-resident font table

Note

Only printer-resident IPDS or AFP fonts can be mapped. This function does not work with printer resident PCL or PostScript fonts.

The table was created and may be listed in the AFP Manager window. The next step is to add an entry to the table as shown in Figure 165.

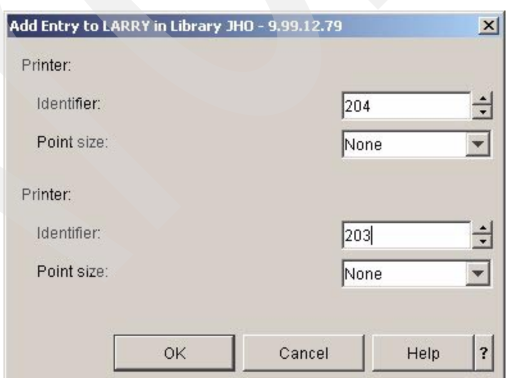


Figure 165. Adding an entry to the table

In this example, we assume that font 204 is not available in the printer and may be replaced by font 203, which is available in the printer. Due to printer compatibility, it makes sense to build a table for a similar printer or printer that has the same set of fonts installed.

Note

Undesirable results may occur if the following types of mapping are made:

- Mapping a monospaced font to a typographic font and mapping a typographic font to a monospaced font.
- Mapping a scalable font to a scalable font with a different point size.

For the printer-resident to printer-resident font substitution table to be used, it must be defined in the Font mapping table parameter in the PSFCFG object as shown in Figure 166.

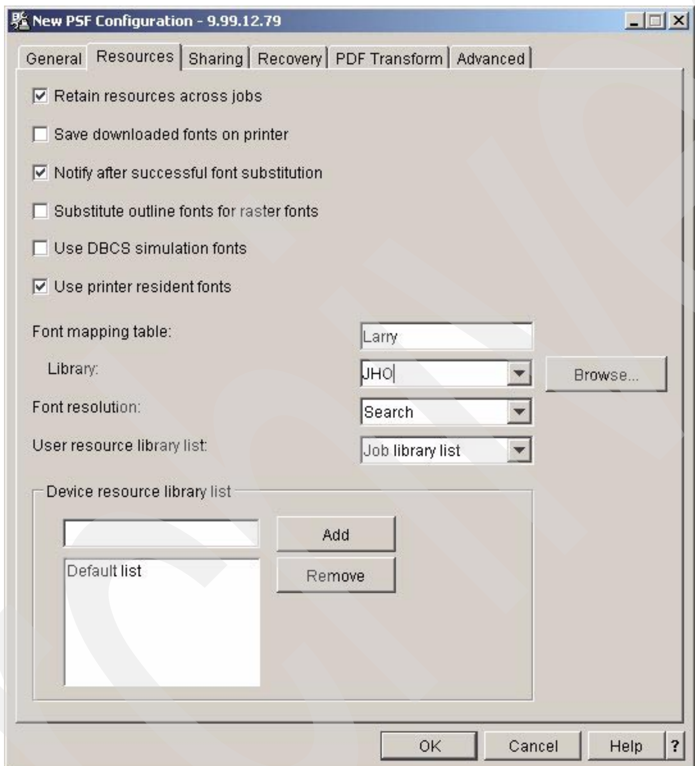


Figure 166. Font mapping table parameter in the PSF configuration object

The advantage of changing the printer-resident to printer-resident font mapping table is two-fold:

- Enhance the look of printed output without needing to change the application. One example concerns a RPG application that was originally designed for its resulting spooled files to be printed on an IBM 3812 in FGID 26 (Matrix Gothic, 10 CPI). The 3812 printer was subsequently replaced with an IBM Infoprint 40 on which FGID 26 is not supported resulting in it being substituted for FGID 11 (Courier, 10 CPI). With the printer-resident to printer-resident substitution table, it was possible to map FGID 26 with FGID other than 11, in this case FGID 304 at Point size 12 (to force 10 CPI).
- Standardize the look of printed output across different printers. Some Lexmark printers have different internal font mapping tables that are equivalent IBM printers. For example, if FGID 87 (Letter Gothic, 12 CPI) is referenced in a spooled file, it will be substituted for FGID 85 (Courier, 12 CPI) on an IBM

Infoprint 40. However, on some Lexmark IPDS printers, this font will be substituted for FGID 86 (Prestige, 12 CPI) even though FGID 85 is also present. Therefore, a printer-resident to printer-resident font table can be created for the Lexmark printer mapping FGID 87 to FGID 86, therefore, enabling the same spooled file to look the same when printed on both printers.

11.3.2 Adding a font table entry

As an example, if you want to use a host-resident font with OfficeVision/400, you must either use a printer that does not support resident fonts (these tend to be larger system printers such as the 3820 and 3835) or switch off printer-resident font support using the CHGPSFCFG command. Your specified font ID is then substituted to a host-resident font according to the font tables documented in Section D.5 of *OS/400 Printer Device Programming*, SC41-5713. This may not be an exact substitution (the table identifies these exceptions), or you may want to use a custom-supplied host font. To do this, you need to add an entry to the QPHFCS font table in QUSRSYS.

Suppose you are using FGID 75 (Courier 12 cpi) in your OfficeVision/400 documents. This is normally substituted to C0S0CR12, which is not an exact match.

Select the **QPHFCS** table as shown in Figure 167.

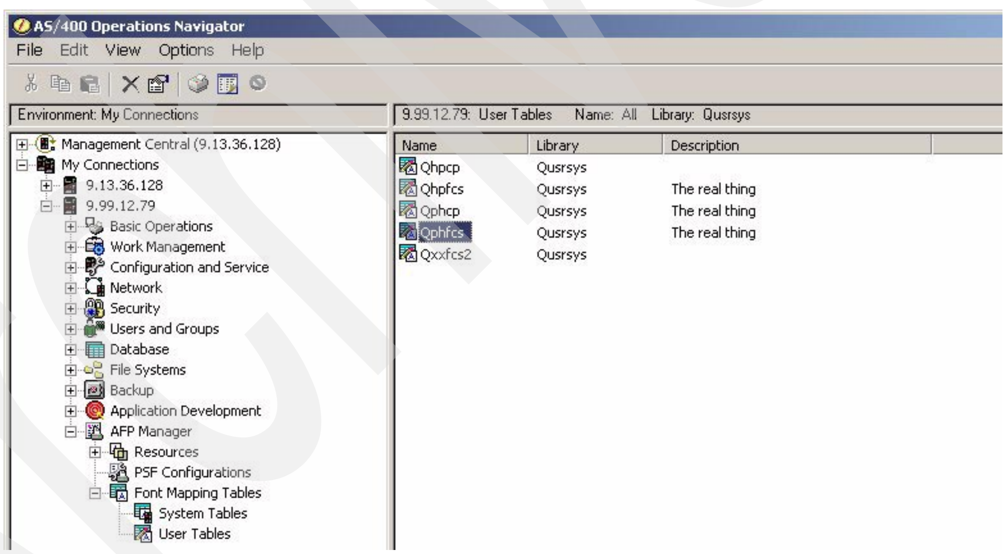


Figure 167. Selecting the QPHFCS font table

If the Core Interchange Fonts are installed on your system, you can substitute C04200B0 instead as shown in Figure 168. The raster font C04200B0 has to be installed on your system and match the printer resolution.

Figure 168. Adding a different printer-resident to host-resident font substitution

> **Note**
>
> The WIDTH keyword in the previous command refers to the characters per inch value (12 in our example) divided into 1440. These values for the common cpi sizes (10, 12, 15, etc.) may be listed on the printer IPDS font listing.

You can then determine whether to end the writers immediately or to defer the font table changes to a later time.

After the entry completes successfully, you can verify the entry in the window as shown in Figure 168.

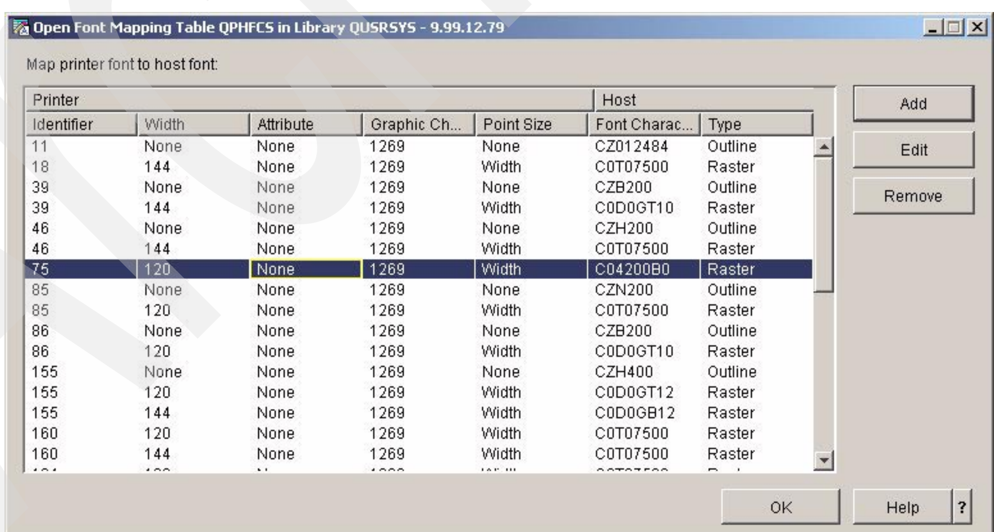


Figure 169. Adding a different printer-resident to host-resident font substitution

Double-click the **QPHFCS** table to view the contents of the table.

A final point to note in the case of OfficeVision/400 is that you are probably still restricted to monospaced (fixed-pitch) host-resident fonts because the alignment of tabs and columns is incorrect if typographic fonts (variable-spaced) are used.

11.4 Additional useful tips

This section provides some additional useful tips about font substitution.

11.4.1 Disabling resident font support

You can disable resident font support on the printer. Otherwise, normal font substitution occurs. To do this, follow these steps:

1. Ensure the printer writer is ended:

   ```
   ENDWTR PRTNP17 *IMMED
   ```

2. Use the CRTPSFCFG or CHGPSFCFG commands. These may be changed without affecting other settings:

   ```
   CHGPSFCFG PSFCFG(PRTNP17) RESFONT(*NO)
   ```

   Or you can use AFP Manager as described in 10.2.1, “Resource configuration” on page 224.

11.4.2 Suppressing font substitution messages

Normally font substitution is logged in the job log, and a message, such as the following example, is sent to the message queue defined in the printer device description (usually QSYSOPR):

```
PQT2072  Font substitution was performed
```

At Version 4.0 Release 2.0, these messages may be suppressed, if desired, using the FNTSUBMSG keyword on the CRTLPSFCFG or CHGPSFCFG command. The default is *YES to continue generating these messages as at present. Otherwise, you can block the messages with this command:

```
CHGPSFCFG PSFCFG(NP17) FNTSUBMSG(*NO)
```

Messages indicating that font substitution failed are not blocked.

11.4.3 Cannot allocate object

You must also ensure that any writers to printers configured as *IPDS, AFP=*YES are ended before you attempt to change the font tables. Otherwise, you receive messages similar to these examples:

```
Cannot allocate object QPHFCS in library QUSRSYS
Font table QPHFCS in library QUSRSYS not changed
```

If this occurs, use the following command to locate which writers are still active:

```
WRKOBJLCK OBJ(QUSRSYS/QPHFCS) OBJTYPE(*FNTTBL)
```

Chapter 12. Using form definitions and page definitions

The use of form definitions and page definitions to format line data for an AFP application originated in the mainframe environment around 1983. Users of Infoprint Manager for AIX and Infoprint Manager for Windows NT and 2000 commonly use these objects. The AS/400 system has supported data streams that use these objects since V2R1 and generated them natively since V3R2/V3R7.

Before V5R1, few customers on the AS/400 or iSeries server used form definitions and page definitions to format their data. There were many other options for output formatting on the iSeries, such as DDS and APU, which could essentially generate comparable results. The original tool for generating form definitions and page definitions on the AS/400 system, Page Printer Formatting Aid (PPFA), required a reasonably high skill level. Prior to V5R1, you could not send spooled files using page definitions to an ASCII printer using Host Print Transform.

These limitations are addressed by a number of enhancements and new products made available with V5R1. This chapter describes these changes and how they affect the use of form definitions and page definitions on the iSeries server. Basic concepts and implementation considerations for the iSeries server are also covered.

This chapter provides a very high level view of form definition and page definition processing. For additional detailed information on any of the topics described in this chapter, refer to *IBM Page Printer Formatting Aid*, S544-5284.

12.1 Form definition and page definition basic concepts

Form definition and page definition objects are used to enhance the formatting of line data in an AFP environment. They provide similar, but not identical, functions to the iSeries concept of using DDS and printer files to format print applications.

12.1.1 Form definition basics

The form definition contains the specifications that deal primarily with the physical attributes of the printed output. It controls page origin and orientation, overlays, copies, paper source and output bin, simplex or duplex, print quality, and multi-up. The form definition also determines whether selected fields from the input data are suppressed.

Within a form definition, you may specify one or more copy groups, and within the copy groups, there may be specifications for one or more subgroups. These are described in 2.8, “Introduction to form and page definitions” on page 51.

12.1.1.1 Copy group

A *copy group* defines the current physical page within a form definition. Note that it defines the physical page and could, therefore, include the formatting instructions for two sides of a piece of paper. A form definition might contain only one copy group. If a different copy group is invoked, it follows that printing must start on a new physical page. An alias for copy group is *medium map*.

Common purposes for a copy group include the ability to choose from a different drawer on the printer. Another example is a switch between simplex and duplex. A final example is the printing of an overlay of standard terms and conditions on the reverse side of client copies of a purchase order. Switching between copy groups in the form definition is controlled by conditional processing in the page definition.

12.1.1.2 Subgroup

A *subgroup* is a subset of a copy group. Up to 127 subgroups can be constructed within any one copy group. The easiest way to remember the function of the subgroup is to think of multi-part stationery (top copy, pink copy, blue copy, etc.), as commonly used on impact printers. To replicate this electronically, we can use multiple subgroups to repeat each printed page as required. The difference is that each copy may be altered subtly if required.

Chapter 2, "Advanced use of IBM Infoprint Designer for iSeries" on page 33, describes a sample application called BACK2. It specifies that each page of the original spooled file is to be printed twice. The first copy uses an Invoice overlay with a constant back containing the Terms and Conditions overlay. The second copy is drawn from a different paper drawer and uses the Packing Slip overlay. Figure 170 illustrates the form definition source you must generate if you are coding that application using PPFA (rather than the Infoprint Designer interface).

```
SETUNITS  10.00 CPI  6.00 LPI LINESP  6.00 LPI ;
FORMDEF BACK2
      OFFSET  0.00 MM   0.00 MM
      REPLACE YES
      COMMENT '';
 SUPPRESSION DFLT;
 SUPPRESSION PACK;
 COPYGROUP COPY1
      DUPLEX NORMAL
      CONSTANT BACK;
  OVERLAY TERMS TERMS NORASTER;
  OVERLAY PACK PACK NORASTER;
  OVERLAY INVOIC INVOIC ;
  SUBGROUP FRONT BIN 1 OVERLAY INVOIC SUPPRESSION DFLT;    /*Invoice*/
  SUBGROUP BACK OVERLAY TERMS;    /*T's & C's*/
  SUBGROUP FRONT BIN 2 OVERLAY PACK SUPPRESSION DFLT PACK; /*Packing*/
  SUBGROUP BACK BIN 2;     /*Blank */
```

Figure 170. Sample PPFA source for a form definition

A form definition is required for all AFP printing. On the iSeries server, if a form definition is not specified explicitly, one is built into the spooled file based on the printer file specifications. Form definitions are used with any printer file type, except one that is used to generate *USERASCII. The most common use of custom or user generated form definitions is with printer files that are designated as *LINE, and, in most cases, they are used along with a page definition.

12.1.2 Page definition basics

A page definition contains the specifications for the logical page layout. You can specify the positioning, rotation, suppression, and font of entire lines of data or specific fields within a line. Data fields may be printed as barcodes. Additional overlays and page segments may be placed on the page as determined by instructions in the page definition.

A page definition must have one page format that contains at least one subpage. The subpage contains format instructions for the line and field definitions.

There are two ways to define the mapping within a page definition:

- Traditional line data processing
- Record format line data processing

12.1.2.1 Traditional line data processing

Until recently, traditional line data processing was the only way to define the formatting rules in a page definition. The input data must be presented in a fairly structured and predictable manner. For example, if a certain piece of information must be printed in a certain way (with a specific font at a specific location), it must be on the same relative line and position on each page in the original spooled file.

The basic element of formatting is the PRINTLINE command. Data records from the input are processed in sequence with the formatting rules in the page definition's PRINTLINE commands. Groups of lines may be processed identically by using the REPEAT subcommand.

The logic may be altered slightly by using channel codes. For example, the character "1" in the first column is usually an indication to start a new page.

A simple page definition is shown in Figure 171. It was designed to simulate PAGRTT(*COR) or (*AUTO), which is commonly used on the AS/400 or iSeries server for system printing. Note the use of the CHANNEL and REPEAT subcommands in the PRINTLINE command.

```
SETUNITS 1 IN 1 IN
         LINESP 8.8 LPI;

PAGEDEF STD132
   REPLACE YES
   WIDTH 10 IN HEIGHT 7.5  IN
   DIRECTION DOWN;

FONT CR13 CS 420090 CP V10500;

PRINTLINE CHANNEL 1 REPEAT 66
   FONT CR13
   POSITION MARGIN TOP;
   ENDSUBPAGE;
```

Figure 171. Simple page definition

Fields within a line may be formatted individually. They are identified within the print record by the START and LENGTH subcommands. If a font is specified for the field, it overrides the font specified in the PRINTLINE. The placement of the field, if specified, is relative to the placement of the PRINTLINE.

Figure 172 illustrates how a PRINTLINE command can have individual fields formatted using the FIELD command. The first PRINTLINE is used to format the first line of data. The second PRINTLINE is used for the next five lines of data.

```
PRINTLINE POSITION 1 IN 1 IN FONT TNR12;
   FIELD START 1 LENGTH 4;
   FIELD START 11 LENGTH 4
      POSITION 4 IN 3 IN
      FONT HELV10;

PRINTLINE POSITION 3 IN 4 IN FONT HELV10 REPEAT 5;
      FIELD START 3 LENGTH 7 FONT TNR12;
      FIELD START 10 LENGTH 4;
```

Figure 172. FIELD formatting within a PRINTLINE

12.1.2.2 Record format line data processing

Traditional line data processing works well enough for users migrating from impact to laser since the data likely prints in predictable places on any pre-printed forms that are in use at the time. However, it has not proven to be flexible enough to meet the needs of some customers who require more control in determining exactly how the data should be printed.

Record format line data processing was developed for page definitions recently to include functions required by customers with more complex print requirements, such as financial statements for clients with multiple account types or telephone bills with multiple types of charges.

Record format line data processing includes features for:

- Automatic page management:
  – Headers, footers, page numbering
  – Grouped item overflow
  – Subheadings
  – Page overflow
- Simple graphics:
  – Lines, boxes, circles, ellipses
  – Filled areas
- Text enhancements:
  – Delimited fields on input
  – Right and left alignment on output

The main difference is that the basic element of formatting has changed from the PRINTLINE command to the LAYOUT command. Each record of data must have a 10 byte record ID that matches the corresponding LAYOUT command that is used to format it. (With V5R1, the DDS record name is mapped automatically to the 10-byte record format name. You can find more on DDS in 12.2.5, “DDS support for record format line data” on page 270.) The sequence of the LAYOUT commands in the page definition is not a consideration.

There are different layout types:

- **Body**: This is used for most data, such as detail lines. A page definition may have multiple body layouts for different detail record types, such as local or long distance phone charges.
- **Page header**: This is automatically printed on each page. You may use a default header or define your own.

- **Page trailer**: Similar to a page header, this prints at the bottom of each page.
- **Group header**: Group headers precede a group of body records. They may continue on the next page if necessary. They are in effect until a new body group is started or NOGROUP is specified.

Another major enhancement that is available with record format line data processing is the ability to use variable length fields in the data, separated by a character defined with the DELIMITER subcommand. The individual fields are selected by the FLDNUM subcommand rather than by the start position and length. In addition, selected characters within delimited fields may be selected by specifying the START and LENGTH with respect to the delimiter. This function may prove useful when dealing with languages that are more adept at string manipulation, such as Java.

Figure 173 shows examples that use LAYOUT for record formatting.

```
Sample LAYOUT and FIELD Commands for Body, no delimiter characters:

LAYOUT C'layout1' BODY GROUP
   DIRECTION ACROSS
   COLOR CYAN
   OVERLAY MYOVL 0.5 0.25 OVROTATE 90;
   FIELD START 1 LENGTH 20
         ALIGN LEFT
         POSITION 0 CURRENT
         FONT FONT1;

Sample LAYOUT and FIELD Commands for BODY using field delimiter:

LAYOUT C'layout2' BODY DELIMITER C'%';
   FIELD FLDNUM 1
      POSITION 0 NEXT
      FONT FONT2;


Sample LAYOUT Command for GROUP HEADER:
LAYOUT C'headr1' GRPHEADER XSPACE .2
   IN POSITION SAME .6 IN;

Sample Layout Command for PAGE HEADER:
LAYOUT C'pghd' PAGEHEADER NEWPAGE
   POSITION .6 IN ABSOLUTE .55 IN;
```

Figure 173. Sample commands for record formatting

Many other functions are supported when record formatting line data processing is used that are not supported with traditional line data processing:

- Field delimiter and subfield
- Automatic page numbering
- Automatic page overflow management
- Simple graphics: Lines, boxes, circles, ellipses
- Boxes or lines that enclose a variable number of input lines
- Right or left alignment of text fields on output
- Conditional processing testing of substrings of delimited fields

Figure 174 shows some examples of commands for the new graphical features available with record format line data processing.

```
Draw four identical lines, 0.25 in apart.
DRAWGRAPHIC LINE POSITION LPOS CPOS +1.0 DOWN 3.0 IN
         LINETYPE DASHDOT LINEWT MEDIUM
         COLOR ORANGE COPY ACROSS 4 SPACED 0.25;

Circle, filled with dotted pattern.
DRAWGRAPHIC CIRCLE POSITION LPOS NEXT
         RADIUS 1.0 IN LINEWT MEDIUM LINETYPE SOLID
         FILL DOT04;

Ellipse
DRAWGRAPHIC ELLIPSE POSITION LPOS NEXT
         AXIS1 -2 -2 AXIS2 +2 +2 LINEWT MEDIUM
         LINETYPE SOLID;

Using ENDGRAPHIC to make variable size box.
DRAWGRAPHIC BOX GRAPHID 01
         BOXSIZE 3 IN POSITION 1.75 IN .3 IN;

ENDGRAPHIC GRAPHID 01 LPOS;
```

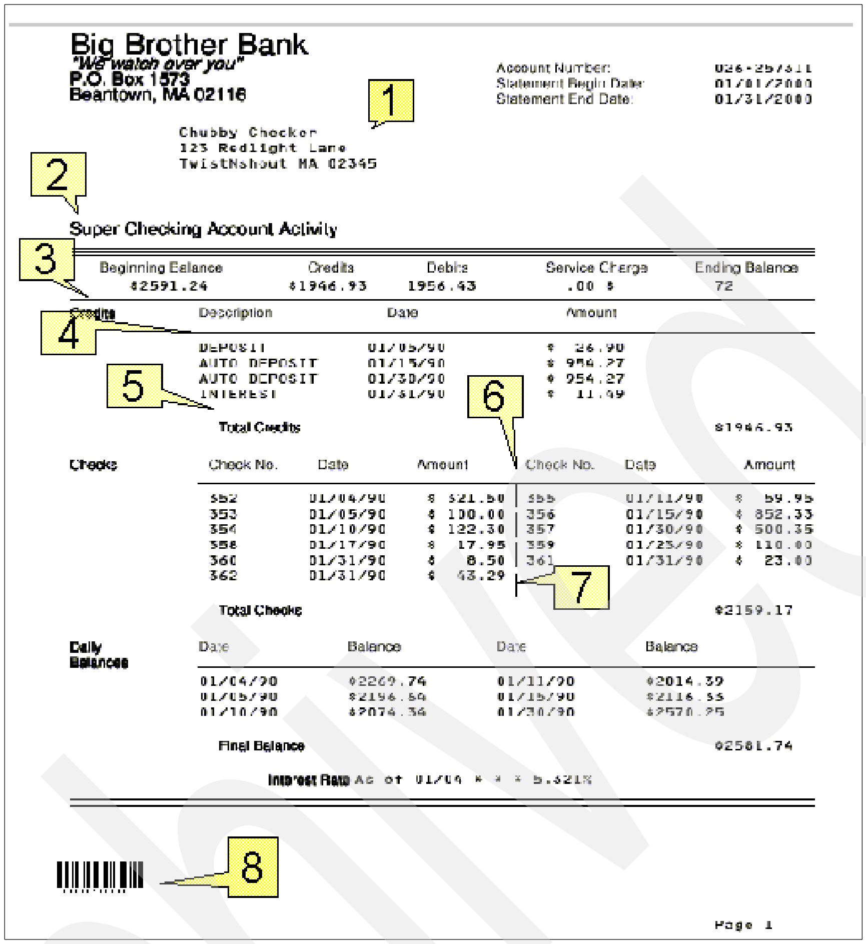
Figure 174. Examples of commands for new graphic features

Note

Neither traditional line data processing or record format line data processing provides a means to edit the data similar to the way DDS and some other iSeries high level languages such as RPG use Edit Words or Edit Codes. Currency, punctuation, or date diameter characters must be included in the original data being passed to the page definition.

An example of output created using record format line data processing is shown in Figure 175. It illustrates the following features of record formatting (note the corresponding numbers):

1. Page header *statmid* appears on each page.
2. Body layout *statsum* produces the summary information. Horizontal lines are created by using the DRAWGRAPHIC command.
3. Group header *crheader* produces a “Credits” subheading. It is repeated if the group spans a page.
4. Body group *crdata* contains the credit detail lines. The amount column is right-aligned.
5. Body group *crtotal* prints the total credit information. The data is passed from the application. It is not calculated automatically.
6. A variable length vertical line in the Checks group is defined by the DRAWGRAPHIC LINE DOWN command.
7. ENDGRAPHIC commands end the vertical line.
8. Page trailer *pgenum* ends the page, prints the page number, and prints a barcode. It is printed on each page.



Big Brother Bank
"We watch over you"
P.O. Box 1573
Beantown, MA 02116

Account Number: 026-257311
Statement Begin Date: 01/01/2000
Statement End Date: 01/31/2000

Chubby Checker
123 Redlight Lane
TwistNshout MA 02345

Super Checking Account Activity

Beginning Balance	Credits	Debits	Service Charge	Ending Balance
$2591.24	$1946.93	1956.43	.00 $	72

Credits

Description	Date	Amount
DEPOSIT	01/05/90	$ 26.90
AUTO DEPOSIT	01/15/90	$ 954.27
AUTO DEPOSIT	01/30/90	$ 954.27
INTEREST	01/31/90	$ 11.49

Total Credits $1946.93

Checks

Check No.	Date	Amount	Check No.	Date	Amount
352	01/04/90	$ 321.50	355	01/11/90	$ 59.95
353	01/05/90	$ 100.00	356	01/15/90	$ 852.33
354	01/10/90	$ 122.30	357	01/30/90	$ 500.35
358	01/17/90	$ 17.95	359	01/25/90	$ 110.00
360	01/31/90	$ 8.50	361	01/31/90	$ 23.00
362	01/31/90	$ 43.29			

Total Checks $2159.17

Daily Balances

Date	Balance	Date	Balance
01/04/90	$2269.74	01/11/90	$2014.39
01/05/90	$2196.64	01/15/90	$2116.33
01/10/90	$2074.34	01/30/90	$2570.25

Final Balance $2581.74

Interest Rate As of 01/04 * * * 5.321%

Page 1

Figure 175. Output example produced using record formatting

For a complete listing of a page definition and corresponding data for a similar example, refer to *IBM Page Printer Formatting Aid*, S544-5284.

Note

You *cannot* mix LAYOUT commands and PRINTLINE commands in the same page definition.

12.1.3 Conditional processing

You can change the formatting of different pages of your output based on rules that you define in the page definition. This is called *conditional processing*. Based on data that the page definition finds in a predictable location, you can choose a different set of rules for the physical aspect of the job, by selecting different copy groups from the form definition. For example, if the customer number begins with the letter "E", use the copygroup that prints an English overlay; if it begins with an "F", use the copygroup that prints a French overlay.

Similarly, you may use conditional processing to select a different layout for the data, or page format, from the page definition. An example of this function is printing the data from a total or summary page differently from the way the detail pages print. Conditional processing may also be used to force printing on a new side or a new sheet of paper in multi-up or duplex applications cases.

Section 2.9.3.2, “Example of simple conditional processing” on page 62, describes a scenario where a different page format is used depending on the information that prints in the total box on the form. If it has the word “Continued”, you should use the page format called “CONTD”. Otherwise, use the TOTAL page format. With Infoprint Designer, the setup of the condition is done interactively.

The equivalent page definition commands you would enter using PPFA are shown in Figure 176.

```
CONDITION TOTALBOX START 70 LENGTH 9
     WHEN EQ
      'Continued' BEFORE SUBPAGE
      CURRENT
      PAGEFORMAT CONTD
     OTHERWISE BEFORE SUBPAGE
      CURRENT
      PAGEFORMAT TOTAL;
```

Figure 176. Sample conditional processing command

If the copy group from the form definition is affected by the results of the condition test, the word “CURRENT” is replaced by a COPYGROUP subcommand. Otherwise, the word CURRENT simply instructs you to keep using the same copy group that was in effect.

There is a slight difference in the syntax for the CONDITION command when it is used with traditional line formatting versus record formatting. However, the overall concept and functions are similar.

12.1.4 Line data basics

The origin of line data goes back to impact printers that printed one line of information at a time. A physical tape with with holes in specific positions or channels was used to control spacing and skipping.

On the iSeries server, the term *line data* specifically refers to spooled files that have DEVTYPE(*LINE) specified. The data may look similar to the default data that is generated when one uses DEVTYPE(*SCS), but SCS supports some embedded formatting instructions, such as font, cpi, and variable line spacing, where LINE does not.

Usually, LINE spooled files are generated with an extra byte at the beginning for the carriage control or forms control byte. In page definition terms, this is referred to as the *channel code*.

The extra byte for the channel code is mandatory if you are embedding selected AFP instructions within the LINE data. The AFP instructions must begin with a Hex’5A’. An example of such an instruction is an Invoke Media Map to change the copy group used within a form definition. Channel codes are ignored if you are using record format line data processing, but may be necessary if you include AFP instructions in the data.

Fonts are usually defined by using the FONT subcommand in a PRINTLINE command. An alternate way is to add one additional character (after the forms control byte) to use as a font selection code. Or, in page definition terms, the character can be used as a Table Reference Character (TRC). TRCs are only supported using traditional line data processing.

In the case of record format line data processing, the first 10 characters of the record (after the carriage control byte if one exists) must contain the record ID. The data following the record ID may be the traditional format where the fields are determined based on their position in the line. An alternative is to separate the fields by a delimiter character.

Figure 177 shows examples of different ways to generate line data for traditional and record format processing.

```
Traditional Line Data with CC
1This is the top of the page
 Field1     Field2         Field3

Record Format data with optional diameter characters
pagehead  This is the top of the page
detail    Field1%Field2%Field3
```

Figure 177. Sample line data

If the data is referenced by FIELD, the START position in the page definition does not include the channel codes or TRCs, in the case of traditional line data processing, or the 10 character record ID used with the record format line data processing.

Version 5 Release 1 provides two new methods to easily produce record format line data. Section 12.2.4, "Java support for record format line data" on page 269, and Chapter 13, "Printing from Java applications" on page 275, describe how to produce record format line data spooled file from a Java application program. Section 12.2.5, "DDS support for record format line data" on page 270, describes the new DDS support for record format line data.

12.1.4.1 Embedding AFP instructions

One method of enhancing the output produced by form definition and page definition processing is to insert Advanced Function Presentation instructions in the form of structured field records right into the data stream. This may be done with either type of formatting, traditional line data, or record format line data. It allows the application programmer to have more flexibility than is otherwise offered by the relatively constrained rules of conditional processing.

A spooled file that contains both data records and AFP structured fields is called *mixed*. On iSeries, the printer file is generated as DEVTYPE(*AFPDSLINE).

The AFP structured fields that may be embedded in the data stream are:

- Invoke Media Map: Used to change copy group
- Invoke Data Map: Used to change page format
- Include Page Segment
- Include Page Overlay
- Presentation Text

If you are building the data stream manually, the data records must be preceded by a carriage control byte and the AFP records must begin with X’5A’. For more information on mixed data streams, refer to *Advanced Function Presentation Programming Guide and Line Data Reference*, S544-3884.

If you are using DDS to create your spooled data, the Invoke Media Map (INVMMAP) and Invoke Data Map (INVDTAMAP) keywords can be used to insert AFP structured fields in an AFPDSLINE data stream.

12.1.5 iSeries printer file keywords

When you create or change a printer file for use with form definitions and page definitions, you must consider the following parameters in the CRTPRTF or CHGPRTF commands.

DEVTYPE

The most common way to take advantage of form definition and page definition processing is to use a printer file defined with DEVTYPE(*LINE), FORMDF(mylib/myformdf), and PAGDFN(mylib/mypagdfn). Page definitions may also be used with printer files defined as DEVTYPE(*AFPDSLINE) if embedded AFP structured fields are to be used.

You may specify a form definition for use with spooled files that have a DEVTYPE value of *LINE, *AFPDSLINE, *AFPDS, *SCS, or *IPDS.

FORMDF and PAGDFN

It is most common to include both parameters in the printer file. They may be library qualified. If *LIBL is used, they must be placed in libraries that PSF/400 can find for the writer, either by the job library list, a default library list, or libraries specified in the PSF configuration object.

If either the FORMDF or PAGDFN parameters are set to *NONE, and you specify DEVTYPE(*LINE) for printing to an AFP printer, PSF/400 builds the required object inline based on the appropriate printer file parameters.

CTLCHR

This parameter tells PSF if you are using control characters in your data, and if so, the type you are using.

If your application uses DDS keywords, such as SKIPA or SPACEB, or other similar high level language instructions to control skipping and spacing, specify CTLCHAR(*NONE). If the application data itself includes the control characters, you must specify CTLCHAR(*FCFC) for ANSI control characters or CTLCHAR(*MACHINE).

TBLREFCHR

If the spooled data includes a table reference character to control font selection, specify TBLREFCHR(*YES).

AFPCHARS

Use this keyword to identify up to four coded fonts to use if you select TBLREFCHR(*YES). The names to use here are the four character-coded font names. PSF/400 adds ‘X0’ to the beginning of the name to find the font resource object.

DRAWER and DUPLEX
In most cases, the form definition and page definition parameters override equivalent functions specified in the printer file. Two exceptions to this rule are the DRAWER and DUPLEX parameters. The value specified for these parameters takes precedence over the form definition specification. To force PSF/400 to use the values in the form definition, you must specify DRAWER(*FORMDF) and DUPLEX(*FORMDF).

CVTLINDTA
This parameter specifies whether to convert line data and a page definition to AFPDS before the data is spooled. See 12.2.6, “CVTLINDTA parameter in printer files” on page 273, for further information on using this parameter.

Other print file keywords
The following printer file keywords are *ignored* when line data is specified and a page definition is used:

- CDEFNT
- CHRID
- CPI
- FOLD
- FONT
- FNTCHRSET
- LPI
- LVLCHK
- MULTIUP
- PAGESIZE
- PAGRTT
- REDUCE

The following printer file keywords are *ignored* when line data is specified and a form definition is used:

- BACKMGN
- DRAWER (if *FORMDF is specified)
- DUPLEX (if *FORMDF is specified)
- FOLD
- FORMFEED
- FRONTMGN
- LVLCHK
- MULTIUP
- PAGRTT
- PRTQLTY
- REDUCE
- SADLSTITCH

12.1.6 Converting existing SCS spooled files to LINE

You may have an application that already produces SCS output, but you want to migrate to form definitions and page definitions. In this case, you must change the printer file in the application or override it in the control language before it is used.

If this is not possible, you may be forced to work with the SCS output after it is already generated as a spooled file. Perform the following steps to convert an SCS spooled file to a LINE spooled file:

1. Create a physical file that has a record length that is one character longer than the length of the spooled file records. This accommodates the forms control character that is used to preserve the line spacing (otherwise, all blank lines are ignored or dropped):

   ```
   CRTPF CPYSPLF RCDLEN(133) MAXMBRS(*NOMAX)
   ```

2. Copy the spooled file data to the physical file using the CPYSPLF command:

   ```
   CPYSPLF FILE(QSYSPRT)
           TOFILE(CPYSPLF)
           JOB(073040/MSHNIER/QPADEV000C)
           SPLNBR(2)
           TOMBR(QSYSPRT)
   ```

3. Create a new printer file for LINE printing. Name the page definition and form definition and set any other parameters as necessary. You must specify CTLCHAR(*FCFC) for the codes in column one to be interpreted as carriage control codes, and not as data:

   ```
   CRTPRTF FILE(mylineprtf)
           DEVTYPE(*LINE)
           CTLCHAR(*FCFC)
           DRAWER(*FORMDF)
           PAGDFN(pagedef)
           FORMDF(formdef)
           DUPLEX(*FORMDF)
           CVTLINDTA(*YES)
   ```

4. Copy the member from the physical file to the printer file. This generates a spooled file, either as DEVTYPE(*LINE) if you specify CVTLINDTA(*NO). Or if you specify CVTLINDTA(*YES), the page definition instructions are processed and the spooled file is generated as DEVTYPE(*AFPDS).

   ```
   CPYF FROMFILE(CPYSPLF) TOFILE(mylineprtf) FROMMBR(QSYSPRT)
   ```

   This generates a spooled file, either as DEVTYPE(*LINE) if you specify CVTLINDTA(*NO). Or if you specify CVTLINDTA(*YES), the page definition instructions are processed and the spooled file is generated as DEVTYPE(*AFPDS).

This process may be automated by a monitor program. For information, see Appendix D, “Output queue monitor” on page 337.

12.1.7 Generating or obtaining form definitions and page definitions

You may obtain form definitions and page definitions from a variety of sources on the iSeries server or from other platforms.

12.1.7.1 Shipped with PSF/400

A standard set of form definitions and page definitions is shipped with the PSF/400 feature. These are compatible with resources that are shipped on other IBM platforms with PSF or Infoprint Manager products. They are listed in the *AS/400 Guide to Advanced Function Presentation and Print Services Facility*, S544-5319.

12.1.7.2 Infoprint Designer for iSeries

Infoprint Designer for iSeries generates the form definition, page definition, and other resource objects used in the project and automatically places them directly in the iSeries library. No additional processing is necessary other than referencing them in the printer file. For more information, see:

- Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33
- Section 2.8, “Introduction to form and page definitions” on page 51
- Section 12.2.1, “Form and page definitions in Infoprint Designer for iSeries” on page 268

12.1.7.3 AFP PrintSuite: Page Printer Formatting Aid/400

Page Printer Formatting Aid/400 (PPFA) is a tool for the iSeries server that takes the source form definition and page definition specifications and converts them to the corresponding AFP data stream.

To use PPFA, use an editor, such as SEU, to enter the source into a source physical file member. Then, use the PPFA CVTPPFTSRC command to generate the physical file members that contain the AFP versions of the form definition and page definition. Finally, use the CRTFORMDF and CRTPAGDFN commands to build the iSeries objects.

You must apply PTF SF65783 to the PPFA product to create page definitions that use the new record format line data features.

12.1.7.4 Resources from AIX or Windows platforms

Infoprint Manager for AIX and Infoprint Manager for Windows NT and Windows 2000 offer PPFA as optional features. There are also a number of other vendor offerings that are available for those platforms that use a WYSIWYG interface (similar to Infoprint Designer for iSeries) to create form definitions and page definitions.

The easiest way to import resources from a Windows platform is to use AFP Manager that is shipped with Client Access Express with V5R1. See Chapter 9, “AFP Manager” on page 213, for more details.

If the files were created on an AIX system, it may be best to transfer them in binary format to a Windows PC (for example, by using FTP) and then use AFP Manager.

Otherwise, you must follow these steps to build the resources on the iSeries server:

1. Create a physical file on the iSeries server. It may have any record length you want, but you need to specify `LVLCHK(*NO)`.
2. Use the TCP/IP FTP function to transfer the resources into members of the physical file you just created. The files must be transferred in BINARY mode. Otherwise, the iSeries server attempts to perform an ASCII to EBCIC conversion on the data.
3. Once the resource data is loaded in the physical file member, use the CRTFORMDF and CRTPAGDFN commands to create the form definition and page definition iSeries objects.

12.1.7.5 Resources from OS/390

Form definitions and page definitions can be created on the OS/390 platform using PPFA or another vendor’s products. Resources from OS/390 are in files that consist of variable length records. The iSeries server cannot handle this format directly. You must first inspect the OS/390 file to determine the length of the single longest record in the resource and then create the iSeries physical file with that length. You also need to specify `LVLCHK(*NO)`.

Once the data is loaded in the physical file, use the CRTFORMDF and CRTPAGDFN command to create the iSeries resources.

12.2 Summary of V5R1 enhancements relating to form and page definitions

With some of the changes announced in V5R1, we expect more customers to begin looking at form definitions and page definitions as candidates for formatting their data. The changes that affect this decision include:

- The new formatting tool, Infoprint Designer for iSeries, generates form definitions and page definitions.
- Infoprint Server for iSeries can generate index records and resource groups for spooled files in *LINE format that have a page definition specified. The index records and resources can then be used:
  – By Infoprint Server to split large spooled file into multiple PDF files
  – When viewing the file with the AFP Viewer to quickly locate specific pages in the document
  – By archive products, such as On Demand or Common Server
- Enhancements to the page definition architecture to support record format line data.
- A new access class for Java to support the generation record format line data on the iSeries server.
- Changes to DDS to support generating record format line data.
- The CVTLINDTA parameter can be specified in a printer files that has DEVTYPE(*LINE) and uses a page definition. If you specify CVTLINDTA(*YES), the spooled file that is created when the application is run will be generated as fully resolved AFPDS, instead of line data. This means that applications that are developed using Infoprint Designer for iSeries or PPRA can now be:
  – Printed on an ASCII printer using Host Print Transform.
  – Viewed with IBM AFP Viewer from the Operations Navigator interface.

12.2.1 Form and page definitions in Infoprint Designer for iSeries

Infoprint Designer for iSeries is available for V4R5 and V5R1. It provides a WYSIWYG interface for creating overlays and generating form definitions and page definitions. It is described in great depth in Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33.

There are a some features, commands, and subcommands that are part of the architecture for form definitions and page definitions that are not supported today in Infoprint Designer for iSeries. These include:

- Record format line data
- Use of Table Reference Characters (TRCs) to select fonts
- PRTDATA subcommand in the PRINTLINE command

- Form definition subcommands dealing with finishing (stapling) and post processing equipment

Nevertheless, Infoprint Designer is likely to be the choice of many iSeries users since the WYSIWYG design lends itself to very fast application development.

12.2.2 Using Infoprint Server for iSeries

Infoprint Server for iSeries consists of a number of powerful tools for manipulating spooled file data and transforming from one data stream to another. One of the tools is the Create AFP Data (CRTAFPDTA) command. This command requires a spooled file in LINE format and an input page definition.

The output of CRTAFPDTA consists of up to four files in the Integrated File System (IFS) directory. The first is an AFPDS file that is a result of processing the LINE data against the page definition. The other optional files are a file containing index records with one containing copies of selected external resources and one that merges the data from the other three.

The index function is useful when used with another component of Infoprint Server (the IPDS to PDF transform). Index records can segment a large spooled file to individual components, such as single invoices or statements. This process is described in Chapter 14, "End-to-end example" on page 283.

The index records and resource groups may also prove useful when working with archive products, such as On Demand or Common Server. By including the resource group with the spooled file, you are ensured that the correct external resources, such as overlays or page segments, are being used with an old spooled file when it is retrieved.

Finally, the AFP Viewer can take advantage of the index records and resource groups when you view the spooled file.

12.2.3 Record format line data processing

The underlying support for record format line data processing was added to PSF/400 with V5R1. It is described in 12.1.2.2, "Record format line data processing" on page 258.

Resources created on other platforms that use this function are supported on the iSeries with V5R1. Record format line data page definitions may be generated natively on the iSeries server with the PPFA/400 component of the AFP PrintSuite (5798-AF3) by applying PTF SF65783.

12.2.4 Java support for record format line data

In the past, the support for print from Java applications on the AS/400 system was limited and cumbersome.

With V5R1, a new class object was introduced to the IBM Toolbox for Java that supports writing of data from Java to a printer file in LINE format. The LineDataRecordWriter class allows a Java programmer to concentrate on generating the data, and the formatting can be done using form definitions and page definitions as described in this chapter.

LineDataRecordWriter writes a record in a line data format, including the record format name in positions 1 through 10. The data may be presented using a fixed layout or a variable layout may be used that uses a delimiter to separate fields.

For more details, see Chapter 13, “Printing from Java applications” on page 275.

12.2.5 DDS support for record format line data

Data Description Specifications (DDS) provides a means to define the layout and presentment of data external from the program using it. Since 1991, keywords to support AFP functions have been supported for printer files. These includes lines, boxes, host fonts resources, graphics, page segments and overlays. These elements can be placed at absolute positions on the page, or the positional values can be passed as parameters from the program.

This capability provides a great deal of flexibility to a programmer. Theoretically, the checking account document as illustrated in Figure 175 on page 261 could be produced using a high level language such as RPG, COBOL, or C. One of the challenges is that the programmer must keep track of the exact positions of any floating elements that are placed on the page, such as any graphics used to separate the different record types. These positions are passed to DDS as variables.

With the new record format line data support for DDS applications, a programmer can continue to keep the field definitions external to the program, but the logic for positioning the data and other AFP elements can be handled by the page definition. Another advantage of using DDS with PPFA record formatting is the fact that DDS supports the EDTCOD and EDTWRD, where PPFA does not.

12.2.5.1 How it works

When an applications writes to a printer file that is defined with DEVTYPE(*LINE), OS/400 checks to see if the page definition that is named in the PAGDFN parameter uses record format line data commands. If it does, the format of the data written to the spooled file is modified to suit the requirements of this type of page definition:

- The record format name used in the DDS specifications is written in the first 10 characters of the output record.
- The individual fields are placed one after another in the printer file record. All data fields follow, starting in position 11. All specifications relating to the position are ignored. This includes the Line and Position fields in columns 39 to 44, the SKIPB, SKIPA, SPACEB, and SPACEA keywords, and the POSITION keyword.

For example, consider an application that normally writes an address block to one record in the DDS, with the company name, street, and city appearing on three different lines. With record format line data processing, all three fields are placed in one record in the output spooled file.

An example of some DDS specifications for a check record is shown in Figure 178. The DDS compiler requires that a value be entered in positions 42 to 44 of the specifications. This value would normally represent an absolute or relative position on the line. It is ignored by record format line data processing.

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7
     A          R CKDATA                        SPACEB(1)
     A                                        1' '
     A            CHECK         12            2
     A            DATE           6  0        40EDTCDE(Y)
     A                                       +0'  '
     A            AMOUNT         6  2        10EDTWRD('$   0.  ') SPACEB(1)
     A                                       +0'   '
     A            PAYTO         25           +5
```

Figure 178. Example of DDS to be used record format line data page definition

This sample contains some blank spaces inserted as constants. This is done to match the data up with the required positions as defined in a page definition that already existed. Any spaces that are required to match up the data with the page definition layout must be hardcoded because the normal way of positioning will be ignored.

Figure 179 shows the page definition LAYOUT command that is used against the CKDATA record. This is a subset of a page definition sample called *rept1* that is found in the *IBM Page Printer Formatting Aid: User's Guide*, S544-5284.

```
LAYOUT C'CKDATA' BODY GROUP;
  FIELD START 2 LENGTH 3 ALIGN LEFT
      POSITION 1.2 in CURRENT
       FONT varb ; /* Variable text - Check number */
  FIELD START 14 LENGTH 8 ALIGN LEFT
      POSITION .1 in CURRENT
       FONT varb ; /* Variable text - Date */
  FIELD START 35 LENGTH 25 ALIGN LEFT
      POSITION 2.0 in CURRENT
       FONT varb ; /* Variable text - Payable to: */
  FIELD START 24 LENGTH 8 ALIGN RIGHT
      POSITION 5.6 in CURRENT
       FONT varb ; /* Variable text - Amount */
```

Figure 179. Page definition layout to use against sample DDS record

The original sample that was created for OS/390 had lowercase record IDs. The sample is changed to use an uppercase name for the layout because DDS does not allow lowercase record format names.

Figure 180 shows spooled file records generated using the DDS from Figure 178. The DDS keywords EDTCDE and EDTWRD were honored. The appropriate punctuation was added to the currency field and the date. The SPACEB(1) keywords and positioning information from column 42 to 44 were ignored.

```
                                        Display Spooled File
File  . . . . . :   REPTPRT2
Control . . . . .
Find  . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6...
CKDATA     314         10/29/01  $ 904.02   ART BY ERIN
CKDATA     513          8/16/01  $5120.32   AMY'S FINE FOODS
```

Figure 180. Spooled file after processing with record format line data

Remember that the when using START and LENGTH to define fields for record format line data processing, the start position begins after the 10-character record ID. In this case, the check number is in position 12 of the printed record, but the page definition references it as starting in position 2.

The following keywords are honored by record format line data processing:

- ALIAS
- DATE
- DATFMT
- DATSEP
- DFT
- DLTEDT
- EDTCDE
- EDTWRD
- FLTFIXDEC
- FLTPCN
- IGCALTTYP
- IGCANKCNV
- INDARA
- INDTXT
- MSGCON
- PAGNBR (may not be of much use)
- REF
- REFFLD
- TEXT
- TIME
- TIMFMT
- TIMSEP

Keywords relating to formatting are ignored:

- BARCODE
- BLKFOLD
- CDEFNT
- CHRID
- CHRSIZ
- COLOR
- CPI
- CVTDTA
- DFNCHR
- DOCIDXTAG
- DRAWER
- DTASTMCMD
- DUPLEX
- ENDPAG
- ENDPAGGRP
- FNTCHRSET
- FONT
- FORCE
- GDF
- HIGHLIGHT
- INVDTAMAP
- INVMMAP
- LINE

- LPI
- OUTBIN
- OVERLAY
- PAGRTT
- PAGSEG
- POSITION
- PRTQLTY
- SKIPA
- SKIPB
- SPACEA
- SPACEB
- STRPAGGRP
- TRNSPY
- TXTRTT
- UNDERLINE
- ZFOLD

Option indicators in position 7 through 16 of the DDS specification will be honored. This could pose a problem because subsequent fields on the same line will have their position altered based on whether the indicator is on or off. We recommend that you force a blank field to print to take up the space.

For example, if the Amount field from the previous example was not included because of an indicator, the Payee field would print in the wrong position. It is the programmer's responsibility to insert blanks in the record to compensate for this. An alternative approach is to insert characters between each field, regardless of whether they are included, to be used as field delimiter characters.

12.2.6 CVTLINDTA parameter in printer files

If you have a printer file that had DEVTYPE(*LINE), and a page definition is specified, the spooled file that is generated remains with DEVTYPE(*LINE) and it contains only the data. The associated spooled file attributes point to the page definition that was requested. At print time, PSF/400 uses the instructions found in the page definition and formats the data accordingly.

Spooled files that are generated with DEVTYPE(*LINE) cannot normally be sent to ASCII printers using Host Print Transform. They also cannot be viewed using the IBM AFP Viewer, which is available as part of Client Access Express.

The CVTLINDTA printer file parameter was added with V5R1 to bypass these two restrictions. By specifying CVTLINDTA(*YES), the line data is processed immediately against the page definition as the spooled file is generated. The resulting spooled file that is generated contains fully resolved AFPDS. This data may be printed on a PCL or PPDS ASCII printer using Host Print Transform and it may be viewed using the IBM AFP Viewer.

Chapter 13. Printing from Java applications

Initially, Java was used to create platform-independent applications for the World Wide Web. However, as with any tool that gains general acceptance, more and more programmers are using Java to develop business applications. These applications require an interface to generate printed output that is easy to use in a program, is flexible, robust, and yet provides a means to generate complex business documents.

The new LineDataRecordWriter class delivers these features. This document describes this new class as well as other print functions that are provided as part of the IBM Toolbox for Java.

For a complete description of all classes, methods, and parameters, refer to the Java information available from the iSeries Information Center on the Internet at:

`http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm`

From the navigation bar, select **Programming-> Java-> IBM Toolbox for Java**. For general information about the classes, select **Access Classes**. For specific programming information, select **Javadocs for IBM Toolbox for Java classes**. You can find the various classes described in this chapter in the *com.ibm.as400.access* package.

Prior to reading this chapter, you must have some experience with Java programming and understand the basic concepts of classes and methods.

13.1 The LineDataRecordWriter class

The LineDataRecordWriter is a new class object that is introduced in IBM Toolbox for Java with V5R1. By using this class, the Java application programmer is mainly only concerned with providing the data fields to the class. The layout of the data on the page is defined external to the application with a form definition and a page definition.

The LineDataRecordWriter class writes a record in line data format, with the name of the record format inserted into positions 1 through 10 of the line data. The record format name corresponds to a LAYOUT command in the page definition. This requires that a page definition using the new record format line data processing be used to format the spooled file that is generated. Record layout page definitions are described in 12.2.4, “Java support for record format line data” on page 269.

The records may be written in one of two formats. The FIXED_LAYOUT_LENGTH format is equivalent to standard output generated by most applications. Here, each field has a specified position and length in the record. Data may be aligned left or right within the designated length.

The VARIABLE_LAYOUT_LENGTH format uses a delimiter character to separate fields. This corresponds to the delimiter used in the LAYOUT command in the page definition.

Java applications usually work using the ASCII data streams. LineDataRecordWriter translates the characters into the coded character set identifier (CCSID) of the iSeries server. The CCSID of the iSeries server is stored in the system value QCCSID.

The printer file being generated must have specific attributes. In Java terms, you must set the following parameters:

- **ATTR_CONTROL_CHARACTER**: Forms Control Character set to `*NONE`
- **ATTR_CONVERT_LINE_DATA**: Convert Line Data set to `*YES` (this not required if you are printing to a device that is configured with AFP(*YES))
- **ATTR_FORM_DEFINITION**: Form definition Integrated File System name
- **ATTR_PAGE_DEFINITION**: Page definition Integrated File System name
- **ATTR_PRTDEVTYPE**: Printer device type set to `*LINE`

13.1.1 Sample Java program description

The listing in this section is a subset of a program from the iSeries Information Center on the Web. Since this program is a self-contained example used for illustrating the programming techniques, the data to be printed is generated within the program and only one record format is used. In a production environment, the data to be printed is a combination of fields entered interactively by a user, extracted from other databases, or calculated with the Java program. A variety of formats would likely be used.

The program illustrates the following Java instructions necessary to set up the environment and to use the LineDataRecordWriter class:

1. The RecordFormat qcustcdt is defined as RecordFormat().
2. The CUSNUM, LSTNAM, BALDUE, and CDTDUE fields are defined within the record format.
3. The print format of the fields is set by defining the length and alignment.
4. The record format ID is set to string CUSTRECID. This prints in the first 10 bytes of the output record as required by record format line data processing.
5. Comments are added to the program to show how a variable layout length record would be defined with a delimiter.
6. This example uses fixed layout length records.
7. The fields are added to the output record format.
8. Sign on to the iSeries.
9. Get the value for CCSID. This is equivalent to the iSeries server value QCCSID. It is used to convert the ASCII data used in the Java environment to EBCDIC, which is used for the LINE data.
10. Define the output queue and printer file parameters.
11. Create a new output spooled file.
12. Create a new instance of the LineDataRecordWriter class called `ldw`.
13. Write the record information into ldw.
14. Close the spooled file.

13.1.2 Sample Java program using the LineDataRecordWriter class

To find the full listing of this program, go to the iSeries Information Center at:

http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm

Select the version/release and language. Using the navigation on the left pane, select **Programming-> Java-> IBM Toolbox for Java-> Javadocs for IBM Toolbox for Java classes**.

There is another, simpler sample program that can be found by selecting **IBM Toolbox for Java-> Access classes-> Data conversion and data description classes-> LineDataRecordWriter**.

```
/////////////////////////////////////////////////////////////////////
//
// LineDataRecordWriter example.  This program uses the line data
// record writer access class to create a line data spooled file
// on the AS/400.
//
// The Source code for this program is not published or otherwise
// divested of its trade secrets, irrespective of what has been
// deposited with the U.S. Copyright Office
//
// This source is an example of using the IBM Toolbox for Java
// "LineDataRecordWriter" class.
//
// This sample code is provided by IBM for illustrative purposes
// only. These examples have not been thoroughly tested under all
// conditions. IBM, therefore, cannot guarantee or imply
// reliability, serviceability, or function of these programs.
//
// All programs contained herein are provided to you "AS IS"
// without any warranties of any kind.  The implied warranties of
// merchantablility and fitness for a particular purpose are
// expressly disclaimed.
//
// IBM Toolbox for Java
// (C) Copyright IBM Corp. 1999
// All rights reserved.
// US Government Users Restricted Rights -
// Use, duplication, or disclosure restricted
// by GSA ADP Schedule Contract with IBM Corp.
//
////////////////////////////////////////////////////////////////////////

import com.ibm.as400.access.*;
import java.io.*;
import java.math.BigDecimal;

public class TestA {

    //Private
    private static int ccsid_           = -1;      // local ccsid variable
    private static AS400 system_        = null;    // the AS/400 system
    private static SequentialFile file_ = null;    // the file

    /**
     **  Create the record field descriptions and record format.
     **/
    public static RecordFormat initializeRecordFormat()
    {
[1]     // Create the record format.
        RecordFormat qcustcdt = new RecordFormat();

        // Create record field descriptions for the record format.
[2]    ZonedDecimalFieldDescription customerNumber =
                         new ZonedDecimalFieldDescription(new AS400ZonedDecimal(6,0),
                                                          "CUSNUM");
        CharacterFieldDescription lastName =
                         new CharacterFieldDescription(new AS400Text(8, ccsid_, system_),
                                                   "LSTNAM");

        ZonedDecimalFieldDescription balanceDue =
                         new ZonedDecimalFieldDescription(new AS400ZonedDecimal(6,2),
```

```
                                                    "BALDUE");
    ZonedDecimalFieldDescription creditDue =
                  new ZonedDecimalFieldDescription(new AS400ZonedDecimal(6,2),
                                                    "CDTDUE");

    // assign constants from FieldDescription class
    int justLeft = FieldDescription.ALIGN_LEFT;
    int justRight = FieldDescription.ALIGN_RIGHT;

[3] // set the length and alignment attributes for writing the fields
    // The length indicates how many characters the field is, and
    // justification indicates where in the layout field the data
    // should be placed.
    customerNumber.setLayoutAttributes(10,justLeft);
    lastName.setLayoutAttributes(10,justLeft);
    balanceDue.setLayoutAttributes(10,justRight);
    creditDue.setLayoutAttributes(10,justRight);

[4] // set the record format ID
    String d = "CUSTRECID";
    qcustcdt.setRecordFormatID(d);

[5] // if this were a variable field length record,
    // we would set the type and delimiter accordingly.  We
    // also would not have needed to specify layoutLength and
    // layoutAlignment values.
    // qcustcdt.setRecordFormatType(RecordFormat.VARIABLE_LAYOUT_LENGTH);
    // qcustcdt.setDelimiter(';');

[6]  // set the record type to fixed field length
    qcustcdt.setRecordFormatType(RecordFormat.FIXED_LAYOUT_LENGTH);

[7] // add the field descriptions to the record format.
    qcustcdt.addFieldDescription(customerNumber);
    qcustcdt.addFieldDescription(lastName);
    qcustcdt.addFieldDescription(balanceDue);
    qcustcdt.addFieldDescription(creditDue);

    return qcustcdt;
}


/**
 ** Creates the actual record with data
 **/
public static void createRecord(Record record)
{
    record.setField("CUSNUM", new BigDecimal(323));
    record.setField("LSTNAM", "Johnson");
    record.setField("BALDUE", new BigDecimal(25.00));
    record.setField("CDTDUE", new BigDecimal(0.00));
}


public static void main(String[]args) {

[8]    // create an instance of the AS/400 system
    system_ = new AS400("SYSTEMA", "JOE", "PGMR");

[9]    // create a ccsid
    ccsid_ = system_.getCcsid();

[10]    // create output queue and specify spooled file data to be *LINE
    OutputQueue outQ = new OutputQueue(system_, "/QSYS.LIB/QUSRSYS.LIB/LDRW.OUTQ");
    PrintParameterList parms = new PrintParameterList();
    parms.setParameter(PrintObject.ATTR_PRTDEVTYPE, "*LINE");
    parms.setParameter(PrintObject.ATTR_PAGDFN,"/QSYS.LIB.QUSRSYS.LIB/LDRW.PAGDFN");
    parms.setParameter(PrintObject.ATTR_CONVERT_LINEDATA,"*YES");


    // initialize the record format for writing data
    RecordFormat recfmt = initializeRecordFormat();

    // create a record and assign data to be printed...
    Record record = new Record(recfmt);
    createRecord(record);

    SpooledFileOutputStream os = null;
```

```
[11]    try {
            // create the output spooled file to hold the record data
            os = new SpooledFileOutputStream(system_, parms, null, outQ);
        }

        if (os != null) { // Output stream was created successfully!
            LineDataRecordWriter ldw;
            try {
[12]            // create the line data record writer
                ldw = new LineDataRecordWriter(os, ccsid_, system_);

[13]            // write the record of data
                ldw.writeRecord(record);
            }
            catch (IOException e) {
                System.out.println("Error occurred writing record data");
            }

[14]        // close the output stream (spooled file)
            try {
                os.close();
            }
            catch (Exception e) {
                System.out.println("Error occurred closing output stream.");
            }
        }
    }
}
```

13.2 Java Report Builder

The Java report writer classes are a set of classes that Java applications can use to access and format application data from an eXtensible Markup Language (XML) source file or data produced by servlets or JavaServer Pages (JSP). These classes are included in the licensed program for IBM Toolbox for Java (5722-JC1) Version 5 Release 1 (V5R1).

The Extensible Style Language (XSL) formatting objects are used as the language for defining how the application data is to be formatted into a document. The formatting objects are defined within an XSL stylesheet when formatting XML data. The formatting objects are contained within a Java servlet or JSP when formatting data generated from a servlet or JSP. The report writer classes can output the formatted data in one of two document formats, including HP PCL and Adobe's Portable Document Format (PDF).

An example of how Jave Report Builder works is shown in Figure 181.

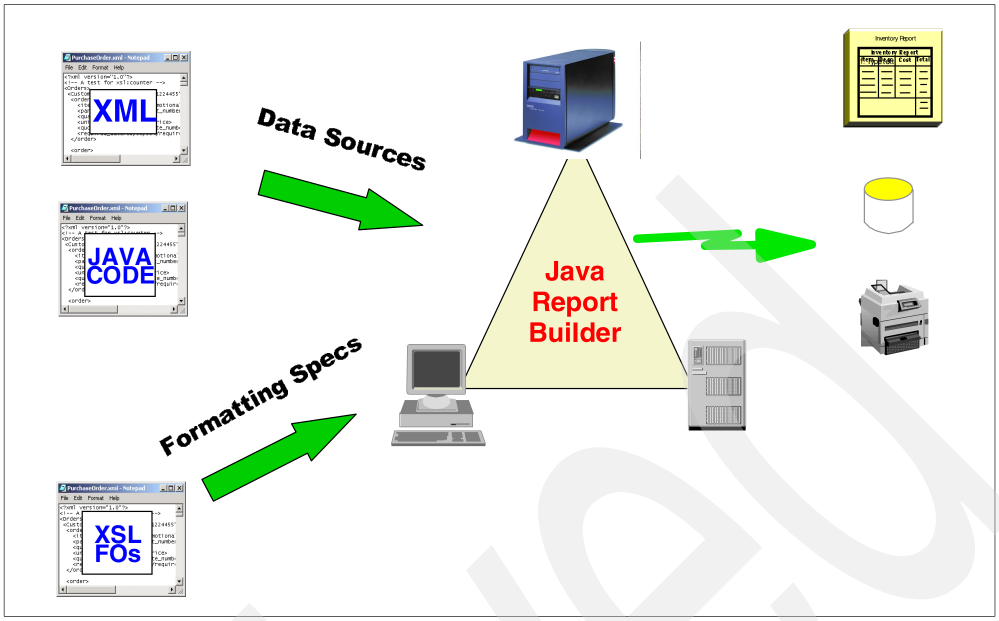


Figure 181. Java Report Builder

The report writer classes are contained in three different but related packages:

- com.ibm.as400.util.reportwriter.pclwriter
- com.ibm.as400.util.reportwriter.pdfwriter
- com.ibm.as400.util.reportwriter.processor

The classes in the processor package are used by the application to specify the locations of the XML data, the XSL stylesheets, and the servlets/JSPs used within an application, and to initiate the actual formation of the output document. The XSLReportProcessor class in this package is used to process XML data with XSL stylesheets. The JSPReportProcess class is used to retrieve data from servlets and JSP pages and format the data.

The context classes (in the pclwriter and the pdfwriter packages) define methods that the ReportProcessor classes need to render XML and JSP data in the chosen format. The PCLContext class in combination with a ReportWriter class is used to generate a report in the Hewlett Packard Printer Control Language (PCL) format. The PDFContext class in combination with a ReportWriter class is used to generate a report in the Adobe Portable Document Format (PDF).

To create a formatted document from the application data, the report writer classes use the XSL formatting objects contained within the XSL stylesheet, the servlet, and the JSP. These formatting objects are defined in the second part of the XSL language, used for expressing stylesheets, as the XML vocabulary for specifying formatting semantics. A description of the formatting objects can be found in the W3C XSL Version 1.0 specification at: `http://www.w3.org/TR/xsl/`

An application programmer or document designer can create the XSL stylesheet or JSP, containing the XSL formatting objects, with any text editor, XSL stylesheet, or JSP editor.

13.3 Creating SCS spooled files

There are a number of classes provided in the IBM Toolbox for Java that may be used to generate SCS spooled files. Each one supports slightly different print characteristics according to the actual device being written to, or emulated, ranging from SCS5256Writer, which is the simplest, to SCS3812Writer, which supports the widest range of function.

It is the programmer's responsibility to set all appropriate parameters for the data stream, including new lines and page eject codes.

For more information on using these classes, and for programming examples, see the Javadoc in the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm`

Select the version/release and language. Using the navigation on the left pane, select **Programming-> Java-> IBM Toolbox for Java**.

13.4 Creating a spooled file from stream data

If a fully composed data stream is available in a format that is available to a Java program, such as in the IFS, the SpooledFileOutputStream class can be used to generate an iSeries spooled file.

Using this class would be synonymous with using the Print API QSPPUTSP that can be used from other high level languages.

It is not usually expected that a programmer would generate the data stream from scratch. However, if the print data stream was generated using some other method, this class can be used to create an iSeries spooled file.

13.5 Other related classes

There are a series of print classes that allow the manipulation of iSeries print objects in the IBM Toolbox for Java. Print objects include spooled files, output queues, printers, printer files, writer jobs, and AFP resources. The AFP resources include fonts, form definitions, page definitions, overlays, and page segments.

The classes for print objects are organized in a base class (PrintObject) with subclasses for each of the six types of print objects. The Javadocs found on the iSeries Information Center provide details of the methods and attributes specific to each of these. You can find them at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm`

Select the version/release and language. Using the navigation on the left pane, select **Programming-> Java-> IBM Toolbox for Java-> Javadocs for IBM Toolbox for Java classes**.

You can find an example of using the AFPResource class to extract the contents of iSeries AFP resources in Appendix C, "Extracting AFP resource contents" on page 331.

Chapter 14. End-to-end example

This chapter examines a case study of a customer who wants to use some of the new functions available for print and presentation to e-mail personalized invoices to their customers. Assume you are an employee for the Super Sun Seeds Company and it is your responsibility to implement their e-business strategy, including delivering output electronically.

The Super Sun Seeds Company currently produces invoices on an IBM laser printer. Infoprint Designer is used to generate two copies, one for the customer and one for packing. The company has decided to take this one step further. As part of their e-business strategy, the Super Sun Seeds Company wants to deliver the invoices to their customers electronically, as a Portable Document Format (PDF) attachment in an e-mail. You need to devise a method to take the output from their current application, split it into individual invoices, and match each one with the corresponding e-mail address.

This chapter describes the steps to produce the desired results, including:

- Modifications to the Infoprint Designer project
- A single, one-time modification to the application
- Using CRTAFPDTA to index the data
- Using PRTAFPDTA to re-spool the indexed file
- Displaying the file using the AFP Viewer
- Creating a PSF configuration object and an output queue for PDF conversion
- Creating an exit program to look up e-mail addresses
- Monitoring for undeliverable mail
- Other maintenance tasks

This chapter is intended for system analysts, developers, and programmers. Certain iSeries server skills are assumed. Programming skills in RPGLE are required to implement the user exit program.

The products used in the example are:

- PSF/400 for iSeries (V5R1) (This is only needed if the documents are printed to an IPDS printer along with being sent electronically.)
- Infoprint Server for iSeries
- Infoprint Designer for iSeries
- A product capable of reading e-mails, such as Lotus Notes or Microsoft Outlook
- Adobe Acrobat Reader
- AFP Viewer, within Client Access Express/400

14.1 Overall flow of the end-to-end application

The overall flow of the data from the application generating the original spooled file to an e-mail being sent to a customer is shown in Figure 182.

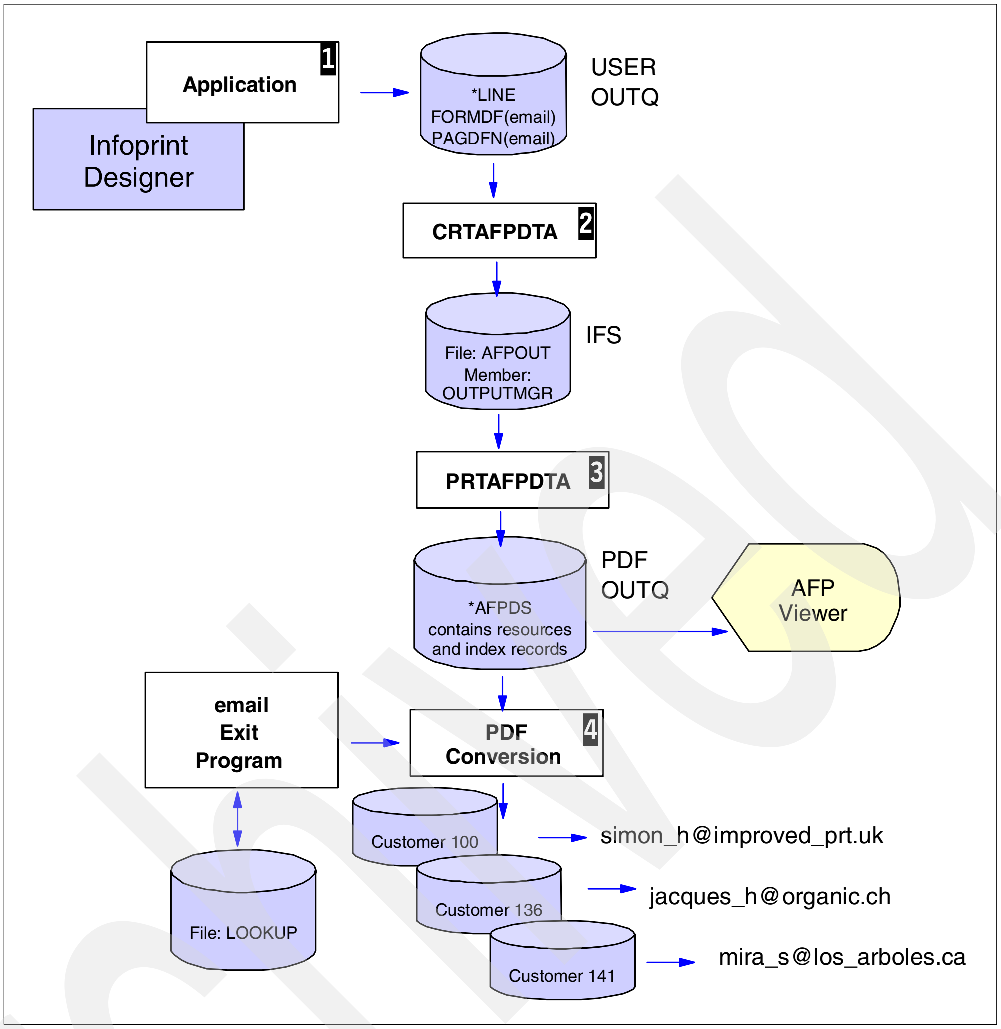


Figure 182. From application to e-mail

The process shown in Figure 182 is explained in the following steps:

1. The application produces the data that is spooled to a printer file defined as DEVTYPE(*LINE). The printer file also references the form definition and page definition that were created using Infoprint Designer for iSeries.
2. The CRTAFPDTA command is run against this spooled file. This creates a physical file member that contains the resolved AFPDS spooled file, the external resources, and the index records.
3. The PRTAFPDTA command is used to create a *new* spooled file that is in AFPDS format. The new spooled file now contains the all the external resources and index records.
4. This new spooled file is moved to the PDF conversion output queue and Infoprint Server converts it to PDF format.
5. The user exit program for e-mail does a lookup operation to the LOOKUP file using the information in the index records to select the appropriate e-mail address for each customer. Each separate invoice is sent to a different destination.

This chapter describes each of these steps and any one-time setup tasks.

14.2 Infoprint Designer changes

The needs of an e-mail application are slightly different than the original printed application. Some changes to the Infoprint Designer project are made to accommodate this new means of output.

14.2.1 Creating a new Infoprint Designer project for e-mail

To create the e-mail project, start with the COND.prj, which uses conditional processing to reformat the marketing message. (For details on the COND.prj, see 2.9.3.1, “Concepts of conditional processing” on page 62.) The invoices that are sent via e-mail do not need the second copy that was used for a packing slip, and they don’t need the Terms and Conditions that would have printed on the back. Remove the extra Packing Slip subgroup from the form definition, redefine the copy group to print simplex, and eliminate the terms and conditions subgroup.

Store this project, the new form definition, page definition, and overlay under the name EMAIL. (These resources are *not* included in the sample Infoprint Designer projects included with the product.)

You must perform this task only once.

14.2.2 Using the new Infoprint Designer objects

As normal, to use the new resources that you just created using Infoprint Designer, the printer file must be changed with the Change Printer File (CHGPRTF) command, or overridden at run time with the Override with Printer File (OVRPRTF) command. This creates LINE data and references the new form definition and page definition.

An example of the command to change the printer file (here called INVSCS) is:

```
CHGPRTF PRTF(INVSCS)
         DEVTYPE(*LINE)
         FORMDF(EMAIL)
         PAGDFN(EMAIL)
         DUPLEX(*FORMDF)
         DRAWER(*FORMDF)
```

If you change the printer file, it only must be done once. If you choose to override the printer file instead, you must insert that command into the CL program that runs the application. Ensure that the AFP resources are in a library within your library list.

There are some additional changes that must be made to the Super Sun Seeds invoice program to accommodate the indexing function. They are described in 14.3.4, “Changing the invoicing program to accommodate the index requirements” on page 287.

One final change to the application is to change the output queue to which the spooled file is directed. In the past, you directed the output to a queue that was directly associated to a physical printer. We recommend that the new e-mail spooled file be directed to an output queue that is not associated with an active printer writer.

14.3 Planning for the index function

For more information on the index function of the CRTAFPDTA command, see Chapter 5, “Infoprint Server for iSeries: CRTAFPDTA” on page 149.

Super Sun Seeds invoices are produced using the INVSCS program. A copy of this program is shipped with the Infoprint Designer for iSeries product. You can find it in the IPDATA library. See *Infoprint Designer for iSeries: Getting Started*, G544-5773, for information on the IPDATA library and using the programs in it.

14.3.1 Why CRTAFPDTA is used

To send individual invoices within a single spooled file to different addresses, you must add index records to the spooled data. Chapter 4, “Using the Infoprint Server for iSeries PDF transform” on page 103, and the *Infoprint Server for iSeries User’s Guide*, G544-5775, describe how to use the DDS keywords STRPAGGRP and ENDPAGGRP to add index records. However, these keywords require that the file be generated with DEVTYPE(*AFPDS), and cannot be used with a printer file that uses page definitions and is generated with DEVTYPE(*LINE).

Super Sun Seeds is using Infoprint Designer for iSeries to develop this application. The layout component of Infoprint Designer produces page definitions that must be used with line data. Consequently they need to find an alternative to those DDS keywords. The CRTAFPDTA command, which is also part of Infoprint Server for iSeries, has the ability to generate index records for a spooled file that contains line data.

14.3.2 Setting up a target file for CRTAFPDTA

The CRTAFPDTA command generates AFPDS files in an Integrated File System (IFS) directory. The index entries generated within the spooled file are used by the PDF generator to separate the one large spooled file to individual files for each customer and to look up the e-mail address based on the customer number.

The CRTAFPDTA command directs its output to any directory in the IFS that you specify. However, in the next step of this process, PRTAFPDTA expects the data to be in a physical file. Before you run the CRTAFPDTA command, you must create the physical file that is to receive the output. You can create the file once, and then clear it out each time. Or, you can create a new file each time with a different name that identifies the run date or some other meaningful information.

This example shows how to create the file called AFPOUT in library MYLIB. The record length is not critical, but it must be specified. You must specify LVLCHK(*NO) to prevent getting a Level Check error when you run the PRTAFPDTA command. You must allow for at least four new members to be added to the file:

```
CRTPF   FILE(MYLIB/AFPOUT)
         RCDLEN(80)
         LVLCHK(*NO)
         MAXMBRS(*NOMAX)
```

You can enter the IFS path to this physical file in each of the TOIDXSTMS, TOMRGSTMF, TORSCSTMF, and TOSTMF parameters in the CRTAFPDTA command. However, you may find it easier to make the AFPOUT file accessible as your current directory and then indicate in each of those four parameters that you want the output to go to *DFT. This is done using CHGCURDIR. You can reference the file by its path in the IFS:

```
CHGCURDIR DIR('/QSYS.LIB/MYLIB.LIB/AFPOUT.FILE')
```

This step is included in the job that is run each time you want to perform the e-mail function.

14.3.3 Requirements of the CRTAFPDTA index function

The indexing function essentially looks for two types of information: the trigger fields and the index fields.

The combination of fields defined as the trigger tell the indexing function where each new customer invoice should begin. This allows for a different number of pages per invoice. The first of the trigger fields defined has a second role of being an *anchor point*. This means that all other trigger fields and index fields are defined by their relative location with respect to the first trigger field, or anchor point.

The index fields are used to identify a distinct piece of data in each separate invoice. The combinations of fields used for the index are used by the e-mail exit program to look up the actual e-mail address of the recipient of the invoice. In this example, you use only one index field (the customer number).

14.3.4 Changing the invoicing program to accommodate the index requirements

In the case of Super Sun Seeds, one identifying feature to start each new invoice is the fact that "Page 1" prints on the bottom of the first page of each invoice, always on line 63. You may think that this would be the ideal trigger point or anchor. However, depending on how many of the invoice detail lines actually have data on them, there may be a different number of "records" in the spooled file before the record that has the page number on it.

Consequently, CRTAFPDTA cannot use this field as the index trigger. This is a common occurrence when the application uses DDS keywords SKIPA or SKIPB, or other similar language instructions, to skip over an unknown or variable number of lines.

To work around this problem, the application program and associated DDS are modified slightly to output the word "NEW" on the same line as the company name on the first page of a new invoice. This record line is always in the same relative position on the page, and the customer number is always four records after it.

> **Note**
>
> These programming changes may not be necessary for all applications. If the trigger fields and index fields are always in the same relative positions to each other in your application, you do not have to modify the program.

You did not map the word "NEW" when you redesigned the Infoprint Designer project as described in 14.2.1, "Creating a new Infoprint Designer project for e-mail” on page 285. Therefore, it does not show on the final document that is delivered to the customer.

When planning for the index function, you may find it helpful to copy the spooled file to a physical file using the Copy Spooled File (CPYSPLF) command and specifying CTLCHAR(*FCFC). Create the physical file with a record length one record longer than that of the spooled file, for example 81 or 133 record length for original spooled files that have used the common length of 80 or 132. The commands you use to do this are:

```
CRTPF FILE(mylib/CPYSPLF) RCDLEN(133) MAXMBRS(*NOMAX)
CPYSPLF FILE(INVSCS) TOFILE(mylib/CPYSPLF) JOB(number/user/name) SPLNBR(1)
TOMBR(INVSCS) CTLCHAR(*FCFC)
```

If you do not know the job information for your spooled file, use the F11 key from either a WRKSPLF or WRKOUTQ display.

If you view the file with the Display Physical File Member (DSPPFM) command, you see a good representation of how the spooled data is presented to the CRTAFPDTA process. Figure 183 shows the first page of the Super Sun Seeds invoice after the modification to add the word “NEW” at the beginning of each invoice. This becomes the first trigger field and is represented in the CRTAFPDTA command as `IDXTRG((* 3 'NEW'))`. This tells the indexing program to look in any record, starting in position 3 for the string “NEW”. Note that the carriage control byte in column 1 is included in the position count. Note also that you must specify an asterisk `(*)` for the record number value of the first trigger field.

```
                         Display Physical File Member

File . . . . . . :   CPYSPLF            Library  . . . . :   MSHNIER
Member . . . . . :   INVSCS             Record . . . . . :   1
Control  . . . . .                      Column . . . . . :   1
Find . . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8.
1
-
-
-
0 NEW       IMPROVED PRINTING CORP                 SAME
            PERFORMANCE BOULEVARD
            PRINTERSVILLE
            CO  45789-2637
-                100             31300        3/28/01            4/28/01
0             BEST WAY           3/28/01      NET 30          YOUR PRINTER REP
-
           1 CT   00000300  HIGH ALTITUDE WATERMELON        1.01          1.01
           1 PK   01100517  SPARTAN SEEDS                   2.39          2.39
           9 PK   04569870  NORTHERN LITE BLUE SPRUCE     858.32      7,724.88
          12 BX   11005004  BUSH GREEN SEEDS                2.50         30.00
          12 CT   11005011  LASSO RED SEEDS               892.23     10,706.76
          26 PK   11005018  EARLY BANTAM SEEDS               .38          9.88
           5 BX   11057893  AFRICAN DAISY, SEEDS            2.35         11.75
```

*Figure 183. Sample spooled file after CPYSPLF with CTLCHAR(*FCFC)*

The customer number appears four records below the trigger field. It is represented in the CRTAFPDTA command as `IDXTAGFLD((4 15 6))`. This tells the indexing program to look four records *after* the anchor point set by the first trigger field. Then, starting in position 15, the program uses the next six characters as the first index. The customer number, as shown in the example, appears to be only three characters long. However, by inspecting the DDS for the application, you can see that it can have up to six characters. If you set the index based only on the three characters visible in the one example, you may have problems at a future time. This shows where it is important to have the original application documentation available.

Be aware of the fact that CRTAFPDTA looks at the relative positions of the *records* in the spooled file, not the relative lines that they may end up printing. The word “NEW” and the customer number are four records apart in the file, but print 6 lines apart, on lines 12 and 18 respectively. This is because there is a carriage control character “-” in position 1 of the record containing the customer number, which inserts a triple space at print time.

This step, if necessary, is done as part of the one-time setup.

14.3.5 Running the CRTAFPDTA command

Now that you have set up a destination for the output of the CRTAFPDTA command, and have identified the trigger fields and index fields, you are ready to run the CRTAFPDTA command as shown below. The four files that are generated by this command are placed in the default directory, which is set up to be AFPOUT, as described in 14.3.2, “Setting up a target file for CRTAFPDTA” on page 286:

```
CRTAFPDTA  FROMSPLF(INVOICE)
            TOSTMF(*DFT)
            JOB(number/user/name)
            SPLNBR(*LAST)
            FORMDF(QGPL/EMAIL)
            PAGDFN(QGPL/EMAIL)
            TOIDXSTMF(*DFT)
            TORSCSTMF(*DFT)
            IDXTRG((* 3 'NEW'))
            IDXTAGFLD((4 15 6))
            IDXTAG(('Customer Number' (*IDXTAGFLD1)))
            RSCDTA(*FORMDF)
            TOMRGSTMF(*DFT)
```

After the command is run, there are four members in the physical file AFPOUT:

- **OUTPUTAFP**: The resolved AFPDS document
- **OUTPUTIDX**: The index records
- **OUTPUTRSC**: The external resources
- **OUTPUTMRG**: Merged output; this contains the information from the other three members

These names are automatically assigned by CRTAFPDTA because a value of *DFT was specified for each of the TOxxxSTMF variables. It is possible to specify different names for these files. See Chapter 5, “Infoprint Server for iSeries: CRTAFPDTA” on page 149, for more information.

You use the member OUTPUTMRG. It contains the AFPDS data, along with the form definition resource and the index records. The form definition is needed inline to view the spooled file using the AFP Viewer, as shown in 14.4.2, “Checking your file with the AFP Viewer” on page 291.

This step must be included in the job that runs for each e-mail.

14.4 Respooling the file using PRTAFPDTA

You cannot use the output of the CRTAFPDTA command directly by the PDF conversion process until you regenerate it back as a spooled file. The system command, Print AFP Data (PRTAFPDTA), is included with OS/400 that you use to do this:

```
PRTAFPDTA  FILE(AFPOUT)
            MBR(OUTPUTMRG)
            DEV(MYOUTQ)
            FORMDF(*INLINE)
            FIDELITY(*CONTENT)
```

The prompt for the parameter DEV implies that a physical device must be named as the destination for this command. An unqualified name of an output queue works just as well and may be preferred.

The form definition that was captured in the OUTPUTRSC member is included inline with this command. This facilitates using it with the AFP Viewer. See 14.4.2, “Checking your file with the AFP Viewer” on page 291.

The spooled file that is generated from the PRTAFPDTA command has the same name as the member used to create it (in this case, OUTPUTMRG). This same name is ultimately used in the subject line of the e-mail. You may choose to rename the member prior to running the PRTAFPDTA command:

```
RNMM FILE(AFPOUT)
      MBR(OUTPUTMRG)
      NEWMBR(EINVOICE)
```

This step must be included in the job that runs for each e-mail.

14.4.1 Directing the output of PRTAFPDTA

You may choose to direct the output of the PRTAFPDTA step directly to the output queue being used by the PDF process. However, it may be prudent to direct it to a temporary output queue and change the attribute to SAVE(*YES) as you move it to the PDF output queue.

14.4.1.1 Changing the USRDTA parameter

You may also add some information to the User Data (USRDTA) attribute this time. The USRDTA parameter is picked up in the e-mail exit program and you may find a use for it there. For example, you may use this field to look up a greeting message to include in the body of the email. In doing so, the program does not have to be changed if the message changes.

To move the spooled file to the PDF output queue and to set other parameters as described in this section, use the Change Spooled File Attributes (CHGSPLFA) command.

```
CHGSPLFA    FILE(EINVOICE)
            JOB(jobno/user/jobname)
            SPLNBR(n)
            OUTQ(*LIBL/PDF)
```

```
SAVE(*YES)
USRDTA('HelloWorld')
```

This step must be included in the job that runs for each e-mail.

14.4.2 Checking your file with the AFP Viewer

This may be a good time, especially while the application is under development, to verify that your index selection is correct. You can do this using the AFP Viewer. Start Operations Navigator from Client Access or Client Access Express. If you do not already have a session with your iSeries server, start the connection and log on. Select **Basic Operations** and then click **Printer Output**. Select your new spooled file from the list.

The index function is illustrated when by selecting **Search-> Find group**. The complete index generated by the CRTAFPDTA is a combination of the customer number and a document sequence number. Figure 184 shows an example of what should appear.

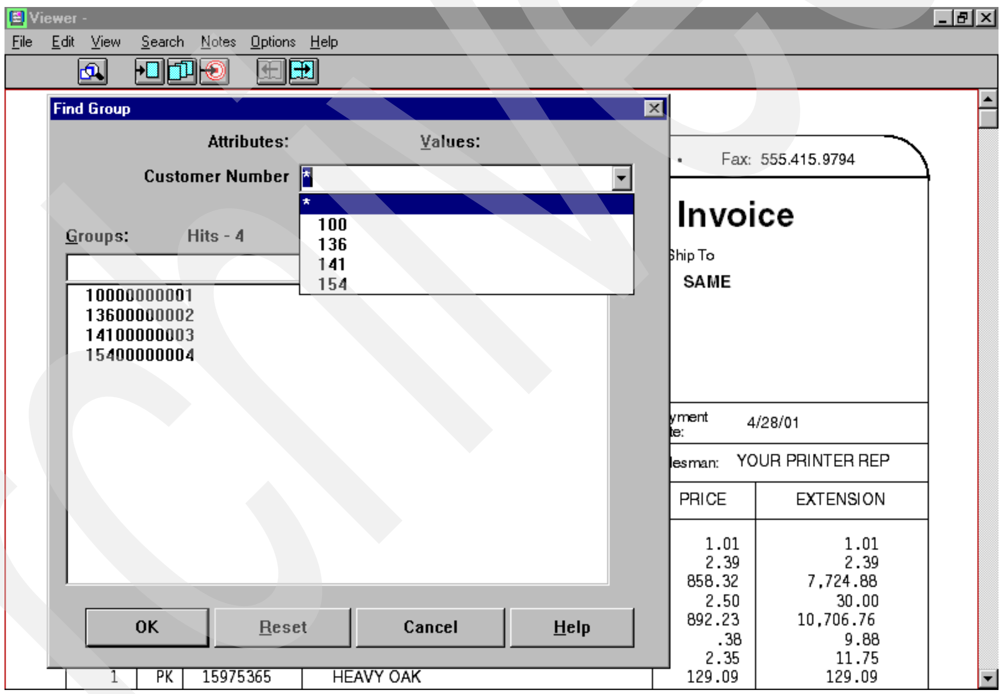


Figure 184. Using the AFP Viewer with an indexed spooled file

This step is helpful during the initial setup and testing of the e-mail process. It may also be used at any time with the production files.

14.4.2.1 Font mapping with AFP Viewer

When using the AFP Viewer, you may find you see messages indicating that it cannot find the requested fonts. This is because Infoprint Designer references fonts by what is known as the *Alternate Coded Font Identifier*, where the AFP Viewer only maps the standard Coded Font names. To rectify the situation, the font mappings for the missing fonts must be appended to the bottom of the ICODED.FNT file that is used by the AFP Viewer.

For the Super Sun Seeds case study, add the following entries to the table:

```
XZ4010=CZ4200,T1V10037
XZH0F0=CZH400,T1V10500
XZH0E0=CZH300,T1V10500
XZH010=CZH200,T1V10037
XZ5010=CZ5200,T1V10037
XZN0E0=CZN300,T1V10500
```

If you find that you are missing entries for other coded fonts, you can determine the corresponding character set and code page by using the Work with Font Resources (WRKFNTRSC) command and looking at the description of the iSeries coded font object as shown in Figure 185.

```
                  Display Font Resource Attributes
                                                System:   BLDASL1
Font Resource  . . . : XZN0E0
  Library  . . . . . :   QFNT10
Object attribute . . : CDEFNT
Text . . . . . . . . : Charset CZN300 and Codepage T1V10500 at 10-pt
```

Figure 185. Displaying the description of a coded font object

This step is done as part of the one-time setup.

14.5 Converting the spooled file to PDF and sending it as an e-mail

There are a number of steps to do when preparing to convert a spooled file to a PDF file and send it to the appropriate destination via e-mail.

14.5.1 Enabling OS/400 to send e-mail

There are some system tasks the must be done to enable the iSeries server to send e-mail. These are described in Appendix A of the *Infoprint Server for iSeries User's Guide*, G544-5775.

If your iSeries server is not already set up for e-mail, you must do this step as part of the one-time setup.

14.5.2 Configuring a PDF device and PSF configuration object

As described in 4.3, “Setting up your PDF virtual printer” on page 108, the PDF creation function depends on a Printer Device Description with the appropriate PSF configuration object being configured. Refer to that section for additional details on the parameters.

For the Super Sun Seeds e-mail function, the Device Description and PSF configuration object are created as follows:

```
CRTDEVPRT  DEVD(PDF)
            DEVCLS(*LAN)
            TYPE(*IPDS)
            MODEL(0)
            LANATTACH(*IP)
            AFP(*YES)
            PORT(5013)
```

```
FONT(416 12)
RMTLOCNAME('127.0.0.1')
USRDFNOBJ(QGPL/PDF *PSFCFG)
TEXT('Device Description for PDF email')
```

The DEVD, DEVCLS, TYPE, MODEL, LANATTACH, AFP, and FONT parameters are assigned values as if you were setting up an IPDS printer on your local area network:

- **PORT:** Use a unique four-digit number to use as a TCP/IP Port number for each PDF virtual printer.
- **RMTLOCNAM** (Remote Location Name): Specify a loopback address or a name for a loopback address. A valid loopback address must have 127 as the first octet.
- **USRDFNOBJ** (User Defined Object): Specify the name and library of a PSF configuration object. The object type is *PSFCFG. If PSF configuration object does not exist when you create the device description, you receive a warning message. This is not a problem provided you create the object before you attempt to start the writer.

```
CRTPSFCFG  PSFCFG(QGPL/PDF)
            PDFGEN(*MAIL)
            PDFDEVTYPE(*IP40300)
            PDFMULT(*YES)
            PDFSENDER(MSHNIER)
            PDFDTAQ(QGPL/PDFDTAQ)
            PDFMAPPGM(QGPL/EMAIL)
            TEXT('PSF configuration for PDF email')
```

- **PSFCFG:** PSF configuration object name. This name is referenced by the PDF device.
- **PDFGEN** (PDF Generation): This parameter tells the PDF Generation subsystem that you want to e-mail the PDF files.
- **PDFDEVTYPE** (PDF Device Type): This tells the PDF subsystem to use functions available on an IBM Infoprint 40, at 300 dots per inch. This is the recommended setting for this parameter.
- **PDFMULT** (Generate multiple PDF files): This tells the PDF subsystem to look for the embedded tag records and to generate multiple PDF e-mails accordingly.
- **PDFSENDER** (PDF Sender): This parameter identifies the user that is sending the e-mails. This user must be enrolled in the iSeries Directory.
- **PDFDTAQ** (PDF Data Queue): Specifies the name of the data queue where PSF/400 will log the IPDS to PDF transformation completion notifications. This parameter is optional. See 14.6.1, “Using the PDF data queue” on page 301, for additional information.
- **PDFMAPPGM** (PDF Mapping Program): This tells the PDF subsystem to run the user to obtain the valid e-mail addresses.

This step is done as part of the one-time setup.

14.5.3 Creating an e-mail exit program

As described in 14.3.5, “Running the CRTAFPDTA command” on page 289, the CRTAFPDTA command causes index records that contain the customer number for each invoice to be inserted into the spooled file at the appropriate places. The e-mail exit program that is named in the PSF configuration object for the PDF transform uses this information to do a lookup on an iSeries database file that contains customer information, including their e-mail address.

You can find a description of the requirements for the input and output parameters and data structures for this program in Appendix E, “Supplemental Infoprint Server information” on page 343.

As an experienced RPG programmer at the hypothetical Super Sun Seeds Company, you choose to write the e-mail exit program in RPGLE. A sample program listing can be found in Appendix E.1, “E-mail exit program” on page 343. A sample C language program is also included in the *Infoprint Server for iSeries User’s Guide*, G544-5775.

Note that the e-mail Address field in the e-mail output information buffer may be defined as any length up to 16M. A length of 255 characters is adequate for the sample program. If you are sending each e-mail to multiple destinations you may need to make that field longer.

The customer number that is found in the index records is passed into the program as the first six characters of the MAILTAG field. The program extracts those characters and performs a CHAIN operation to the LOOKUP file to retrieve the corresponding e-mail address.

Figure 186 shows an example of the DDS used to create the LOOKUP physical file.

```
A                                   UNIQUE
A          R EMAILR
A            CUSTNO         6A      COLHDG('CUSTOMER' 'NUMBER')
A            COMP          25A      COLHDG('COMPANY' 'NAME')
A            PERSON        25A      COLHDG('CONTACT')
A            EADDR         80A      COLHDG('E-MAIL' 'ADDRESS')
A          K CUSTNO
```

Figure 186. DDS for e-mail LOOKUP file

Figure 187 shows a sample of a few records from the e-mail LOOKUP file. Note that the e-mail addresses are surrounded by single quotes.

```
                       Display Physical File Member
File . . . . . . :   LOOKUP              Library  . . . . :   QGPL
Member . . . . . :   LOOKUP              Record . . . . . :   1
Control  . . . . .                       Column . . . . . :   1
Find . . . . . . .
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....
   100Improved Printing Corp   Simon Hodkin           'simon_h@improved_prt.uk'
   136Organic Garden Supplies  Jacques Hofstetter     'jacques_h@organic.ch'
   141Los Arboles Del Mundo    Mira Shnier            'mira_s@los_arboles.ca'
   154The Last Leaf            Bill Shaffer           'bill_s@last_leaf.com.'
                          ****** END OF DATA ******
```

Figure 187. Sample records in e-mail LOOKUP file

If the CHAIN operation to the LOOKUP file is successful, the Disposition (DISPOSTN) field is assigned a value of 1, indicating that the e-mail should be sent. The e-mail address from the LOOKUP file (EADDR) is moved into the ADDRESS field. A short message is built, including the company name and the contact person, and is moved into the MSGTEXT field. The program ends and these values are returned to the PDF writer and are used to generate the electronic mail and attachment.

If the lookup fails, control is transferred to the RPGLE subroutine called "Not_Found". The disposition field is set to "0", indicating that the e-mail is not to be sent. A CL program, NOTFOUND, is called. This program performs two actions:

1. It sends a message to a user that contains the customer number that failed and the name of the PDF file that contains that customer's invoice.
2. It copies the PDF file from the temporary location in QDLS to a user-specified folder in the IFS.

   This second step must be done because the e-mail PDF process cleans up all of the temporary files at the end of the job, regardless of whether they are successfully sent.

You can find a listing for the NOTFOUND program in E.1.2, "NOTFOUND program called by e-mail" on page 344. Figure 188 illustrates a sample message that is issued by the NOTFOUND program. Note the structure of the PDFFILE name. The syntax for this name is described in the *Infoprint Server for iSeries User's Guide*, G544-5775.

```
                              Display Messages
                                                       System:   BLDASL1
Queue . . . . . :   MSHNIER                 Program . . . . :   WSMSGQ
  Library . . . :     QUSRSYS                 Library . . . :     AFP
Severity  . . . :   00                      Delivery  . . . :   *HOLD

Type reply (if required), press Enter.

  From  . . . :   QSPLJOB        06/17/01   17:48:37
  Cannot find e-mail address for customer number    136 .  File
    /QDLS/PSF/092962/QPRTJOB/066591/MSHNIER/000004/OUTPUTMR/
    06172001/000002.PDF not sent.
```

Figure 188. Sample message issued by the NOTFOUND program

This step is done as part of the one-time setup.

14.5.3.1 Using USRDTA in the e-mail exit program

In this example, the exit program looks up the e-mail addresses based on the customer number passed to it in the index records. You may also find a similar exit program can be used to e-mail individual spooled files.

The USRDFNDTA parameter in the printer file can be used to specify an e-mail address by entering a value for MAILTAG. This is described in Chapter 4, “Using the Infoprint Server for iSeries PDF transform” on page 103. However, this parameter may only be set when the printer file is created with CRTPRTF or by using an override printer file (OVRRPRTF). If the spooled file already exists, you cannot use the Change Spooled File Attributes (CHGSPLFA) command to change the value of USERDFNDTA.

One solution is to change the USRDTA field for the spooled file. This field is passed as input to the e-mail exit program and can easily be set up as the search key for a lookup function to find the associated e-mail address.

14.5.4 Sending the spooled file as a PDF

You are now ready to send the spooled file as an e-mail. Move the spooled file to the output queue associated with the PDF device description and start the writer.

Once the file arrives in the output queue for the PDF writer, all the conversion and e-mail tasks are done as defined in the PSF configuration object for that output queue. See 14.5.2, “Configuring a PDF device and PSF configuration object” on page 292.

Figure 189 shows an example of the file as it appears in the mailbox of a Lotus Notes user.

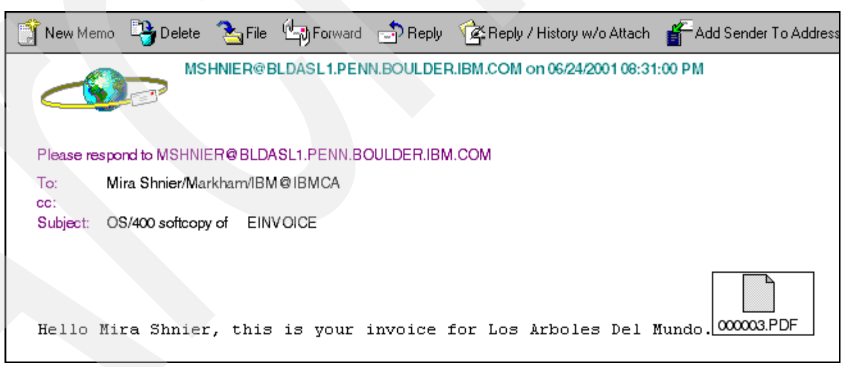


Figure 189. Sample Lotus Notes e-mail with PDF attachment

When you open the attached PDF file with Adobe Acrobat, you see the invoice from Super Sun Seeds as shown in Figure 190.

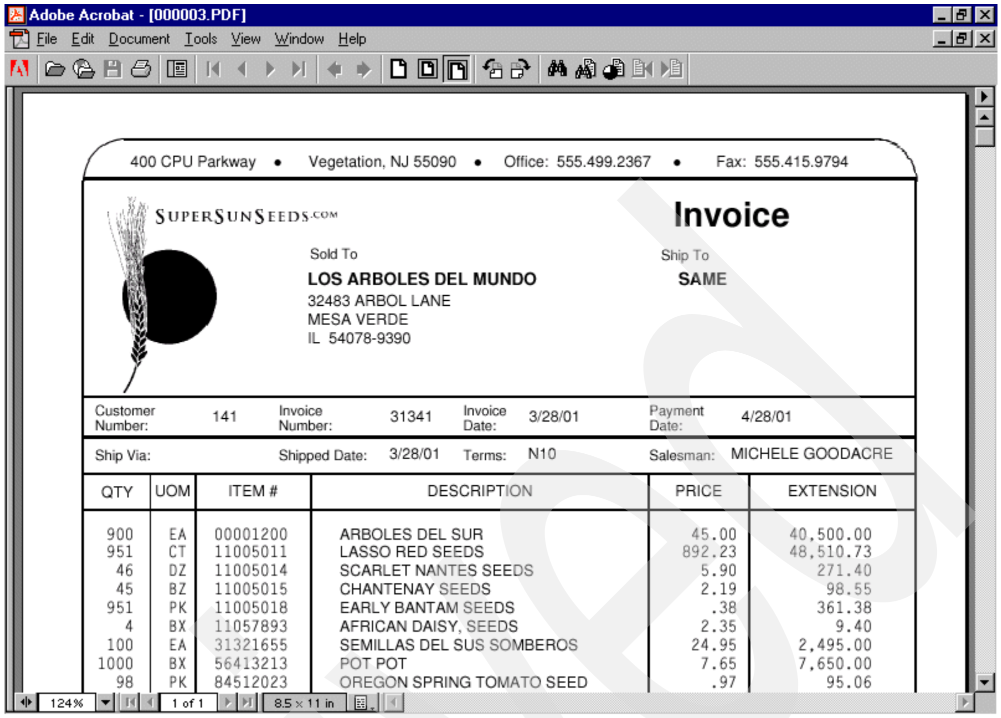


Figure 190. Opening the PDF attachment with Adobe Acrobat

Moving the spooled files to the output queue is a step that is included in the job that processes each spooled file.

14.5.5 Checking for non-delivery messages

The e-mail function of Infoprint Server for iSeries uses the Send Distribution (SNDDST) command to do the mailing. SNDDST in itself was originally designed for distribution using SNA Distribution Services (SNADS). However, its functions are mapped to TCP/IP Simple Mail Transfer Protocol (SMTP) and e-mail addressing.

When an e-mail document is sent, and either the destination user name or the domain name is incorrect, a message is returned to the sender. The SNDDST command does not have a specific vehicle to check for these incoming messages, but they are accessible and should be monitored.

The non-delivery messages are directed to the user who is designated as the sender of the e-mails. The sender name can be specified with the PDFSENDER parameter in the PSF configuration object used for the PDF function. Or it may be set in the USRDFNDTA in the spooled file itself by specifying a value for MAILSENDER. The spooled file setting overrides the PSF configuration object.

One way to set up the sender to receive the non-delivery messages is to set up a Post Office Protocol Version 3 (POP3) client associated with their user ID. This is done by mapping the sender's entry in the iSeries server directory to an SMTP address. An SMTP address is the address you see on the Internet (myname@domain). There are three steps:

1. Enroll the sender in the system directory.
2. Associate the sender’s user ID with an SMTP user ID.
3. Configure a client to receive e-mail for that user.

Be aware that you are dealing with three different types of user identification here. There is the iSeries server user profile that is created with the CRTUSRPRF command. There is the SNADS user ID in the iSeries server directory that is created using ADDIRE. Finally there is the SMTP address used for Internet mail that is created using the WRKNAMSMTP command. To avoid confusion, we recommend you make all three of these names the same.

14.5.5.1 Enrolling the sender in the system directory

The sender’s user ID must be added to the iSeries server directory for the e-mail function to work. To direct the non-delivery messages to a POP3 client for the sender, use the Add Directory Entry (ADDDIRE) or Change Directory Entry (CHGDIRE) command. Then select option `2` (System message store) for Mail service level and option `3` (SMTP name) for Preferred address as shown in the example in Figure 191.

```
                              Change Directory Entry

User ID/Address . . . . :   MSHNIER   BLDASL1

Type changes, press Enter.

  Mail service level  . .   2                1=User index
                                             2=System message store
                                             4=Lotus Domino
                                             9=Other mail service
    For choice 9=Other mail service:
      Field name  . . . .                    F4 for list

  Preferred address . . .   3                1=User ID/Address
                                             2=O/R name
                                             3=SMTP name
                                             9=Other preferred address
    Address type  . . . .                    F4 for list
    For choice 9=Other preferred address:
      Field name  . . . .                    F4 for list
                                                                    More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F18=Display location details
F19=Change name for SMTP           F22=Remove name for SMTP
```

Figure 191. CHGDIRE for sender to receive non-delivery notices

14.5.5.2 Associating the sender’s user ID with an SMTP address

The SMTP address for e-mail must be associated with the user’s SNADS address name that is in the system directory. While you are enrolling the user in the system directory, press F19 from the CHGDIRE panel as shown in Figure 191, which takes you to the Change Name for SMTP panel. An alternative is to enter WRKNAMSMTP on the command line. The panel shown in Figure 192 should appear the same either way.

```
                        Add Name for SMTP
                                                 System:   BLDASL1
User ID/Address . . . . . :   MSHNIER   BLDASL1


SMTP user ID  . . . . . . :   MSHNIER
SMTP domain . . . . . . . :   BLDASL1.PENN.BOULDER.IBM.COM


SMTP route  . . . . . . . :



Press Enter to continue.

F3=Exit   F12=Cancel
```

Figure 192. Associating an SMTP address to a user ID using WRKNAMSMTP

14.5.5.3 Configuring a POP3 client to receive the messages

There are many software products that can be used to act as a client to the iSeries POP3 server. These include clients such as Eudora or Netscape. The configuration steps naturally depend on each product's individual interface. However, the information you must provide is basically the same.

This example uses Netscape Mail. Identify the incoming mail server by selecting **Edit-> Preferences-> Mail & Newsgroups-> Mail Servers**. Then you see the Preferences panel (Figure 193).

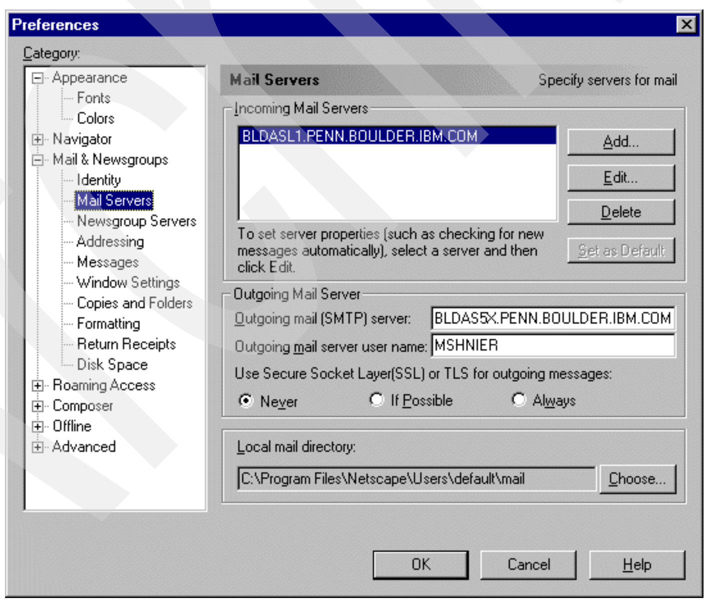


Figure 193. Netscape Mail Server Preferences

Click **Add** to define the incoming mail server to the address you previously defined for the e-mail sender. Figure 194 shows an example of the Mail Server Properties panel that appears.

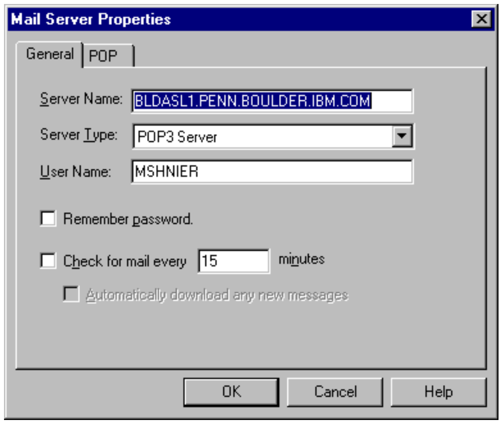


Figure 194. Netscape Mail Server Properties

The user name you enter here should match the iSeries user ID. When the sender retrieves mail, they are prompted for the iSeries password for that user ID.

If you use this client to send electronic mail to other destinations via the iSeries server, you must set up the outgoing mail server information and other identification information.

Figure 195 shows a portion of a returned e-mail that was sent to an incorrect domain. The format is slightly different if the domain name is correct, but the user name is wrong. Not shown here is the fact that the original e-mail is contained in this message further down the page, including the PDF attachment.

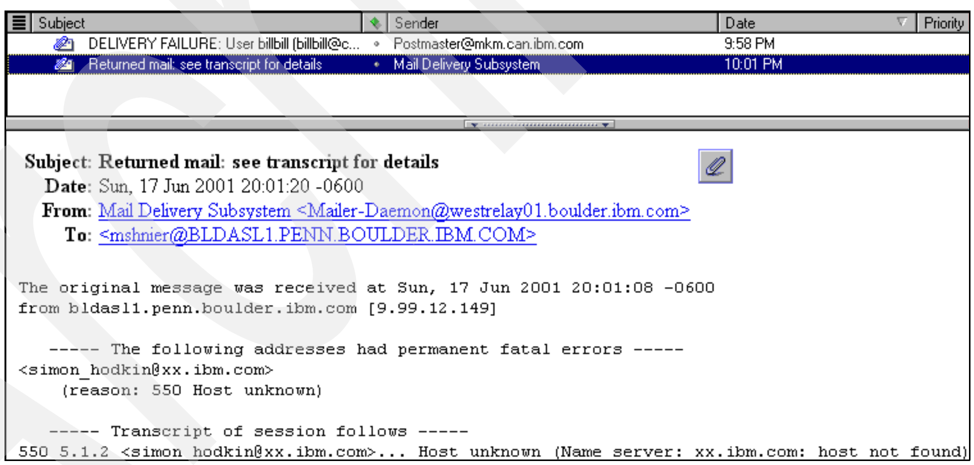


Figure 195. Returned e-mail message in Netscape

You should give some thought as to how you set up and manage the sender user ID for the PDF distribution function. If you set up your PSF Configuration object with the default value of PDFSENDER(*SPLFOWN), you may have to enroll many different users so they can monitor for the non-delivery messages. For business-critical applications, consider setting up a special user ID for the sender function and have one individual responsible for checking for messages and taking appropriate action, such as contacting the intended recipient and re-sending the document.

For more information on using the iSeries server as an e-mail server, go to the iSeries Information Center on the Internet at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

Select **V5R1-> Networking-> TCP/IP-> E-mail**.

The configuration of the sender to send and receive e-mail is done only once during setup.

14.6 Other maintenance tasks

The example provided in this chapter involves only the basic minimum coding required to produce the desired electronic mail. To implement this as a robust business solution, additional tasks must be included in the application. Some of these include:

- Building CL programs to automate the process
- Checking the data queue for the status of each of the PDF files
- Maintenance routines to add, change, and delete records in the LOOKUP file
- Cleanup of any temporary files created in the IFS or standard library structure

14.6.1 Using the PDF data queue

The IPDS to PDF process creates entries in a data queue object if you specify one in the PDFDTAQ parameter in the PSF configuration object. To use this function, the data queue must exist and have a minimum length of 752 characters. Use the iSeries QRCVDTAQ API to read the entries. For more information on QRCVDTAQ, see the iSeries Information Center on the Internet (see the Web address in the previous section).

Using the data queue helps you identify other types of problems that may occur. For example, if you forget the single quotes on the e-mail address in the LOOKUP table, it is treated as an invalid address and the file is not sent. A return value of ‘05’ is logged in the data queue.

The layout of the fields in the data queue are listed in E.2, “PDF data queue structure” on page 346.

14.7 An alternative to e-mail

There are situations for which it is more appropriate to post the PDF files on a Web site for users to retrieve rather than to have the files e-mailed directly. This approach is sometimes referred to as having the end user “Pull” the information rather than you “Push” it to them.

In this case, you may choose to follow most of the steps described in this document. However, rather than e-mailing the final documents, you store them in the system IFS as stream files by specifying PDFGEN(*STMF). A Web server application posts the files as required. In this situation, the information in the data queue is useful. It can be used to match up the index information in the Mail Tag field to the path and name of the PDF file that is created in the IFS.

You may also choose to store the files as stream files if you plan on using a different method to e-mail the PDF files. For example, you may want to integrate the PDF stream files in a Domino-based application.

14.8 One-time setup versus ongoing tasks

This chapter describes a number of steps that are needed to complete the task of converting an iSeries applications from a spooled file to an e-mail containing a PDF attachment. These tasks can be divided into two categories:

- One-time setup tasks
- Tasks to be repeated for each spooled file

To help in planning, the two types are summarized here.

14.8.1 One-time setup

You perform the following tasks once as part of the one-time setup of the e-mail function:

- Create or modify the Infoprint Designer project.
- Modify the printer file or change the CL to use the new form definition, page definition, and overlay.
- Modify the application program (if necessary) to generate data that can be used by the CRTAFPDTA index function.
- Modify the ICODED.FNT file in AFP Viewer if necessary.
- Check the output using the AFP Viewer.
- Enable OS/400 to send e-mail.
- Create the PDF printer device description and PSF Configuration object.
- Create the e-mail exit program.
- Enroll the sender in the iSeries server directory and create an SMTP address.
- Configure a POP3 client to receive the non-delivery messages.
- Create a data queue to capture status messages.

14.8.2 Repeated tasks

It is likely that you will want to build a CL program to manage the steps that will be repeated for each spooled file you are mailing using this process. The steps to incorporate in the program include:

- Creating a physical file to receive the output from the CRTAFPDTA command.
- Setting the default target directory for the CRTAFPDTA command using the CHGCURDIR command.
- Running the CRTAFPDTA command.
- Changing the OUTPUTMRG member name to a more meaningful name.
- Re-spooling the output using the PRTAFPDTA command.
- Moving the output to the PDF output queue.
- Monitoring for files not sent.
- Cleaning up any temporary files created during this process.

Chapter 15. Print support in J.D. Edwards' OneWorld Xe

For more than 22 years, J.D. Edwards and Company has provided innovative and flexible business solutions essential to running complex and fast-moving multi-national organizations. It has grown into the leading provider of agile and collaborative solutions for the Internet economy.

J.D. Edwards' OneWorld Xe is a network-centric solution that separates business rules from the underlying technology. As new technologies evolve, customers can add them to the framework without disrupting their ongoing business.

In October of 2000, J.D. Edwards announced a significant new release of their OneWorld Xe product. One of the enhancements was the addition of support to output AFPDS from the OneWorld report generator. Previous to that, only PCL, PostScript, or unformatted line printer output in EBCDIC or ASCII was generated.

Most available documentation describes printing in PCL or PostScript. This chapter focuses on configuration tasks and usage considerations that are specific to AFPDS and EBCIDC Line Printers. In addition, this chapter describes how these two types of output can be used and enhanced for enterprise printing.

15.1 General print setup

When OneWorld Xe is used to generate a report, it first creates the output in PDF format in a physical file. It then uses the various transformations as selected in the Printer Setup to convert the output to PCL, PostScript, line printer data (EBCDIC or ASCII), or AFPDS.

You can find detailed information, including sample windows, on configuring printing in J.D.Edwards OneWorld in the redbook *J.D. Edwards OneWorld Implementation for AS/400,* SG24-5195. The process is summarized in the following steps:

1. From the main OneWorld display, select **GH9013** to bring up the Printers menu.
2. Double-click the **Work With Physical Printers** option.
3. Click **Add Printer**.
4. A welcome screen appears. Click **Next**.
5. Enter the Platform Type (AS/400) and the corresponding output queue and library name to associate with the J.D. Edwards printer definition. Click **Next**.

> **Note**
>
> The AS/400 or iSeries output queue, and its corresponding printer device description, must be created separately as an iSeries task. This is not done by the J.D. Edwards software.

6. Enter the Printer Model and a description of the printer location. Select the default paper type. Click the **Details** tab.
7. You are presented with the Printers [Printer Setup] Details panel. At this point, you have the opportunity to select the data stream or Printer Definition

Language that you want to generate. This is described in greater detail in 15.2, “AFPDS printing from OneWorld Xe” on page 304, and 15.3, “Line printing from OneWorld Xe” on page 307.

8. After you complete the printer setup, you return to the panel that was displayed in step 3. You can click **Define Default Printer** to set it as the default for the appropriate user or group.
9. On the Work with Default Printers panel, click the **Add** button.
10. Enter the appropriate User ID, Printer Name (library/output queue), Host Name, and Object Status. Click the **OK** button.

15.2 AFPDS printing from OneWorld Xe

The latest release of OneWorld provides support for generating AFPDS. This new support gives iSeries users of OneWorld the ability to print to a wide range of IPDS printers. This offers the fidelity and recoverability not usually offered by PCL or PostScript print solutions. See 1.3.2, “Print Services Facility/400” on page 21, for more information on the advantages of printing AFPDS spooled files using PSF/400.

15.2.1 Configuration

Follow the normal OneWorld menus as described in 15.1, “General print setup” on page 303, until you reach the Printers [Printer Setup] panel. Click the **Details** tab and then you see a panel like the example shown in Figure 196.

For AFPDS printing, select **Custom** under Printer Definition Language, and click the corresponding radio button to make it the Default.

At this point, you must complete the customization. Click **Advanced** from the **Form** menu on the left of the display.

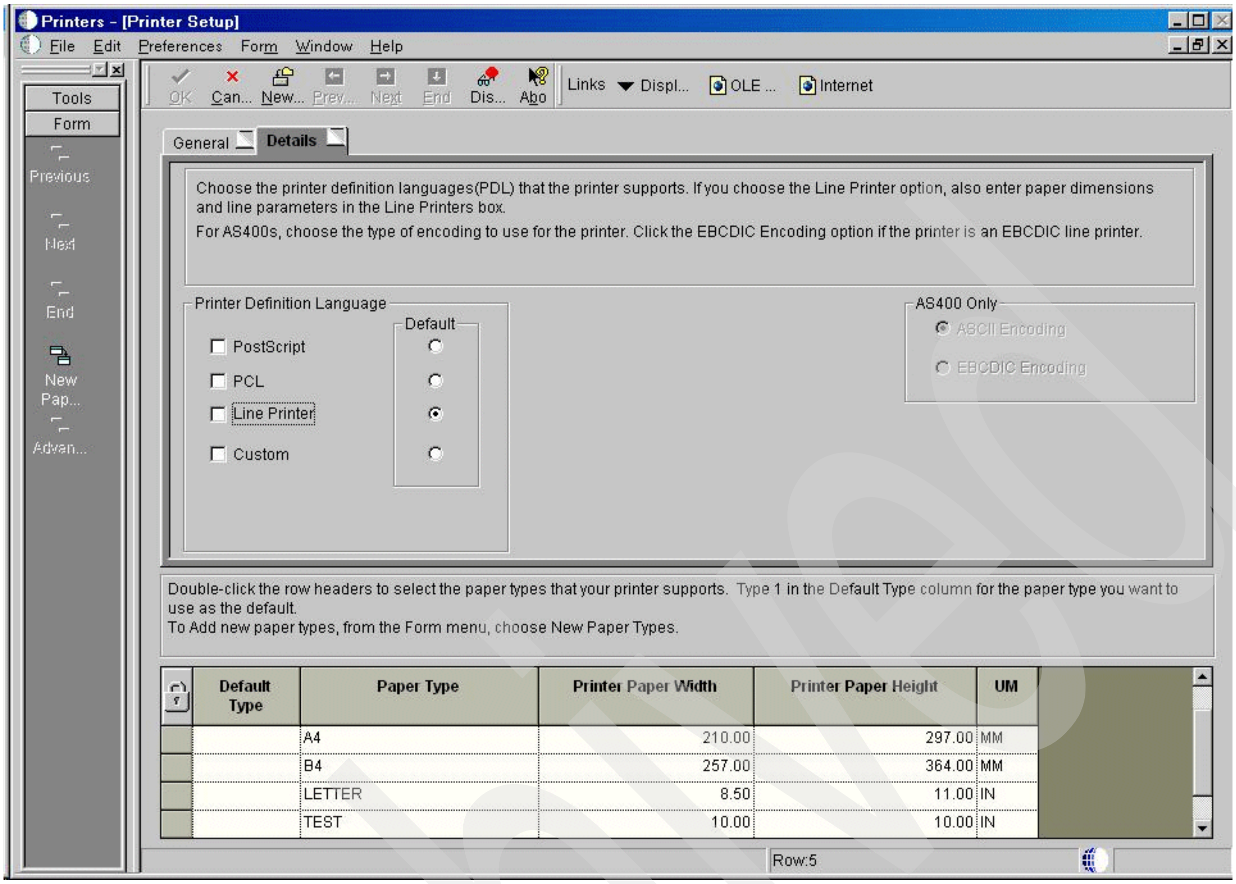


Figure 196. Printers [Printer Setup]

The Printers [Work with Conversion Programs] panel appears as shown in Figure 197. Highlight the ***JDE AFP** conversion program from this panel and click **Select**. If the AFP conversion program is not shown (which may be the case if you upgraded to OneWorld Xe from an earlier release), you must add it.

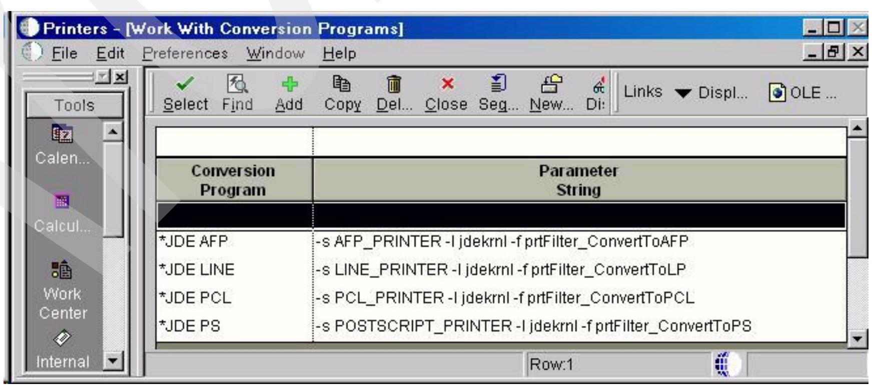


Figure 197. Printers [Work With Conversion Programs]

Click the **Add** icon to display the Printers [Advanced Conversion Program] panel as shown in Figure 198.

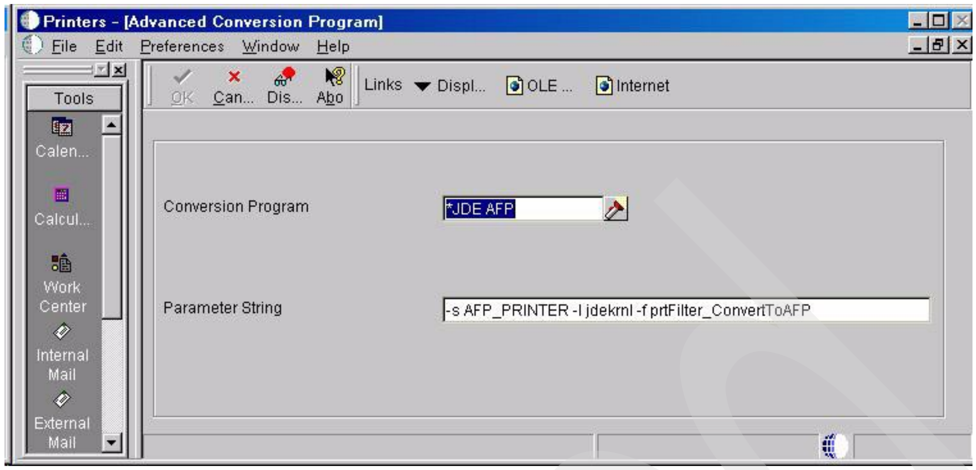


Figure 198. Printers [Advanced Conversion Programs}

Enter the following information in the fields shown in Figure 198:

- Conversion Program: `*JDE AFP`
- Parameter String: `-s AFP_Printer -l jkekrnl -f prtFilter_ConvertToAFP`

For other printer configuration tasks, see *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195, and other documentation available from J.D. Edwards. Remember that you associate the OneWorld printer definition with an iSeries output queue. This output queue, in turn, is associated with a printer device description that is configured with TYPE(*IPDS) and AFP(*YES). It is the user's responsibility to create the iSeries device description. This is not done automatically by J.D. Edwards software.

15.2.2 Considerations

Almost all functions that are supported through the J.D. Edwards print interface can be printed on an IPDS printer that is configured with AFP(*YES). There are some important restrictions of which users must be aware:

- The text output is generated using host outline font resources. These are supported on most of the current IBM laser printer product line, such as the Infoprint 20, 21, 32, 40, 60, 62, 2000, 3000, and 4000.

  If you are printing to Network Printer 12, 17, or 24, Print Services Facility converts the request for a host font resource to the corresponding printer resident outline font. Fractional point sizes are not supported on these printers, and rounding may impact column alignment.

  Older laser printers and impact printers do not support outline fonts, and they do not work in this environment.

- Although J.D. Edwards supports color printing, this support is not transferred to the AFP support.
- Barcodes are not supported.
- There is no support for DBCS.
- There is no support for overlays because there is no place to specify the overlay in the J.D. Edwards panels.

- Paper trays cannot be configured with the J.D. Edwards printer setup. Output defaults to the primary bin of your printer.
- For those of you who are familiar with the concept of form definition, understand that an inline form definition is generated as part of the spooled file. This prevents any external form definition from acting on the spooled file.

Note

Print Services Facility/400 (PSF/400) is required to print AFPDS spooled files to a printer configured with TYPE(*IPDS) and AFP(*YES).

AFPDS output is also supported on other OneWorld Xe platforms, such as AIX or Windows NT. The appropriate Infoprint Manager software is required on those platforms to print AFPDS spooled files.

15.2.3 When to use J.D. Edwards support for AFPDS printing

AFPDS provides the type of mission critical print that many businesses require. This was previously unavailable in earlier versions of the J.D. Edwards solutions. If page level recovery and print fidelity are important to your business, and the formatting provided from within OneWorld Xe is sufficient for your needs, then use the AFPDS support described in this section. This provides a robust printing solution with the least amount of effort.

15.3 Line printing from OneWorld Xe

OneWorld Xe supports the generation of unformatted line data in EBCDIC or ASCII format. If EBCDIC is selected, the data is placed in a spooled file that is designated with DEVTYPE(*SCS). ASCII spooled files are generated with DEVTYPE(*USERASCII).

This type of output can be sent to traditional iSeries server impact line printers (such as 4230 or 6400 in SCS mode) or to other printers in ASCII mode.

SCS spooled files can be reformatted and enhanced using the AFP tools described in 1.1, “Formatting your application data” on page 1, and 1.2, “Building or obtaining external resources” on page 14. IBM offers a wide variety of iSeries printing software solutions for the creation, formatting, and management of your electronic documents described in Chapter 1, “iSeries printing software to create, manage output” on page 1.

15.3.1 Designing a OneWorld report to run on a line printer

When you run a report for printing on a line printer, or a report to be passed to other tools for future processing, you must follow certain guidelines to ensure that the information is formatted on the page correctly. If a report is to be used for line printing and printing to other types of output, you should create separate versions of the report for each destination type.

In the Report Design Tool, you must make the following changes:

1. Set the vertical grid spacing in the Alignment Grid form to `16`.
2. Select a fixed pitch font, such as Courier New.

3. Adjust the width of some fields on the report as necessary. Repeat this action for any Group sections in the report.
4. Align the report to the Top Edge.

For more information on these steps, refer to the following J.D. Edwards OneWorld documents:

- *Enterprise Report Writing Guide*
- “Printing OneWorld Reports” in the *System Administration Guide*

15.3.2 Configuring a OneWorld Xe for line printing

The initial configuration step for line printing is the same as the steps for AFPDS until you reach the Printers - [Printer Setup] panel as shown in Figure 196 on page 305. Select **Line Printer** under Printer Definition Language. You are prompted to select either ASCII or EBCDIC encoding (this option is for iSeries servers only; other J.D. Edwards platforms support only ASCII encoding).

You are then asked to select the values for characters per inch, lines per inch, characters per page, and lines per page that are used by this printer, as shown in Figure 199. These values determine the paper dimensions that your line printer uses when printing OneWorld reports. If the same physical printer is used to print different applications using different values for these settings, you must set multiple J.D. Edwards printer definitions. These, in turn, point to different iSeries output queues, which could point to the same AS/400 or iSeries printer device.

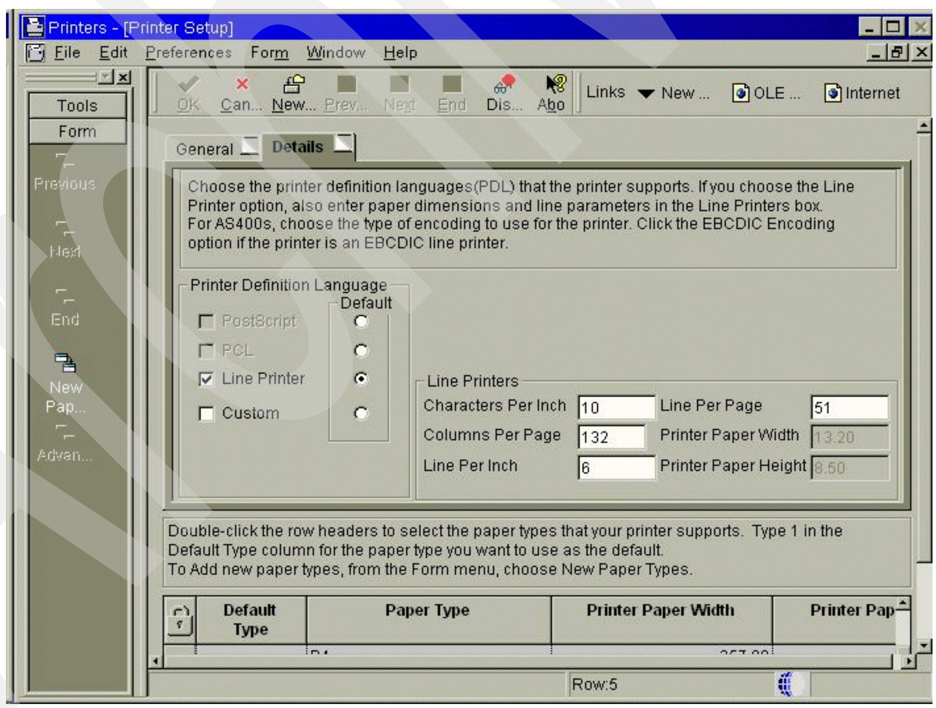


Figure 199. Setting up a line printer

15.3.3 Enhancing your OneWorld Xe output

Once you generate the SCS spooled files on your iSeries server, you can send them to a line printer, such as a 6400 impact printer. At this point, it is also possible to use some of the solutions described in 1.1, “Formatting your application data” on page 1, to convert the SCS spooled files to AFP and print them on a laser printer. This allows you to take advantage of the wide variety of formatting options.

This provides two advantages. The first is the ability to output to an IPDS printer and have the recoverability associated with Print Services Facility/400 printing. The second is that the limitations described 15.2.2, “Considerations” on page 306, no longer apply. You can use any type of font that your printer supports. You can convert fields to barcodes, add overlays, and add graphics. Fields can be moved around or suppressed. You can print multiple copies of selected pages, and you can control the paper source for each copy.

The easiest route is probably APU, since the spooled files coming out of J.D. Edwards are in SCS format. See the description of APU in 1.1.4, “AFP Printsuite – Advanced Print Utility (APU)” on page 9. In addition, it is a fairly easy task to set up the APU monitor to match up the spooled file with the required APU print definition. A separate tool (such as AFP Utilities/400) is required to create other fixed resources, such as overlays or page segments.

Infoprint Designer for iSeries or PPFA can be used to create form definitions and page definitions to enhance the way the output looks. However, those objects require that the spooled file be in LINE format. This means that you must write or acquire a monitor program to catch any new spooled files as they arrive in the output queue, copy them to a physical file, and then copy them back to a new printer file. See Appendix D, “Output queue monitor” on page 337, for tips on writing such a monitor program. Infoprint Designer includes components for the creation of overlays and page segments. If you use PPFA, you require a separate tool to create those resources.

15.3.4 Differentiating the output

Before you apply APU, PPFA, or Infoprint Designer formatting objects to the SCS output from OneWorld Xe, you must identify which resources are going to be used with which spooled file. In most cases, the spooled file name corresponds to the application that created it. You can use this information in the APU monitor or if you use a custom monitor to work with Infoprint Designer for iSeries objects or PPFA (Figure 200).

```
                         Work with Output Queue

Queue:   DEVDES3A       Library:   QGPL           Status:   HLD/WTR

Type options, press Enter.
 1=Send  2=Change 3=Hold 4=Delete  5=Display  6=Release  7=Messages
 8=Attributes        9=Work with printing status

Opt  File        User   User Data   Sts   Pages   Copies  Form Type
     R0008P      JDEOW              RDY      1*      1   *STD
     R0513P      JDEOW              RDY      1*      1   *STD
     R47071      JDEOW              RDY      1*      1   *STD

                                                               Bottom
Parameters for options 1, 2, 3 or command
===>
F3=Exit   F11=View 2   F12=Cancel   F20=Writers   F22=Printers
```

Figure 200. Sample output queue showing different spooled file names

15.3.5 When to use line printer output from OneWorld Xe

This is the obvious choice when printing to impact printers, either traditional EBCDIC (or SCS) printers attached directly to the iSeries server or ASCII printers attached remotely.

This method can also be used if you want recoverability available when printing to an IPDS printer with PSF/400 and you want to format the output beyond what is available today using the new AFPDS print support provided with OneWorld Xe. Features that may lead to this decision include adding an electronic form, barcodes, and complex reformatting of the data.

Chapter 16. Using Content Manager OnDemand for iSeries

This chapter provides an introduction on Content Manager OnDemand for iSeries. Content Manager OnDemand provides high compatibility with the AFP Architecture and iSeries print environment.

16.1 Introduction to Content Manager OnDemand

IBM Content Manager OnDemand for iSeries is an application solution to store large volumes of data and retrieve selective data, whether on disk, optical, or tape storage media. It provides computer output to laser disk (COLD) and extended archiving functions for the iSeries server. Content Manager OnDemand key features are:

- Provides a functionally rich, cost-effective application solution to store and retrieve large volumes of data
- Provides users with a powerful search and browse capability to locate specific information
- Extracts index values, and compresses and stores data on disk, optical, or tape storage media
- Manages the life cycle requirements of data and automatically moves data to the appropriate media
- Helps reduce costs by eliminating the need to print to paper or microfiche, improving the usability of report information, increasing productivity, and enhancing customer service capabilities
- Offers a powerful graphical administration system for data definition, management, and security

16.1.1 Document indexing

The Content Manager OnDemand report capture process automatically extracts index information, such as order number, customer number, customer name, and date from electronic reports and documents during the capture process and then builds a relational table.

These index values are used to segment the reports into logical information units, such as individual orders or invoices. Each of these report segments is individually indexed to allow direct access to the data required without retrieving the entire report file. OnDemand also provides an easy to use graphical indexing interface that allows an administrator to define the indexes for a report by simply highlighting the values on a sample page.

16.1.2 Document viewing

Content Manager OnDemand supports Windows NT, Windows 98, Windows 2000, Windows ME, and optional Internet browser interfaces for document retrieval and viewing. Multiple documents from different applications can be displayed simultaneously on the screen. These documents can be of multiple formats including AFP, line data, PDF, and standard image formats. An electronic forms capability is supported through AFP overlays. This allows a single electronic form to be used for viewing, printing, and faxing. Also, unlike other systems, there is no need to create a special image or vector form just for viewing.

16.1.3 Document storage

Content Manager OnDemand provides hierarchical storage management across magnetic, optical, and tape devices. Each application defined to the server can specify the length of time report data will be stored on magnetic disk and the life of the data. For example, a customer order application may require three months of data to be stored on magnetic disk and have a life of seven years.

16.2 Report type

This section describes the various data types that Content Manager OnDemand supports for archiving.

16.2.1 SCS spooled files

SCS spooled files are the most common on the iSeries server. Frequently referred to as line data, these spooled files include transaction-type reports as well as document-type reports that do not require special fonts or overlays.

16.2.2 SCS-extended spooled files

SCS-extended spooled files contain variable fonts or other extended spooled file attributes, such as variable lines per inch, variable characters per inch, variable fonts, etc. Content Manager OnDemand keeps these attributes for displaying, printing, or faxing.

16.2.3 AFP spooled files

AFP spooled files often contain logos, special fonts, boxes, and shading. These advanced print characteristics are archived along with the text data in Content Manager OnDemand, and used for viewing, printing, faxing, and e-mailing retrieved documents.

To handle fully-resolved AFP data, the archiving process stores the AFP resources with the report. Resources that are completely identical are stored once only. Content Manager OnDemand checks for changed resources each time a new report is archived.

16.2.4 PDF

To extract index data from Adobe PDF documents, the Content Manager OnDemand PDF Indexer is used. Indexing parameters include information that allow the PDF indexer to identify the key fields in the print data stream, and create index elements. Content Manager OnDemand uses the index data for efficient, structured search and retrieval.

16.3 Case study

Content Manager OnDemand complements the Super Sun Seeds invoicing application well. It provides a comprehensive system to manage documents after they have been created. A spooled file containing thousands of invoices can be defined to Content Manager OnDemand, automatically indexed, compressed, and archived to disk, optical, or tape media. Super Sun Seeds invoices can then be easily retrieved and viewed, down to the individual invoice. The invoice can also be reprinted (either on an iSeries server printer or on a workstation printer), faxed, or sent via e-mail. This saves the Super Sun Seeds Company money by not needing to print multiple copies of invoices for their records, not needing to send the data to microfiche, and not needing to keep large spooled files on their iSeries server. It more significantly improves productivity and customer responsiveness by providing the capability for customer service representatives, for example, to view, print, fax, and e-mail invoices right from their desks while speaking with customers.

The invoices are defined to Content Manager OnDemand using a graphical administration tool. Using the graphical tool, you begin by identifying the name by which the archived data will be known to the end-users and the data type (AFP, for example). You also define the date and index fields your end-users will use for searching. You do this by using a full-function point-and-click definition panel that allows you to work with actual spooled file data from your iSeries server. Even if your desired index fields are not always on the same line or in the same column on the printed page, Content Manager OnDemand can still find them!

Many other data characteristics are defined, such as how Content Manager OnDemand should break up the entire spooled file into separate, multipage invoices. Content Manager OnDemand doesn't care how many pages belong to each invoice – that number can vary – as long as you tell Content Manager OnDemand how to determine when one invoice ends and the next one begins within the spooled file. OnDemand defines this “logical break” so that users can retrieve a specific invoice rather than have to scroll through the entire spooled file to find a particular one.

A typical example of this concept is to define a “break” every time invoice number changes within the spooled file. When a change is detected, OnDemand knows that the end of the previous invoice has occurred, and that the start of a new invoice has begun. When a user retrieves an invoice, OnDemand knows what pages belong to what invoice.

The invoices are defined to Content Manager OnDemand one time. Then each time the spooled file of invoices is created (weekly or monthly, for example), the spooled file can be automatically captured and stored within Content Manager OnDemand. In addition, there are a number of ways to automate the process to avoid any operator intervention, using the Content Manager OnDemand output monitor or by placing Content Manager OnDemand CL commands in batch programs.

Once the invoices are stored, the real value of OnDemand becomes obvious, using the capability to quickly and easily find any invoice (or set of related invoices) ever stored. Once found, an invoice can easily be viewed, printed, faxed, or sent via e-mail, as illustrated in the following section.

16.3.1 Retrieving archived information

As a user searching for an invoice, the first step is to tell Content Manager OnDemand what you are looking for. In the case study, we choose the name INVOICES for our archived invoices (a logical choice), so you select INVOICES from the list of folders presented, as shown in Figure 201. Note that the folder list shows only the archived data to which you are authorized.

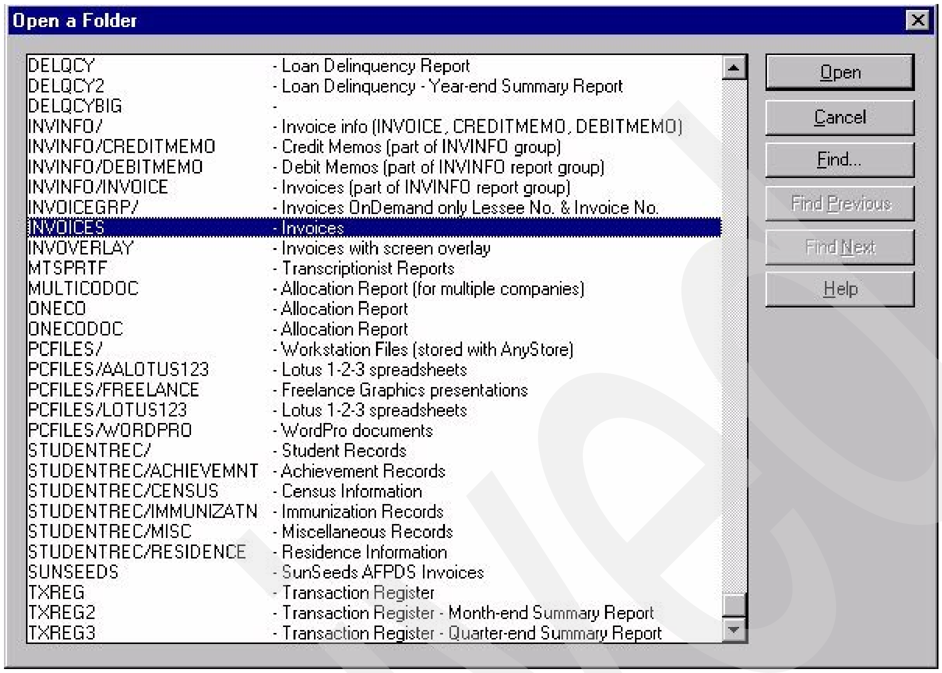


Figure 201. OnDemand folder

To select the invoices, double-click **INVOICES**, or you can simply select **INVOICES** and click **Open**. On the resulting panel, you enter your search criteria. You may know the actual number, or you may need to review all the invoices for a particular customer number or customer name for the last four months, for example (by expanding the date range). You may select alternate search operators such as Between, Like, Greater Than, and others instead of Equal To if desired. You may also enter customer name IMP% to save keystrokes and let Content Manager OnDemand find any invoices for customers with names beginning with IMP (such as IMPROVED PRINTING COMPANY) as shown in the example.

When you enter your search criteria and choose your search operator, select **Search**. Content Manager OnDemand fills the documents list with the documents that satisfy your search criteria and updates the status bar with a count of the documents. From this list, you can either choose a single document to view or select multiple documents to view simultaneously.

Figure 202 shows the search capabilities of Content Manager OnDemand.

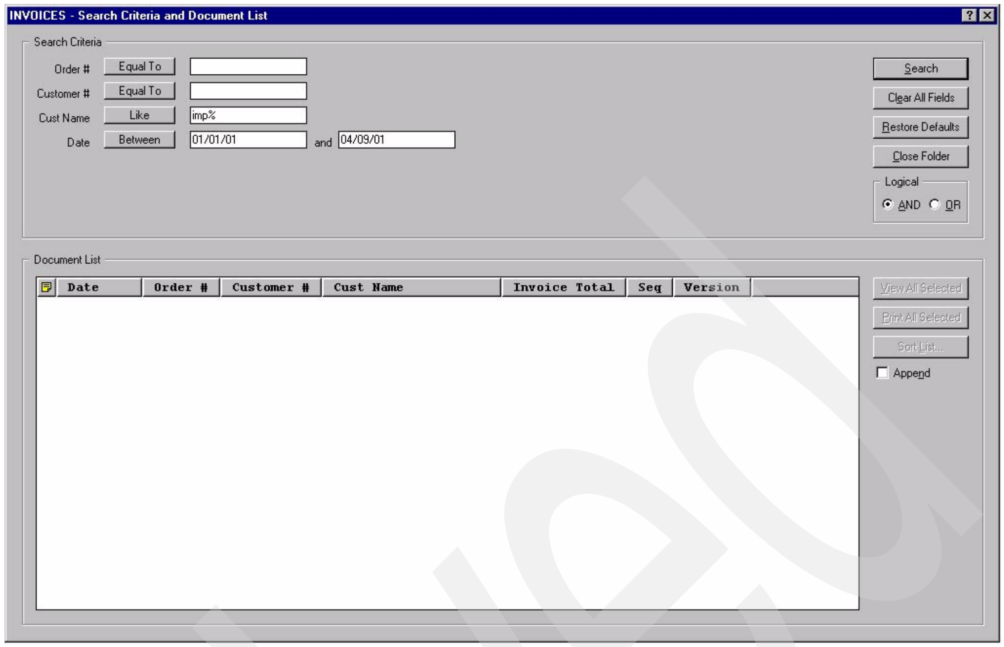


Figure 202. Search Criteria window

OnDemand uses standard search operators to search for documents. Choose a search operator button (to the right of the search field name), and OnDemand displays the Select Operator dialog box. The list of operators valid for that field appears in the dialog box. Choose an operator button or type the keyboard shortcut for an operator name.

The results of clicking Search after you type the search criteria are shown in Figure 203.

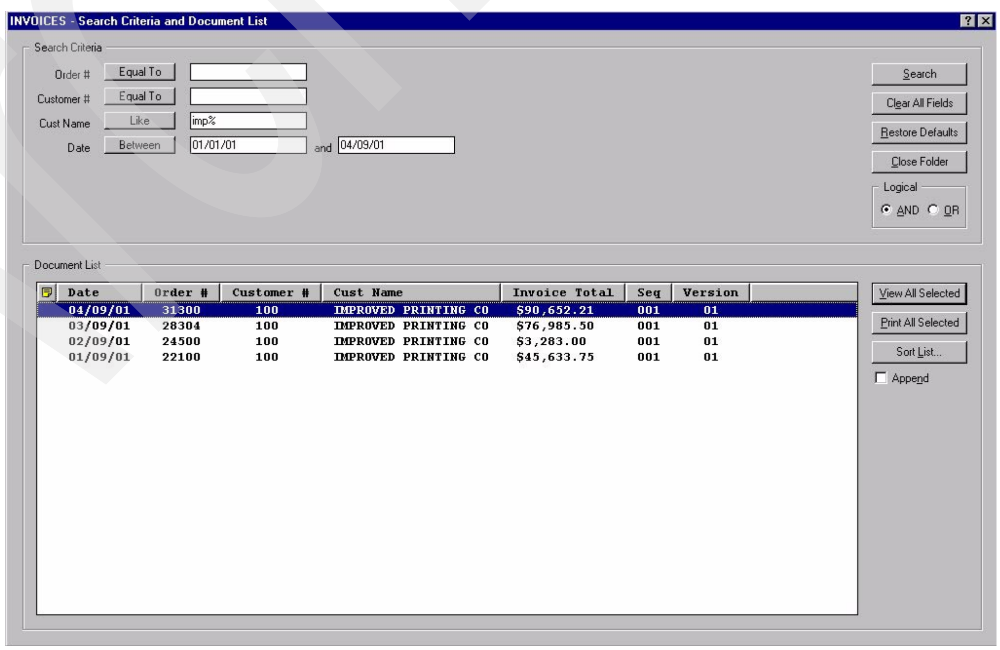
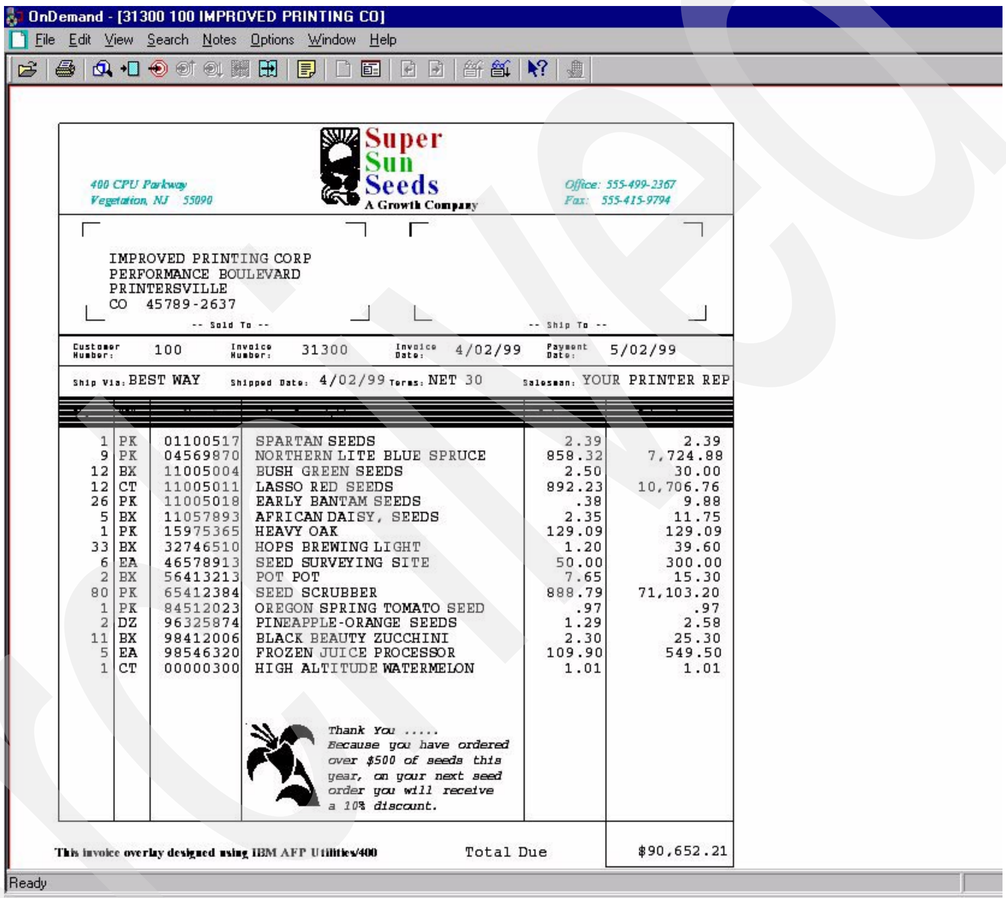


Figure 203. Search criteria result

The real power of Content Manager OnDemand becomes evident when you consider the value of being able to search for *all* invoices for this particular customer for the past five years. Over time as your data archives grow, more and more historical data is available at your fingertips.

If you want to view a particular invoice, double-click an item on the list or select it and then click **Search**. Content Manager OnDemand displays a fully-resolved view of the invoice, including all of its AFP components.

Figure 204 shows a good example how an AFP document may be displayed on the viewer.


OnDemand - [31300 100 IMPROVED PRINTING CO]

File Edit View Search Notes Options Window Help

Super Sun Seeds
A Growth Company

400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

IMPROVED PRINTING CORP
PERFORMANCE BOULEVARD
PRINTERSVILLE
CO 45789-2637

-- Sold To --

-- Ship To --

Customer Number: 100 Invoice Number: 31300 Invoice Date: 4/02/99 Payment Date: 5/02/99

Ship Via: BEST WAY Shipped Date: 4/02/99 Terms: NET 30 Salesman: YOUR PRINTER REP

1	PK	01100517	SPARTAN SEEDS	2.39	2.39
9	PK	04569870	NORTHERN LITE BLUE SPRUCE	858.32	7,724.88
12	BX	11005004	BUSH GREEN SEEDS	2.50	30.00
12	CT	11005011	LASSO RED SEEDS	892.23	10,706.76
26	PK	11005018	EARLY BANTAM SEEDS	.38	9.88
5	BX	11057893	AFRICAN DAISY, SEEDS	2.35	11.75
1	PK	15975365	HEAVY OAK	129.09	129.09
33	BX	32746510	HOPS BREWING LIGHT	1.20	39.60
6	EA	46578913	SEED SURVEYING SITE	50.00	300.00
2	BX	56413213	POT POT	7.65	15.30
80	PK	65412384	SEED SCRUBBER	888.79	71,103.20
1	PK	84512023	OREGON SPRING TOMATO SEED	.97	.97
2	DZ	96325874	PINEAPPLE-ORANGE SEEDS	1.29	2.58
11	BX	98412006	BLACK BEAUTY ZUCCHINI	2.30	25.30
5	EA	98546320	FROZEN JUICE PROCESSOR	109.90	549.50
1	CT	00000300	HIGH ALTITUDE WATERMELON	1.01	1.01

Thank You .....
Because you have ordered over $500 of seeds this year, on your next seed order you will receive a 10% discount.

This invoice overlay designed using IBM AFP Utilities/400 Total Due $90,652.21

Ready

Figure 204. Viewer

This invoice, in its fully-resolved form, can be reprinted, faxed, or e-mailed (if available) from the workstation.

Appendix A. Configuring LAN-attached printers

This subject was covered extensively in *AS/400 Printing V*, SG24-2160. This appendix summarizes the methods, and compares and contrasts them. Ensure you are using the second edition (the -01 suffix) of *AS/400 Printing V*, because several changes have been added to this version. Only the configuration method for the ASCII Simple Network Management Protocol (SNMP) driver is given in full, because the recommendations have changed slightly since the last redbook was published.

This appendix covers:

- How to identify which method to use
- IPDS LAN-attached printers
- ASCII LAN-attached printers:
  - LPR/LPD
  - PJL
  - SNMP
- Other printing methods:
  - IPP
  - PC5250
- Summary of LAN attachment methods by printer

For the latest information on configuring LAN-attached printers on the iSeries, consult the Print section of the SupportLine Knowledge Base at:

`http://www-912.ibm.com/`

Under Technical Databases, click **Software Knowledge Base** and on the next page, choose the **Print** topics in the list.

A.1 Identifying the method to use

First you must determine your printer type. We can divide LAN-attached printers into two groups:

- IPDS LAN-attached printers
- ASCII LAN-attached printers

A.1.1 IPDS or ASCII

If your printer contains an IPDS feature, such as a SIMM or feature card, it will be be configured on the iSeries as a printer of type *IPDS. For LAN-attached IPDS printers, the Advanced Function Printing parameter will always be AFP=*YES. The printer may be an impact printer or a non-impact printer (laser or thermal).

If your printer is capable of printing data streams such as HP-PCL, PostScript, or PPDS, it is an ASCII printer. Impact printers usually use IBM PPDS or a variety of Epson data streams. These printers are normally configured on the iSeries as an emulated printer of type 3812 (non-impact) or 4234, 4214, 5225, etc. However this is not a guarantee that the printer is an ASCII printer. For example, twinaxial SCS printers, virtual printers, and printer emulation sessions use a similar convention.

Note that printers may frequently be capable of performing both roles (for example, IPDS and ASCII support). It is perfectly possible to create multiple LAN configurations for the same printer on the same system.

A.2 IPDS LAN-attached printers

There is *only* one method to use, which is the *IPDS device description*. This is covered in *AS/400 Printing V*, SG24-2160. The principal points to note are:

- IPDS port number is 5001 for IBM printers (may rarely be 9100 for some IPDS printers).
- A PSF configuration object is *optional*, unless you are sharing the printer (with another host, or another configuration on the same host) in which case it is *required.*
- Host Print Transform is not used.

A.3 ASCII LAN-attached printers

You can choose a method, in order of increasing functionality:

- Remote output queue
- PJL device description
- SNMP device description

These are covered in *AS/400 Printing V*, SG24-2160. The principal points to note are:

- All three methods use Host Print Transform (which converts the native iSeries EBCDIC data stream into a suitable ASCII data stream). SCS and AFPDS conversion to ASCII is supported but IPDS conversion to ASCII is not supported.
- The remote output queue does not use a device description, just an output queue.
- TCP/IP port numbers for PJL and SNMP are usually 2501 (IBM laser) or 9100 (most OEM printers and IBM 6400).
- PJL is for PJL-compatible printers only (no impact printer support).
- PJL support may be provided by a LAN print server (adapter), where the printer is attached to a bi-directional parallel port on the print server and the printer also supports PJL commands.
- SNMP support may also be supported by a LAN print server, where the printer is attached to a parallel port on the print server.

A.3.1 Remote output queue (LPR/LPD)

On the iSeries, the common TCP/IP function, line printer requestor (LPR) is invoked by the SNDTCPSPLF command. This is used to send a print file to a server or device running the corresponding receiving program, line printer daemon (LPD). This command can be run from the command line or built into a CL program. The LPR command is also a synonym for SNDTCPSPLF. A remote output queue is a way of automating the LPR command.

The characteristics of the remote output queue include:

- There is no error checking or recoverability.
- The entire spooled file is sent.
- Page range selection is not honored.
- There is no device description.
- The writer is started using the STRRMTWTR command. The writer can be started automatically by a spooled file arriving on the queue, if required.

A remote output queue may be used to send spooled files to other print servers, such as Infoprint Manager for AIX or Windows NT/2000, with or without conversion (for example, Host Print Transform). You can include the LPR/SNDTCPSPLF command in a CL program to customize how the file is to be sent. It is possible to set Destination Options (DESTOPT) that are specific to the target server. For an example of using SNTCPSPLF in this manner, see Appendix H, “AS/400 to AIX printing” in *IBM AS/400 Printing V*, SG24-2160.

Remote output queues may also be used over an SNA network. The Send Network Spooled File (SNDNETSPLF) command is used.

The configuration of remote output queues is described in a number of publications, including:

- *AS/400 Printer Device Programming*, SC41-5713
- *AS/400 Printing IV*, GG24-4389
- *AS/400 Printing V*, SG24-2160

At a minimum, you need the IP address of the remote system or server (the printer itself often performs this role) and the queue name that already exists on that remote system. If the remote system is a printer, then most printer LPDs have built-in internal print queue names that *must* be used. A sample table of common print queue names is listed in *AS/400 Printing V*, SG24-2160. Or you can find a more extensive and up-to-date list on the SupportLine Knowledge Base Web site as explained at the beginning of this chapter.

A.3.2 PJL driver

Most recent ASCII laser printers support Printer Job Language. Since Version 3.0 Release 7.0, OS/400 has included PJL drivers to use when connecting to those printers. This allows you to create a device description for the device as opposed to just a remote output queue. More enhanced control is offered to the user with this attachment method. There is better error recovery and improved operator messages. One key advantage is the ability to select page ranges of a spooled file, instead of printing it in its entirety.

You can use the PJL driver any time you have a TCP/IP LAN-attached PCL or PostScript printer that supports PJL. This includes the following printers from IBM:

- Network Printers 12, 17, and 24
- Infoprint 20, 32, and 40

Note that impact printers do *not* support PJL.

For detailed information on the PJL driver, including the step-by-step configuration method, see “Configuring LAN-attached ASCII printers using PJL drivers” in Chapter 11 of *IBM AS/400 Printing V*, SG24-2160.

A.3.3 SNMP driver

The SNMP ASCII printer driver is a relatively new printer driver for TCP/IP LAN-attached ASCII printers. Provided as standard with OS/400 V4R5 (and available with PTFs at V4R3 and V4R4), this printer driver provides a similar level of function to be found in the PJL printer driver. However this method also offers some support for impact printers. At the present time, the only supported IBM impact printer for use with this method is the IBM 6400 with internal Ethernet card. This printer driver is also the recommended configuration method for ASCII models of the IBM Infoprint 21, not the PJL method.

To determine whether this method can be used with a particular non-IBM printer, the following requirements may be useful. If the printer attaches via a separate LAN adapter, these requirements apply to both the printer and adapter:

- The printer must support the industry standard Host Resource Management Information Base (MIB), RFC 1514. It is also recommended that the printer support the Printer MIB, RFC 1759.
- If an external LAN adapter is used, the printer should connect to the first parallel port and be the only SNMP device on that adapter.
- The printer must have a minimum of read-only access to the “public” SNMP community name. This is usually a default setting.

A.3.3.1 Printer hardware settings

The LAN-attached printer will use either an internal or external LAN adapter to connect to the LAN. These may be referred to by several names, for example:

- Network interface card (NIC)
- Modular input/output (MIO)
- Print server

Branded examples of these include the IBM Network Print Server feature, HP JetDirect print server, and Lexmark MarkVision range.

With any print server, we recommend that you to set the hardware timeout appropriately, so that the device can service other hosts, interfaces, etc., should the current print job fail for any reason (for example, timeout over a slow connection). This hardware timeout may be accessed from the printer control panel, sometimes from a manufacturer’s printer management utility, and usually via a direct method such as Telnet. Unfortunately there are at least two types of timeouts:

- Those concerning the host connection (usually found in the print server configuration)
- Those concerning the internal print process within the print engine itself.

You need to be sure you are changing the correct one!

Print server timeouts

This may be referred to by various names, such as *I/O Timeout* or *Port Timeout*. A value of 0 usually disables it, while a value of 300 seconds is usually the appropriate (and maximum) value. However a value of 0 does not disable the setting on HP JetDirect print servers. This value cannot be set (cannot be disabled) on the IBM Network Printers and earlier Infoprint workgroup printers. Be aware that if the timeout is disabled, no other host or print process can acquire that printer until the original job releases hold of the printer.

Sometimes individual timeouts may also apply to the separate data streams or protocols on the print server, for example IPDS or NetBIOS. This is seen on IBM Infoprint 21.

Print processor timeouts

This is sometimes referred to as *Job Timeout*, *End-of-Job Timeout*, *Idle Timeout*, or *Wait Timeout*. It controls the time in seconds the printer waits before printing the last page of a job. This timeout only comes into effect if the printer does not receive a page eject at the end of a print job.

A symptom of the latter is seen when the transmission of a following job flushes the last page of the previous job out of the printer. It should be set to a high value, the maximum value or disabled, if possible. A typical value is 300 seconds. Avoid setting the timeout too low (5 seconds). A complex job (including transformation) combined with a slow printer could result in the timeout executing and a partial page printed. A value of 0 seconds may disable this feature on some printers.

A.3.3.2 iSeries device configuration

To create an SNMP LAN-attached device description, type the following command:

```
CRTDEVPRT
```

Then press the F4 (Prompt) key. The display in Figure 205 is shown for an OS/400 V4R5 system. The breaks between displays may vary slightly between different OS/400 releases).

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Device description . . . . . . .   RED_SNMP      Name
 Device class . . . . . . . . . .   *LAN          *LCL, *RMT, *VRT, *SNPT, *LAN
 Device type  . . . . . . . . . .   3812          3287, 3812, 4019, 4201...
 Device model . . . . . . . . . .   1             0, 1, 2, 3, 4, 10, 13, 301...
```

Figure 205. Creating a device description for SNMP LAN-attached ASCII printer (Part 1 of 7)

Fill in the parameter values as shown in the example. The device type is always 3812 model 1. Press Enter to continue. The display in Figure 206 appears.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Device description . . . . . . . > RED_SNMP      Name
 Device class . . . . . . . . . . > *LAN          *LCL, *RMT, *VRT, *SNPT, *LAN
 Device type  . . . . . . . . . . > 3812          3287, 3812, 4019, 4201...
 Device model . . . . . . . . . . > 1             0, 1, 2, 3, 4, 10, 13, 301...
 LAN attachment . . . . . . . . .   *IP            *LEXLINK, *IP, *USRDFN
```

Figure 206. Creating a device description for SNMP LAN-attached ASCII printer (Part 2 of 7)

Change the parameter values as shown in Figure 206. Then press Enter. The display in Figure 207 appears.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Device description . . . . . . . > RED_SNMP      Name
 Device class . . . . . . . . . . > *LAN          *LCL, *RMT, *VRT, *SNPT, *LAN
 Device type  . . . . . . . . . . > 3812          3287, 3812, 4019, 4201...
 Device model . . . . . . . . . . > 1             0, 1, 2, 3, 4, 10, 13, 301...
 LAN attachment . . . . . . . . . > *IP           *LEXLINK, *IP, *USRDFN
 Port number  . . . . . . . . . .   2501           0-65535
 Online at IPL  . . . . . . . . .   *YES           *YES, *NO
 Font:
   Identifier . . . . . . . . . .   416            3, 5, 11, 12, 13, 18, 19...
   Point size . . . . . . . . . .   *NONE         000.1-999.9, *NONE
 Form feed  . . . . . . . . . . .   *AUTOCUT        *TYPE, *CONT, *CONT2, *CUT...
 Separator drawer . . . . . . . .   *FILE         1-255, *FILE
 Separator program  . . . . . . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
```

Figure 207. Creating a device description for SNMP LAN-attached ASCII printer (Part 3 of 7)

The parameters that are highlighted on the display in Figure 207 are explained as follows:

- **Port number:** Specify `2501` for IBM Infoprint and Network Printers. Specify `9100` for all other printers, including the IBM 6400 with internal Ethernet adapter.
- **Font Identifier:** Specify `11` (Courier) normally, or if you want, the IBM Infoprint and Network Printers' scalable version of Courier, font ID `416`. The system assigns a point size.
- **Form feed:** Use `*AUTOCUT` for a printer with cut-sheet paper feed, or `*CONT` for a continuous forms printer such as the IBM 6400.

Press Enter once more, and the display shown in Figure 208 appears.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Device description . . . . . . . > RED_SNMP      Name
 Device class . . . . . . . . . . > *LAN          *LCL, *RMT, *VRT, *SNPT, *LAN
 Device type  . . . . . . . . . . > 3812          3287, 3812, 4019, 4201...
 Device model . . . . . . . . . . > 1             0, 1, 2, 3, 4, 10, 13, 301...
 LAN attachment . . . . . . . . . > *IP           *LEXLINK, *IP, *USRDFN
 Port number  . . . . . . . . . . > 2501          0-65535
 Online at IPL  . . . . . . . . .   *YES          *YES, *NO
 Font:
   Identifier . . . . . . . . . . > 416           3, 5, 11, 12, 13, 18, 19...
   Point size . . . . . . . . . .   *NONE         000.1-999.9, *NONE
 Form feed  . . . . . . . . . . . > *AUTOCUT      *TYPE, *CONT, *CONT2, *CUT...
 Separator drawer . . . . . . . .   *FILE         1-255, *FILE
 Separator program  . . . . . . .   *NONE         Name, *NONE
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
Printer error message  . . . . .   *INFO          *INQ, *INFO

                                                                       More...
```

Figure 208. Creating a device description for SNMP LAN-attached ASCII printer (Part 4 of 7)

Note the following parameter:

- **Printer error message:** This defaults to *INQ, but change it to `*INFO`. This way, if a recoverable error occurs (such as paper out), the print writer will continue once the intervention has been corrected. A setting of *INQ would post a message on the printer's message queue that must be answered before printing can continue.

Now press the Page Down key. Then the display in Figure 209 appears.

```
                    Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Message queue  . . . . . . . . .   QSYSOPR        Name, *CTLD, *SYSOPR, QSYSOPR
   Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
 Activation timer . . . . . . . .   *NOMAX         1-2550, *NOMAX
 Inactivity timer . . . . . . . .   *SEC15         1-30, *ATTACH, *NOMAX...
 Host print transform . . . . . .   *YES          *NO, *YES
```

Figure 209. Creating a device description for SNMP LAN-attached ASCII printer (Part 5 of 7)

The parameters are explained as follows:

- **Message queue:** This names the message queue to which operational messages for the printer are sent.
- **Activation timer:** Specifies the amount of time to wait for the printer to respond. If the printer is attached to only one iSeries, the default of 170 (seconds) may be used. If more than one system is sharing the printer, set the value to `*NOMAX` so that the iSeries continually tries to establish a connection.
- **Inactivity timer:** Specifies the amount of time after which the driver program closes the connection when no files are in ready status. A value of `*SEC15` (15 seconds) is usually adequate for sharing, but *SEC30 is also available. If you

choose a numeric value, note that these are in minutes, not seconds. See also the setting for the User-defined options parameter later in this example.

- **Host Print Transform:** This is necessary for the print data EBCDIC to ASCII conversion (either SCS or AFPDS to ASCII).

Press the Enter key to continue, and the display in Figure 210 appears.

```
                  Create Device Desc (Printer) (CRTDEVPRT)

 Type choices, press Enter.

 Message queue  . . . . . . . . . . > QSYSOPR       Name, *CTLD, *SYSOPR, QSYSOPR
   Library  . . . . . . . . . . .       *LIBL       Name, *LIBL, *CURLIB
 Activation timer . . . . . . . . . > *NOMAX        1-2550, *NOMAX
 Inactivity timer . . . . . . . . . > *SEC15        1-30, *ATTACH, *NOMAX...
 Host print transform . . . . . .    *YES           *NO, *YES
 Manufacturer type and model  . . > *IBM4322
 Paper source 1 . . . . . . . . .    *LETTER         *MFRTYPMDL, *LETTER...
 Paper source 2 . . . . . . . . .    *LETTER         *MFRTYPMDL, *LETTER...
 Envelope source  . . . . . . . .    *C5            *MFRTYPMDL, *MONARCH...
 ASCII code page 899 support  . .    *NO            *NO, *YES
 Image configuration  . . . . . .    *NONE          *NONE, *IMGA01, *IMGA02...
 Character identifier:
   Graphic character set  . . . .    *SYSVAL        1-32767, *SYSVAL
   Code page  . . . . . . . . . .                   1-32767


                                                                        More...
```

Figure 210. Creating a device description for SNMP LAN-attached ASCII printer (Part 6 of 7)

Note the following parameters:

- **Manufacturer type and model:** Enter a value closest to your printer type. The value shown here (`*IBM4322`) really is for an IBM Infoprint 21!
- **Paper sources:** Enter a value appropriate for your printer and country or region, for example `Letter` or `A4`.

Press the Page Down key to continue. The display in Figure 211 appears.

```
                Create Device Desc (Printer) (CRTDEVPRT)

Type choices, press Enter.

Remote location:
  Name or address . . . . . . .   1.2.3.4



User-defined options . . . . . .   *IBMSHRCNN    Character value, *NONE
               + for more values
User-defined object:
  Object . . . . . . . . . . . .   *NONE         Name, *NONE
    Library  . . . . . . . . . .                 Name, *LIBL, *CURLIB
  Object type  . . . . . . . . .                 *DTAARA, *DTAQ, *FILE...
Data transform program . . . . .   *NONE         Name, *NONE
  Library  . . . . . . . . . . .                 Name, *LIBL, *CURLIB
System driver program  . . . . .   *IBMSNMPDRV
Text 'description' . . . . . . .   IBM Infoprint 21 via SNMP driver

                                                                  Bottom
```

Figure 211. Creating a device description for SNMP LAN-attached ASCII printer (Part 7 of 7)

The parameters that are highlighted in Figure 211 are explained as follows:

- **Remote location:** This is the IP address of the printer, or its equivalent host name.
- **User-defined options:** The value of `*IBMSHRCNN` causes the printer driver to open and close the connection to the data port on the printer for every copy of every spooled file. This allows multiple writers and systems to access the printer even if there are more copies or RDY files to be processed. If this option is specified it overrides the setting of the Inactivity timer.
- **System driver program:** Use `*IBMSNMPDRV`.

The value for the User-defined options is *required* for an IBM Infoprint 21. For other printers, including printers that are not shared with other systems, the use of this option is not required (the *IBMSNMPDRV system driver program is sufficient).

If any other parameter values are offered, choose the defaults and press Enter to create the device description.

A.3.3.3 When to use SNMP rather than PJL

Using the *IBMSHRCNN user-defined option provides true network sharing. This means that the network connection to the printer interface is opened and closed for every copy of every spooled file. We therefore rely much less on setting hardware timeouts.

Another factor is that the PJL method is more prone to driver failures caused by simple, common interventions such as “paper out” or “paper jam” occurrences. With the correct printer and software settings, including OS/400 PTFs, these can largely be avoided. However, the SNMP driver has the advantage of being able to notify the print writer that operator intervention is required.

Because the SNMP driver method is the preferred method for printers, such as the IBM Infoprint 21 and IBM 6400 ASCII printers, we recommend you adopt this method for new LAN-attached printer installations. Existing printer configurations may be left as PJL, or migrated to an SNMP driver method.

A.3.3.4 Migrating a PJL printer device description to SNMP

To do this, perform the following steps:

1. End the PJL printer writer.
2. Vary the printer offline.
3. Change the device description:
   a. Add `*IBMSHRCNN` to the User-defined options parameter (notice “options” not “object”).
   b. Change the System driver program to `*IBMSNMPDRV`.
4. Vary the printer online.
5. Start the writer and test printing with some spooled files.

Note that the output queue (and any spooled files on it) is unaltered, and the printer device description uses all other configuration parameter values as before.

A.4 Other methods of printing

The following methods may also be used to submit print jobs to an iSeries printer. However, they do not involve directly creating a printer device description on the iSeries server. These are not the only other methods either, but they are the most common (or likely to become the most common).

A.4.1 Internet Printing Protocol (IPP)

Starting with OS/400 V5R1, printing using the Internet Printing Protocol is supported. This support provides an IPP Server (the iSeries) that responds to print requests from IPP-enabled clients (such as Windows 2000) and sends the print file to any iSeries-attached printer. This support is not yet at the stage where remote printers on the Internet may be used by the iSeries, but a number of useful features are enabled with IPP support, such as:

- Single page range support
- Cancel/hold/release/restart print job

Strictly speaking, the iSeries is continuing to use one of the existing print drivers (PSF, HPT, etc.), but it appears to the user as a different printer driver process. Note that the iSeries implementation is also not restricted to LAN-attached printers.

For more information, refer to 3.1, “Internet Printing Protocol (IPP) Server” on page 87.

A.4.2 PC5250 printer emulation

This heading covers a wide area, including both SNA and TCP/IP connections with the Personal Communications 5250 (PC5250) product. We briefly review the latter method of connection.

TCP/IP printer emulation support is a printer emulation session that runs alongside (or instead of) a conventional 5250 terminal emulation session. It requires that the PC is up and running. Other users can also access this printer session (although they are usually created on a one printer-to-one terminal session basis). However if the PC is offline (switched off or rebooted), the printer session is temporarily unavailable.

This support is available with all versions of Client Access, namely:

- Client Access/400 Enhanced for Windows 3.1
- Client Access/400 for Windows 95/NT
- Client Access Express for Windows

But PTFs are required in some cases. Refer to either the SupportLine Knowledge Base as explained at the beginning of this chapter, or to the Client Access Web page at: `http://www.ibm.com/iseries/clientaccess`

Creation and configuration is done through the PC5250 session (Communication - Configure...). A virtual printer description (device class *VRT, and usually of type 3812 model 1) and output queue with active print writer are automatically created on the iSeries when configuration is successful. Unless you specified otherwise, the printer device name is the same as that of your terminal session, for example QPADEV0001. Version 5 Release 1 of Client Access Express for Windows added some useful enhancements in this area, including the use of the Windows PC name for the printer session or that of the signed on user.

Unlike the printer attachment methods we have seen so far, the default setting for this device is not to use Host Print Transform. In other words, the EBCDIC to ASCII transform is performed by the PC5250 session, not by OS/400 itself. Therefore, you have little control over how your output looks, and worse, may have different output on different printers. However, there is a simple radio button setting in the PC5250 Configure menu option that enables Host Print Transform.

A.5 Summary of LAN attachment methods

Table 30 summarizes the methods by which selected IBM ASCII printers and print devices may be driven from the iSeries server.

Table 30. LAN-attached ASCII printer support

Printer or print device	Supported driver methods	Notes
Infoprint Color 8	LPR, PJL, IPP	IPP refers to native IPP printer support
Infoprint 12	LPR, SNMP	
Network Printer 12, 17, 24	LPR, PJL, SNMP	
Infoprint 20, 32, 40	LPR, PJL, SNMP	
Infoprint 21, 70	LPR, PJL, IPP, SNMP	SNMP recommended. IPP refers to native IPP printer support
Network Print Server feature	LPR, PJL	For PJL, the attached printer must also support PJL
IBM 6400	LPR, SNMP	SNMP support via the integrated Ethernet adapter only

Refer to Table 30 only for *ASCII* printers. If you have an IPDS version of one of these printers, such as the Infoprint 21 or 70, the primary LAN attachment method for the iSeries is an IPDS device description.

Appendix B. Infoprint Designer for iSeries projects

Table 31 describes the projects that are included with Infoprint Designer for iSeries (5733-ID1). It indicates the specific form definition and page definition features that each one demonstrates. See Chapter 2, “Advanced use of IBM Infoprint Designer for iSeries” on page 33, for step-by-step instructions on how each project was created.

Table 31. Infoprint Designer projects included with software

Project	Description	Form definition features demonstrated	Page definition features demonstrated
BASIC	Basic data layout with Invoice overlay. Simplex.	Single copy	Mapping most but not all fields. Changes in position and fonts.
TWOCPY	Two copies of each page - Invoice and Packing Slip.	Subgroups to select input bins and overlay, and to select suppression of prices on packing slip.	Field mapping and suppression of prices for packing slip.
BACK1	Like BASIC with Terms & Conditions page on back.	Constant back.	Field mapping.
BACK2	Like TWOCPY. With Terms & Conditions on the back of the invoice. Blank on the back of the packing slip.	Subgroups and constant back.	Field mapping and suppression of prices for packing slip.
COND	Like TWOCPY with Conditional Processing based on “Continued”.	Subgroups to select input bins and overlay, and to select suppression of prices on packing slip.	Different page map used for Total page versus Continuation pages based on conditional processing.
COMBO	Subgroups, constant back, and conditional processing.	Subgropus and constant back.	Different page map used for Total page versus continuation pages based on conditional processing.
COR*	Landscape project to simulate Computer Output Reduction (COR).	Overlay used is rotated 90 degrees.	Data mapping rotated down. Manual manipulation of line spacing.
SEEDS*	Multiple logical pages printed per physical page.	Overlay used is rotated 90 degrees.	Manual duplication and repositioning of printlines.

* These projects are not included in the initial shipment of Infoprint Designer. They can be obtained by ordering the most current PTF for the product.

Appendix C. Extracting AFP resource contents

This appendix describes how to write a simple Java program to extract the contents of AFP resource objects.

C.1 When to use this program

AFP resources, as a rule, are considered platform independent. This is true for AIX, Windows NT, and OS/390. It is relatively easy to move resources, such as overlays, page segments, fonts, form definitions, and page definitions between these other platforms.

It is also fairly easy to move those resources to the iSeries server. Prior to V5R1, this was a two-step operation. First, transfer the resource AFP data stream into an OS/400 physical file. Then, run the appropriate create command, namely CRTOVL, CRTPAGSEG, CRTFNTRSC, CRTFORMDF, or CRTPAGDFN. With V5R1, these steps are combined in the interface, referred to as AFP Manager, which is included in Chapter 9, "AFP Manager" on page 213.

There are times when a company must move AFP resources to other platforms. They may have custom resources that they want to use on another platform but no longer have the source to recreate them. For example, one division of a company may have created an overlay using the AFP driver (see 1.2.5, "AFP Driver for Windows" on page 18) based on a word processing document. Another division of the same company that runs its operations on AIX may decide that it wants to use the same overlay. If the original word processing document is not available, the second group of users would have to recreate the overlay from scratch.

The problem is that, as part of the iSeries architecture, all objects have an object context area that contains additional information to manage security, track changes, control how the object is to be used, and so on. For example, you can update the contents of a file or run a program, but you cannot "run" a file or update the contents of a program. In this light, there are no standard iSeries server commands shipped with OS/400 that allow you to access the contents of the AFP objects.

The AFP Utilities for iSeries licensed program includes the Convert Overlay to Physical File Member (CVTOVLPFM) and Convert Page Segment to Physical File Member (CVTPAGSPFM) commands. These are used to extract the AFPDS information from iSeries overlays and page segments and put it into physical file members that can then be transferred to the other platforms. However, if a customer does not have this package, or they need to exchange other types of resources, they do not have a simple system tool to use.

The solution to this problem is to use a Java program and the IBM Toolbox for Java to extract the AFP data stream from the resource.

C.2 Requirements

In this appendix, we assume that you are compiling and running the Java program from a Windows client. In the course of running the program, you sign on to the iSeries server, and the contents of the requested objects are copied to the local directory on the client. A TCP/IP connection is required between the iSeries server and the client. The specific instructions assume that you are using a Microsoft Windows client, although the Java developers kit and IBM Toolbox may be installed on a variety of platforms.

To create and run this program from a Windows environment, you must have:

- The Java Software Developers Kit (SDK)
- IBM Toolbox for Java

The Java Software Developer Kit and Toolbox for Java are included with every OS/400 that is shipped. They are also preloaded on new iSeries servers, but must be separately installed when upgrading to the current OS/400 operating system on existing RISC systems. For information on compiling and running Java programs directly on the iSeries, see the iSeries Information Center at:
`http://publib.boulder.ibm.com/html/as400/v5r1/ic2924/index.htm`

This appendix describes a scenario that works even if you do not have the Java licensed programs installed on the iSeries.

C.2.1 Java Software Developers Kit

The following steps explain how to download the Java Software Developers Kit from the Sun Microsystems Java Technology site. The sites may change over time, so use your best judgement to navigate if the Web site has changed and the same path is not apparent. Versions and releases of the packages change frequently, so use the designation for the current release where appropriate:

1. Log on to: `http://www.javasoft.com`
2. Select **Products & APIs**.
3. Select **Java™ 2 Platform, Standard Edition (J2SETM)**.
4. Select **Java™ 2 SDK, Standard Edition, v x.x.x (SDK)**.
5. Select your platform (for example, **Microsoft Windows**).
6. Download **Java 2 SDK, v x.x.x Software for Windows 95 / 98 / 2000 / NT / ME 4.0 (Intel Platform)**. Select one large bundle (approximately 31 MB) or multiple disk size pieces (less than 1.44 MB each). If you choose the multiple disk option, you must also download the installation instructions. Click the **Continue** box.
7. Read the Terms and Conditions document and click the **ACCEPT** box at the bottom.
8. Select the appropriate FTP site for your geography.
9. Use the Save As... dialogue box to display the target directory.
10. Open the directory where you saved the files and run the installation program (**j2sdk-x_x_x_x-win.exe**). Or, follow the installation instructions for the multiple file download. A folder called `jdkx.x.x_x` is created on your workstation. It contains the Java compilation and runtime code.

C.2.2 IBM Toolbox for Java or JTOpen

The IBM Toolbox for Java is a library of Java classes that give Java programs easy access to iSeries or AS/400 data and resources. JTOpen is the open source version of Toolbox for Java.

There are two ways to obtain the IBM Toolbox for Java:

- Client Access Express
- IBM Toolbox for Java Web site

C.2.2.1 Loading IBM Toolbox for Java from Client Access Express

IBM Toolbox for Java is shipped as part of Client Access Express. If you have not yet installed it, perform Selective Setup. For information on installing Client Access components, refer to *Client Access Express for Windows - Setup*, available from the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

The installation process of Client Access Express Java Toolbox creates a directory (\Program Files\IBM\Client Access\jt400\lib). Within that directory, a file called JT400.zip or JT400.jar exists, depending on what release of Client Access you are using. If you have the JT400.zip file, do *not* unzip it.

C.2.2.2 Loading IBM Toolbox for Java from the Web

If you are not running Client Access Express, follow these steps to download the IBM Toolbox for Java from the IBM Toolbox for Java and JTOpen Web site at:

`http://www-1.ibm.com/servers/eserver/iseries/toolbox/downloads.htm`

1. From the Web site, select the link to IBM Toolbox for Java and JTOpen.
2. You are prompted to supply a user ID and password to register.
3. Enter the registration information and click **Accept License**.
4. Select **Single File Download: jtopen_x_xx.zip**. It is a very large file (over 13 MB) with over 1,300 files. You only need one file from within it.
5. Select a temporary directory to which to download it.
6. Create a directory called `JTOpen` on your hard drive.
7. Run an unzip program. Select only the **/lib/jt400.jar** file and direct it to the JTOpen directory.

C.2.3 Preparing your Windows environment

Once you obtain the necessary software, modify your Windows environment to access the necessary resources automatically. You do this by updating PATH and CLASSPATH. The method for doing this depends on the version of Microsoft Windows operating system you are running.

C.2.3.1 Setting PATH and CLASSPATH in Windows 95 or 98

For Windows 95 or Windows 98, the additions are made in the AUTOEXEC.BAT file.

These are examples of PATH and CLASSPATH statements that are added to the end of an AUTOEXEC.BAT file. In this case, the JTOpen version of the toolbox is downloaded from the Web, which explains the JTOpen entry:

```
rem path & classpath entries added for AS400 toolbox - extract resources
SET PATH=%PATH%;c:\jdk1.x.x_xx\bin
SET CLASSPATH=%CLASSPATH%;c:\JTOpen\lib\jt400.jar
```

C.2.3.2 Setting PATH and CLASSPATH in Windows NT or 2000

For Windows NT or 2000, update PATH and CLASSPATH using a GUI interface:

1. Right-click **My Computer**.
2. Select **Properties.**
3. Select the **Environment** tag.
4. Enter `PATH` for Variable and `c:\jdk1.x.x_xx\bin` for Value.
5. Repeat these steps for CLASSPATH and the appropriate entry for the jt400.* file.

C.2.4 Entering and compiling the program

The sample program is listed in C.2.6, “Sample Java program listing” on page 335. You may enter the program using any ASCII text editor. If you are reading this document in a softcopy format, you may be able to cut and paste the code. Windows Notepad suffices for a short program such as this one. There are other more sophisticated editors for Java programming available that provide built-in formatting and color coding. VisualAge for Java is such an offering from IBM.

Give your Java source program a `.java` file extension. The name of the file must match the name given in the *class* statement, which is ExportAFP.java in this example. Be aware that Java language, and even the program name, is case-sensitive. To compile the program, enter the following information at C:\>:

```
javac ExportAFP.java
```

This creates the ExportAFP.class class module.

C.2.5 Running the sample program

The program expects the following parameters to be passed to it:

- Library name
- The names of any number of AFP resource objects

The AFP resource object names must be entered with the appropriate name extension as shown in Table 32.

Table 32. Resource file name extensions

Resource object type	Resource file name extension
Overlay	.OVL
Page segment	.PAGSEG
Font resource	.FNTRSC
Page definition	.PAGDFN
Form definition	.FORMDF

To extract the overlay called INVOICE and three page segments (SS2TOP, SSBOT1, and SSWM) from library MSHNIER, enter the following command:

```
java ExportAFP MSHNIER INVOIC.OVL SS2TOP.PAGSEG SSBOT1.PAGSEG SSWM.PAGSEG
```

Note

As mentioned earlier, the program name is case-sensitive. However, the program was written to accept the parameters in upper or lowercase. Therefore, it would be equally acceptable to enter the command as:

```
java ExportAFP mshnier invoic.ovl ss2top.pagseg ssbot1.pagseg sswm.pagseg
```

The iSeries server name, user ID, and password are not hard coded in the program. You are prompted to enter that information at run time as shown in Figure 212. You may enter the system name or its IP address (such as 9.29.5.13).

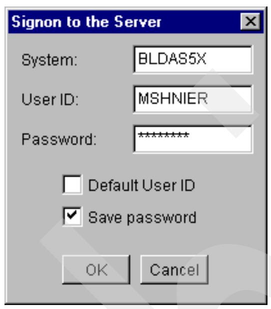


Figure 212. Enter system name, user ID, and password

If you want, you can modify the program to include any, or all three, of the parameters. See the comment lines in the program listing that begin with "//" for syntax. If you choose to hard code your password in the program, you must make sure you protect your copy of the program source and compiled class object from unauthorized use.

The names given to the exported objects are generated by adding `.afp` to the end of the original object name. For example, the contents of INVOICE.OVL is exported to invoice.ovl.afp. The IBM AFP Viewer may be used to view overlay and page segment objects that use this naming convention.

The program writes a status message for each file it extracts. Figure 213 shows an example of the screen you see after running the program.

```
C:\AFPJava>java ExportAFP MSHNIER INVOIC.OVL SS2TOP.PAGSEG SSBOT1.PAGSEG SSWM.PAGSEG
Exporting /QSYS.LIB/MSHNIER.LIB/INVOIC.OVL to C:\AFPJava\invoic.ovl.afp
Exporting /QSYS.LIB/MSHNIER.LIB/SS2TOP.PAGSEG to C:\AFPJava\ss2top.pagseg.afp
Exporting /QSYS.LIB/MSHNIER.LIB/SSBOT1.PAGSEG to C:\AFPJava\ssbot1.pagseg.afp
Exporting /QSYS.LIB/MSHNIER.LIB/SSWM.PAGSEG to C:\AFPJava\sswm.pagseg.afp
```

Figure 213. Example of running Java ExportAFP program

C.2.6 Sample Java program listing

Figure 214 contains the sample program described in this appendix.

```
import com.ibm.as400.access.AS400;
import com.ibm.as400.access.AFPResource;
import com.ibm.as400.access.PrintObjectInputStream;

import java.io.File;
import java.io.FileOutputStream;

public class ExportAFP {
  public static void main(String args[]) {

    AS400 as400 = new AS400();

    // You may hardcode the system, username and/or password as follows.
    // AS400 as400 = new AS400("system", "username", "password");

    try {

      int resourceCount = args.length;

      String libraryPath = "/QSYS.LIB/"
                         + args[0].toUpperCase()
                         + ".LIB/";

      String resourcePath;
      AFPResource resourceObj;

      File outputFile;
      FileOutputStream outputStream;
      PrintObjectInputStream afpDataStream;
      byte[] buffer = new byte[4096];
      int bytesRead;

      for ( int i=1; i<resourceCount; i++ ) {
        resourcePath = "/QSYS.LIB/"
                     + args[0].toUpperCase()
                     + ".LIB/"
                     + args[i].toUpperCase();
        resourceObj = new AFPResource(as400, resourcePath);
        afpDataStream = resourceObj.getInputStream();
        outputFile = new File(args[i].toLowerCase() + ".afp");
        outputStream = new FileOutputStream(outputFile);

        System.out.println("Exporting "
                         + resourcePath
                         + " to "
                         + outputFile.getAbsolutePath());
        while ( (bytesRead = afpDataStream.read(buffer)) != -1 )
          outputStream.write(buffer, 0, bytesRead);

        outputStream.close();
        afpDataStream.close();
      }

      as400.disconnectAllServices();
      System.exit(0);
    } catch (Exception fmv) {
      System.out.println(fmv);
      as400.disconnectAllServices();
      System.exit(-1);
    }
  }
}
```

Figure 214. Sample Java program for extracting AFP resources

Appendix D. Output queue monitor

Some of the formatting tools described in this document may require you to change the attributes of your printer files. For example, if you want to use the form definitions and page definitions created using Infoprint Designer for iSeries or PPFA, you must use printer files that are defined with DEVTYPE(*LINE). Other parameters in the printer file also must be changed, such as the FORMDF and PAGDFN parameters.

Unfortunately, many customers may not be in a position to make these changes for any of the following reasons:

- They do not have access to the source code from their vendor.
- They use the same printer file (often QSYSPRT) for all applications.
- They are concerned about having the changes overwritten if their vendor ships a new release of their software.
- They do not have the programming skills to make the changes.

One solution is to build an output queue monitor. This document provides guidance on how to do this.

D.1 How an output queue monitor works

The spooled files continue to be generated by the application untouched. Each output queue you want to monitor has an entry in its definition for the DTAQ parameter. This points to an existing data queue object. Multiple output queues may point to the same data queue. Any time a spooled file is placed in the output queue in a RDY state, or if its state changes to RDY, an entry is added to the data queue with the spooled file name, job identifier, and output queue name.

The output queue monitor, a never-ending batch job, constantly checks for information arriving in the data queue. When it finds a new entry, the logic of the program can determine what to do with that spooled files and take appropriate action.

Do not use an output queue that is associated with an active printer writer for monitoring. You do not want conflicts between the writer trying to print your spooled file while the monitor attempts to perform other processing on it.

D.1.1 Basic logic of an output queue monitor

Your output monitor likely requires a combination of CL and some other high level language, such as RPG. Use CL commands to perform the object manipulation, such as changing spooled file attributes. The RPG, or other language programs, may be used for the table lookups or to call system API programs.

Assuming you have SCS spooled files, and you cannot modify the way they are created, you may want to use Infoprint Designer to enhance your output. This would be the basic logic to use in your monitor programs:

1. Use the system API QRCVDTAQ to read the record from the data queue and to identify the spooled file you want to process. For more information about

using QRCVDTAQ, refer to the iSeries Information Center at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

To learn more about using the data queue reference in an output queue, and the layout of the information within it, see *iSeries Printer Device Programming*, SC41-5713.

2. Perform a table or file lookup to find out how that file should be handled. For example, what is the destination output queue and the name of the form definition and page definition objects that you created using Infoprint Designer that are to be used?

   In some cases, enough information is provided from the data queue entry, such as the spooled file name, job or user name, or output queue, to uniquely determine the resulting action. In other cases, you must call system API QUSRSPLA to Retrieve Spooled File Attributes to obtain more information on which to make the decision. The information retrieved about the spooled file includes data stream type, page size, overlay name, user data, and form type, among other things. For more information about QUSRSPLA, see *AS/400 System API Reference*, SC41-5801, or the iSeries Information Center on the Web.

3. Use the Copy Spooled File (CPYSPLF) command to copy the spooled file to a physical file. To preserve the page and line spacing of the original spooled file, you must capture the carriage control information in the first byte of the file. To do this, make sure you specify `CTLCHAR(*FCFC)` in your CPYSPLF command, and make sure the target physical file record length is one more than the longest record in the spooled file.

4. Create or modify a printer file based on the requirements of the application. In the case of Infoprint Designer applications, you must specify `DEVTYPE(*LINE)`, the form definition name (`FORMDF`), the page definition name (`PAGDFN`), and the target output queue (`OUTQ`). You must also specify `CTLCHAR(*FCFC)` so it knows to use the first byte as the carriage control information rather than as data.

   If the form definition you use specifies duplex or different paper sources, you also must specify `DUPLEX(*FORMDF)` or `DRAWER(*FORMDF)`.

   Note

   The newly generated spooled file ends up being “owned” by the user who started the monitor program rather than the original owner. Therefore, the original owner is not able to perform a WRKSPLF command with the default parameters to see their new files. One suggestion is to pick up the original user’s name from the job information that is picked up from the data queue and place it in the USRDTA field of the printer file.

5. Copy the data from the physical file to this new printer file using CPYF. This automatically creates a new spooled file.

6. Delete the original spooled file or move it to a hold or done queue.

Along with the monitor program, a robust system must also include housekeeping functions, such as error checking and table maintenance. If there is a problem, and the monitor ends abnormally, spooled files may have to be held and released to put a record back in the data queue.

D.2 Other examples of use

Output queue monitors are not specific to the conversion of *SCS to *LINE. They can be used any time you require processing on a spooled file external to the original application that created it.

D.2.1 Monitor for sending spooled files to AIX

You can use an output queue monitor to send spooled files to another machine where the destination, or other options, are variable- or parameter-driven. Appendix H in *AS/400 Printing Redbook V*, SG24-2160, describes using an output queue monitor for sending AS/400 spooled files to an Infoprint Manager for AIX server.

D.2.2 Supplementing the APU monitor function

Advanced Print Utility (APU) uses a built-in output queue monitor to automate the reformatting of the spooled files. Each entry in the Monitor Action table contains selection criteria based on the output queue and the spooled file. However, if there are a large number of different unique spooled file designs to be sent to a large number of printers, the number of entries that you require in the APU monitor table may become unmanageable. A custom monitor that selects the print definition and the destination queue independently provides a more manageable solution.

D.3 Sample monitor shell program

This section contains a listing for a shell that may be used as a basis for your own monitor program. The mainline program shown here is CVTS0021. It calls CVTS0022 to perform the file manipulation. You must create CVTS0022 to suit your own requirements, such as conversion from SCS to LINE, including any table lookups to determine the appropriate form definition and page definition names.

You must also add additional instructions in the ERROR HANDLING section.

```
/*                                                                    */
/*    ***********************************************************    */
/*    *                                                         *    */
/*    *    PROGRAM    CVTS0021                                  *    */
/*    *                                                         *    */
/*    *    By         Roger Drolet    rdrolet@ca.ibm.com        *    */
/*    *    DATE       May 1999                                  *    */
/*    *                                                         *    */
/*    *    @ COPYRIGHT IBM CANADA 1999                          *    */
/*    *    This software (source code and compiled objects)     *    */
/*    *    is the property of IBM Canada Ltd.                   *    */
/*    *    No guarantees or warranties are made or implied.     *    */
/*    *    The user is expected to carry out testing to his     *    */
/*    *    or her own satisfaction.                             *    */
/*    ***********************************************************    */
/*                                                                    */
/*                                                                    */
/*    CVTS0021 - READ DTAQ AND CONVERT SPOOL FILE                    */
/*                                                                    */
/*    This Program:                                                   */
/*    - Wait up to 300 seconds to receive an entry from              */
/*      the data queue                                                 */
/*    - If no entry read, it check if the program should end         */
/*      The program stop if:                                          */
/*          - Job was cancel with a ENJOB command                    */
/*          - Subsystem was stop    ENDSBS command                   */
```

```
/*          - System shut down      PWRDWNSYS command                */
/*    - If the entry is "STOP" program stop immediately               */
/*    - Split the entry in fields                                     */
/*    - Call program CVTS0022 to do the spool file conversion */
/*    - Move or Delete the spool file                                 */
/*    - Return wait for an entry                                      */
/*                                                                    */
/*--------------------------------------------------------------------*/
/*                                                                    */
/*      PROGRAM CHANGES                                               */
/*                                                                    */
/*      VERSION   DATE   PROGRAMMER DETAIL                           */
/*              YY/MM/DD                                              */
/*                                                                    */
/*      1.0     99/05    RDROLET     INITIAL VERSION                  */
/*                                                                    */
/*--------------------------------------------------------------------*/

             PGM

             DCL        &DTAQNAME  *CHAR   10  VALUE(CVTSDTAQ)
             DCL        &DTAQLIB   *CHAR   10  VALUE(*LIBL)
             DCL        &ENTLEN    *DEC     5  VALUE(128)
             DCL        &ENTRY     *CHAR  128
             DCL        &WAIT      *DEC     5  VALUE(300)
             DCL        &ENDSTS    *CHAR    1

             DCL        &SPLFIL    *CHAR   10
             DCL        &JOBNAM    *CHAR   10
             DCL        &JOBUSR    *CHAR   10
             DCL        &JOBNBR    *CHAR    6
             DCL        &SPLNBRB   *CHAR    4    /* BINARY    */
             DCL        &SPLNBRD   *DEC     6    /* DECIMAL   */
             DCL        &SPLNBR    *CHAR    6    /* CHARACTER */
             DCL        &OUTQN     *CHAR   10
             DCL        &OUTQL     *CHAR   10
             DCL        &USRDTA    *CHAR   10

/* ---------------------------------------------------------------- */
/* RECEIVE AN ENTRY FROM THE DTAQ (FIRST IN FIRST OUT)               */
/* ---------------------------------------------------------------- */
 RCVDTAQ:    CALL       PGM(QRCVDTAQ) PARM(&DTAQNAME &DTAQLIB +
                          &ENTLEN &ENTRY &WAIT)

/* ---------------------------------------------------------------- */
/* CHECK IF AN ENTRY WAS RECEIVE OR TIMEOUT                          */
/* ---------------------------------------------------------------- */
             IF         COND(&ENTLEN *EQ 0) THEN(GOTO CMDLBL(TIMEOUT))

/* ---------------------------------------------------------------- */
/* CHECK IF THE ENTRY RECEIVE IS THE "STOP" COMMAND                  */
/* ---------------------------------------------------------------- */
             IF         COND(%SST(&ENTRY 1 4) *EQ 'STOP') THEN(GOTO +
                          CMDLBL(END))

/* ---------------------------------------------------------------- */
/* READ THE FIELD FROM THE RECEIVE ENTRY                             */
/* ---------------------------------------------------------------- */
             CHGVAR     &JOBNAM   VALUE(%SST(&ENTRY 13 10))
             CHGVAR     &JOBUSR   VALUE(%SST(&ENTRY 23 10))
             CHGVAR     &JOBNBR   VALUE(%SST(&ENTRY 33 6))
             CHGVAR     &SPLFIL   VALUE(%SST(&ENTRY 39 10))
             CHGVAR     &SPLNBRB  VALUE(%SST(&ENTRY 49 4))
             CHGVAR     &SPLNBRD  VALUE(%BIN(&SPLNBRB))
             CHGVAR     &SPLNBR   &SPLNBRD

/* ---------------------------------------------------------------- */
/* CALL PROGRAM CVTS0022 TO WORK WITH THE SPOOL                      */
/* ---------------------------------------------------------------- */
             CALL       PGM(CVTS0022) PARM(&ENTRY &OUTQN &OUTQL +
                          &USRDTA)

/* ---------------------------------------------------------------- */
/* Move or Delete the spool file                                     */
/* ---------------------------------------------------------------- */

             IF         COND(&OUTQN *EQ '*DELETE') THEN(DLTSPLF +
                          FILE(&SPLFIL) +
```

```
                         JOB(&JOBNBR/&JOBUSR/&JOBNAM) +
                         SPLNBR(&SPLNBR) SELECT(*ALL))
             ELSE       CMD(CHGSPLFA FILE(&SPLFIL) +
                         JOB(&JOBNBR/&JOBUSR/&JOBNAM) +
                         SPLNBR(&SPLNBR) SELECT(*ALL) +
                         OUTQ(&OUTQL/&OUTQN) USRDTA(&USRDTA))

/* ------------------------------------------------------------- */
/* GO READ NEXT ENTRY IN THE DTAQ                                 */
/* ------------------------------------------------------------- */
             GOTO       CMDLBL(RCVDTAQ)

/* ------------------------------------------------------------- */
/* TIME OUT (CHECK IF THE JOB MUST END)                           */
/* ------------------------------------------------------------- */
 TIMEOUT:    RTVJOBA    ENDSTS(&ENDSTS)
             IF         COND(&ENDSTS *EQ '1') THEN(GOTO CMDLBL(END))
             ELSE       CMD(GOTO CMDLBL(RCVDTAQ))

             GOTO       CMDLBL(RCVDTAQ)

/* ------------------------------------------------------------- */
/* ERROR HANDLING                                                 */
/*                                                                */
/* ------------------------------------------------------------- */
 ERROR:
             GOTO       CMDLBL(RCVDTAQ)

 END:        ENDPGM
```

Appendix E. Supplemental Infoprint Server information

This appendix contains additional information needed to develop an e-mail exit program. It also contains detailed field listings for the data queue used by the PDF subsystem.

E.1 E-mail exit program

The following sections contain a template for an e-mail exit program using a combination of RPGLE and CL. This program is referenced in Chapter 4, “Using the Infoprint Server for iSeries PDF transform” on page 103, and Chapter 14, “End-to-end example” on page 283.

You can find a sample C Language program in the *Infoprint Server for iSeries User’s Guide*, G544-5775.

E.1.1 RPGLE e-mail exit program mainline

Use this sample RPGLE program as a template to customize your own e-mail application:

```
FLOOKUP    IF   E           K DISK
D*
D INPUTDS         DS
D  JOBNAM                 1     26
D  SPLFID                27     36
D  SPLNO                 37     40B 0
D  MAILTAG               41    290
D  PDFFILE              291    630
D  RES1                 631    632
D  PATHCCSID            633    636B 0
D  SENDER               637    646
D  USRDTA               647    656
D***********************************************
D OUTDS           DS
D  DISPOSTN               1      1
D  RES2                   2      4
D  MSGLEN                 5      8B 0
D  ADDRLEN                9     12B 0
D  MSGTEXT               13    267
D  RES3                 268    287
D  ADDRESS              288    542
D*
C     *ENTRY        PLIST
C                   PARM                    INPUTDS         656
C                   PARM                    INPUTLEN          4 0
C                   PARM                    OUTDS           542
C                   PARM                    OUTPUTLEN         4 0
C                   PARM                    OUTINFO           4 0
C*
C                   EVAL      MSGLEN    = 255
C                   EVAL      ADDRLEN   = 255
C                   EVAL      OUTPUTLEN = 542
C                   EVAL      OUTINFO   = 542
C*
C                   MOVE      *BLANKS       ADDRESS
C     6             SUBST     MAILTAG       CUSTNO            6
C     CUSTNO        CHAIN     LOOKUP
C                   IF        NOT %FOUND
C                   EXSR      Not_Found
C                   RETURN
C                   ENDIF
C*
C                   MOVEL     EADDR         ADDRESS
C*
C                   EVAL      MSGTEXT = ‘Hello ‘ + %TRIMR(PERSON)
C                                  + ‘, this is your invoice for ‘
C                                  + %TRIMR(COMP) + ‘.’
```



```
C                   EVAL      DISPOSTN = '1'
C*
C                   RETURN
C************************************************************
C     Not_Found     BEGSR
C************************************************************
C
C                   CALL      'NOTFOUND'
C                   PARM                    CUSTNO
C                   PARM                    PDFFILE
C                   EVAL      DISPOSTN = '0'
C                   ENDSR
C*
```

E.1.2 NOTFOUND program called by e-mail

This is a sample program called by an e-mail exit program when the lookup operation fails. You may add other commands to this program to perform other functions such as change authority of the generated PDF file in the IFS or to rename it.

```
PGM        PARM(&CUSTNO &PDFFILE)
DCL        VAR(&CUSTNO)  TYPE(*CHAR) LEN(6  )
DCL        VAR(&PDFFILE) TYPE(*CHAR) LEN(340)

SNDMSG     MSG('Cannot find e-mail address for customer +
             number' *BCAT &CUSTNO *BCAT '.  File' +
             *BCAT &PDFFILE *BCAT 'not sent.') +
             TOUSR(MSHNIER) MSGTYPE(*INFO)
CPY        OBJ(&PDFFILE) TODIR('/mira')
MONMSG     MSGID(CPFA09C CPFA0A0)

ENDPGM
```

E.1.3 Sample DDS for LOOKUP file

This is a sample DDS layout for the lookup file used by the e-mail exit program:

```
A                                       UNIQUE
A          R EMAILR
A            CUSTNO         6A          COLHDG('CUSTOMER' 'NUMBER')
A            COMP          25A          COLHDG('COMPANY' 'NAME')
A            PERSON        25A          COLHDG('CONTACT')
A            EADDR         80A          COLHDG('E-MAIL' 'ADDRESS')
A          K CUSTNO
```

E.1.4 Exit program parameters

The e-mail exit program uses the parameters listed in Table 33.

Table 33. E-mail exit program parameters

Description	Usage	Type	Comments
E-mail exit input information	INPUT	CHAR(*)	The e-mail exit program information that is input to the e-mail exit program from the PSF/400 print writer.
Length of input information	INPUT	BINARY(4)	The length of the input information.
E-mail exit output information	OUTPUT	CHAR(*)	The e-mail exit program information that is output from the e-mail exit program to the PSF/400 print writer.
Length of output information buffer	INPUT	BINARY(4)	The size, in bytes, of the mail tag output information buffer.

Description	Usage	Type	Comments
Length of output information available	OUTPUT	BINARY(4)	Length of the output information available. To avoid losing information, this should be less than the e-mail tag output information.

E.1.5 E-mail exit program input data structure

Table 34 shows the structure for the e-mail input parameter.

Table 34. E-mail input information

Position	Type	Description	Comments
1	CHAR(26)	Qualified job name	
27	CHAR(10)	Spooled file name	
37	BINARY(4)	Spooled file number	
41	CHAR(250)	Mail tag	As provided by DDS STRPAGGRP or generated by CRTAFPDTA
291	CHAT(340)	Path and name of PDF file	
631	CHAR(2)	Reserved	
633	BINARY(4)	Path name CCSID	
637	CHAR(10)	Mail sender	
647	CHAR(10)	User Data (USRDTA)	

E.1.6 E-mail exit program output data structure

Table 35 shows the structure for the e-mail output parameter.

Table 35. E-mail output information

Position	Type	Description	Comment
1	CHAR(1)	Disposition of the PDF file	0=do not mail 1=e-mail to specified address
2	CHAR(3)	Reserved	Set to null
5	BINARY(4)	Length of message text	0 to 255 bytes
9	BINARY(4)	Length of mail address data	0-16,000,000
13	CHAR(255)	Message text data	
268	CHAR(20)	Reserved	Initialized to null
288	CHAR(*)	E-mail address or addresses	Each address must be delimited by single quotation marks

E.2 PDF data queue structure

Table 36 describes the field layout for the PDF data queue. An entry is made in this data queue (if specified in the PSF configuration object) for every PDF file created. You can find information on specifying and using this queue in 4.9.9, “Using the PDF data queue” on page 138, and Chapter 14, “End-to-end example” on page 283.

Table 36. PDF transform data queue parameters

From position	To position	Data type	Description
1	10	CHAR(10)	**Function:** Identifies the function that created the data queue entry. The value for converting a spooled file to PDF is *PDFWTR.
11	12	CHAR(2)	**Record type:** Identifies the record type within the function. For PDF conversion, the value is ‘01’.
13	14	CHAR(2)	**Return value:** 01 - File successfully converted and stored in specific folder. 02 - File successfully converted and placed on specified output queue. 03 - File successfully converted and electronically mailed. 04 - File successfully converted, but e-mail exit program specified not to mail PDF file. 05 - File successfully converted, but not mailed due to invalid mail tag or address. 06 - Conversion error. 07 - File could not be spooled to the output queue.
15	40	CHAR(26)	**Qualified job name:** CHAR(10) - Job name CHAR(10) - User name CHAR(6) - Job number
41	50	CHAR(10)	**Spooled file name:** Identifies the name of the spooled file that was converted to PDF.
51	52	CHAR(2)	Reserved
53	56	BINARY(4)	**Spooled file number:** Identifies the unique number of the spooled file that was converted to PDF.
57	306	CHAR(250)	**Mail tag:** Identifies the mail address or mail tag specified on the printer file USRDFNDTA parameter or the DDS STRPAGGRP keyword.

From position	To position	Data type	Description
307	646	CHAR(340)	**Path and name of the PDF file:** Identifies the path and name of the PDF file.
647	648	CHAR(2)	Reserved
649	652	BINARY(4)	**Path name CCSID**
653	662	CHAR(10)	**Mail sender:** Identifies the sender of the electronic mail. The value is specified on the printer file USRDFNDTA parameter or on the PSF configuration object PDFSENDER parameter.
663	672	CHAR(10)	**User Data**
673	752	CHAR(80)	Reserved

Appendix F. Matrices for choosing a software print solution

As part of your decision-making process for selecting the appropriate software tools, you must consider a number of factors. This appendix summarizes some of the key differentiating features of IBM printing software products based on:

- Document formatting methods
- Building or obtaining external AFP resources
- Transport Mechanisms
- Data stream conversion
- Product packaging

F.1 Document formatting methods

You may consider the following factors when selecting a software product for formatting iSeries documents:

- **Programmer required:** This is whether programming skills are required. It does not imply knowledge of any particular programming language.
- **Application Independent:** Can the document formatting method be used independently of the application or must it be taken into consideration when writing the application?
- **Dynamic positioning:** Can the text be placed at different positions from page to page depending on the amount of data being produced?
- **Complex documents:** At a minimum, different pages within the same document should be able to have different formatting applied (if required).
- **Barcodes:** Can individual fields be printed as individual barcodes?
- **One-pass operation:** This refers to the method of producing the final document output. For example, is any re-spooling of a new file required; or, is the formatting applied inline with the normal print spooling?
- **Cross-platform compatibility:** Can the AFP resources created by the formatting method be used on other platforms? Note that this refers to the resources created rather than the formatting products themselves.

The methods considered are:

- iSeries printer files and DDS printer files
- Infoprint Designer for iSeries
- AFP Printsuite – Advanced Print Utility (APU)
- AFP Printsuite – Page Printer Formatting Aid (PPFA)
- AFP PrintSuite – AFP Toolbox
- AFP Utilities/400 – Print Format Utility

Table 37 compares the functions of the software product for formatting iSeries documents.

Table 37. Document formatting methods – Comparison

Product	Programmer required	Application independent	Dynamic positioning	Complex documents	Barcodes	One-pass operation	Cross-platform compatibility	Charges/licensing
Printer File part of 5722-SS1	Y	Y	N	N	N	Y	N	Included as part of any supported release for OS/400.
DDS part of 5722-SS1	Y	Y[1]	Y	Y	Y	Y	N	Included as part of any supported release for OS/400.
Infoprint Designer for iSeries (5733-ID1)	N	Y	N	Y	Y	Y	N[3]	One licence per system. One price for all systems. One user per license. Additional users via PRPQ, see Table 41 on page 355. Requires a Client Access Express connection. PSF/400 needed to print[2] . Requires AFP Font Collection V4R5 or higher.
APU feature of 5798-AF3	N	N	N	Y	Y	N	N	One license per system. Price based on system size. Supports multiple simultaneous users. APU required on system at run time. PSF/400 required to print[2] . Uses fonts from AFP Font Collection. Any supported release.
PPFA feature of 5798-AF3	Y	Y	N	Y	Y	Y	Y	One licence per system . Price based on system size. Supports multiple simultaneous users. Objects may be used on other systems without PPFA PSF/400 required to print[2]. AFP Font Collection recommended. Any supported release.
AFP Toolbox feature of 5798-AF3	Y	N	Y	Y	Y	Y	Y	One license per system. One price for all systems. Required on system at run time. PSF/400 required to print[2]. Requires AFP Font Collection. Any supported release.
Print Format Utility part of 5722-AF1	N	Y	N	Y	Y	Y	N	Part of AFPU/400. One license per system Multiple users. AFPU/400 required at run time. PSF/400 required to print[2]. AFP Font Collection optional. Any supported release.

Product	Programmer required	Application independent	Dynamic positioning	Complex documents	Barcodes	One-pass operation	Cross-platform compatibility	Charges/licensing

Notes:

1. If you make simple changes to your DDS, you may not have to change or recompile your program. Extensive changes, such as adding new AFPDS keywords, usually require some corresponding programming changes.
2. PSF/400 is required to print to a printer configured with DEVTYPE(*IPDS) and AFP(*YES). PSF/400 is not required to print to an ASCII printer using Host Print Transform, but not all functions may be supported. PSF/400 is not required to generate PDF files using Infoprint Server for iSeries.
3. Objects created using Infoprint Designer may not be modified on any other system, but the objects themselves may be migrated.

F.2 Building or obtaining external AFP resources

The following factors may be considered when selecting a software product for creating AFP resources:

- **WYSIWYG interface:** What you see is what you get – a graphical user interface, usually Windows-based.
- **Objects generated directly on iSeries:** This refers to whether the object is created as an iSeries object by the product rather than being created as an AFP object on another system or workstation that must be uploaded and compiled on the iSeries as a separate step.
- **Resolutions supported:** Fonts are available as 240 or 300 dpi. In the case of 600 dpi printers, 300 dpi fonts are scaled accordingly. Outline fonts are resolution-independent, but the printer must be able to receive them.
- **Convert from TrueType and Type 1:** This is the ability to convert a Windows PC font in either of the two common formats to an AFP font.
- **Cross-platform compatibility:** Can the AFP resources created by the formatting method be used on other platforms? Note that this refers to the resources created (not to the formatting products themselves).

The software products considered are:

- Infoprint Designer for iSeries
- AFP Utilities for iSeries
- AFP Font Collection for Workstations and OS/400
- AFP Fonts/400
- AFP Driver for Windows

Table 38 compares the functions of the software product for creating AFP resources.

Table 38. Building or obtaining external resources: Comparison

	Overlays		Page segments		Fonts			
Product	**WYSIWYG interface**	**Objects generated directly on iSeries**	**WYSIWYG interface**	**Objects generated directly on iSeries**	**Resolutions supported**	**Convert from TrueType & Type 1**	**Cross Platform compatibility**	**Charges/Licensing**
Infoprint Designer for iSeries (5733-ID1)	Y	Y	Y	Y	-	-	Y[1]	One license per system. One price for all systems. One user per license. Additional users via PRPQ. See Table 41 on page 355. Requires a Client Access Express connection. PSF/400 required to print.[2] Requires AFP Font Collection V4R5 or higher.
AFP Utilities for iSeries (5722-AF1)	N[3]	Y	N	Y[4]	-	-	Y[1]	One license per system. Multiple users. AFP Font Collection optional. Any supported release of OS/400.
AFP Font Collection for Workstations and OS/400 (5658-B45)	-	-	-	-	240 300 Outline	Y	Y[2]	One license per system. Any supported release of OS/400. Font conversion tools and utilities are a separate orderable feature of 5648-B45.
AFP Fonts/400, 5769-FN1 and 5769-FNT	-	-	-	-	240	N	Y[2]	One license per system. Any supported release of OS/400. Recommended only for use for compatibility with legacy systems.
AFP Driver for Windows	Y	N	Y	N	-	-	Y	Free from Web (supported only as-is) and shipped and supported with other products, such as Client Access, Content Manager OnDemand, and AFP Workbench.

Notes:

1. Objects created using Infoprint Designer for iSeries or AFP Utilities for iSeries may not be modified on any other system, but the objects themselves may be migrated.
2. Font objects are compatible with fonts on other systems. They are licensed for the one system they are purchased for.
3. AFPU uses a near-WYSIWYG 5250 interface, and a call to the AFP Viewer shows the in-progress design in full WYSIWYG.
4. IOCA version of image required as input.

F.3 Transport mechanisms

Table 39 compares the different methods of sending spooled files to a device and maintaining the connection. Consider the following factors for transporting:

- **Page-level recovery:** Information provided to the operator to identify on which document page an error occurred. Can printing be restarted after an error at the correct page to prevent missing or duplicate pages?
- **Resource management:** Does the system manage external resources and send the correct resource to the printer only when necessary? Resource management allows for resources to be maintained in the device across job boundaries under system control.
- **Efficient data stream:** This is important when printing remotely over a slow speed connection or printing locally to a high speed printer if the network is congested. A data stream based on text is more efficient than one that is image based.
- **AFPDS spooled files:** Can spooled files generated using AFPDS and external resources be sent to the device using this transport method?
- **Page range selection:** Is it possible to select a subset of the pages of a large document?
- **Impact printers supported:** Does the transport method support the data streams used by impact printers?

The transport methods considered are:

- OS/400 support for SCS and IPDS
- PSF/400
- LPR with Host Print Transform
- PJL with Host Print Transform
- SNMP with Host Print Transform

Table 39. Transport mechanisms: Comparison

Product	Page-level recovery	Resource management	Efficient data stream	Support AFPDS spooled files	Page range selection	Impact printers supported	Charges/licensing
OS/400 (*SCS & *IPDS)	Y	N	Y	N	Y	Y	Part of OS/400.
PSF/400	Y	Y	Y	Y	Y	Y	Fee based on the speed of the fastest printer attached to the system.
LPR with HPT	N	N	N	Y[1]	N	Y	Part of OS/400.
PJL with HPT	N	N	N	Y[1]	Y	N	Part of OS/400.
SNMP with HPT	N	N	N	Y[1]	Y	few	Part of OS/400.

Note:

1. AFPDS spooled files are only supported if the target printer supports HP-PCL or PPDS. Some functions not supported.

F.4 Data stream conversion

Table 40 summarizes the available data steam conversions and the products that support them.

Table 40. Data stream conversions

	Target spooled file or object data stream						
Initial spooled file or object data stream	**SCS**	**IPDS with AFP(*no)**	**AFPDS**	**LINE or AFPDSLINE**	**PCL**	**PostScript**	**PDF**
SCS	-	OS/400	PSF/400[3]	Monitor Program [1]	HPT		IP Server
IPDS with AFP(*NO)	OS/400[2]	-	PSF/400[3]				IP Server
AFPDS			IP Server[4]		HPT		IP Server
LINE or AFPDSLINE			PSF/400	-			IP Server
PCL			Infoprint Server		-		
PostScript			Infoprint Server		Image Print Transform[5]	-	
PDF			Infoprint Server				-
TIFF/GIFF/JPEG			Infoprint Server Image Print Transform		Image Print Transform[5]	Image Print Transform[5]	
BMP			Image Print Transform		Image Print Transform[5]	Image Print Transform[5]	

Notes:

1. Although not a direct data stream conversion, a monitor program, as described in Appendix D, “Output queue monitor” on page 337, can facilitate an SCS to LINE conversion.
2. IPDS spooled files may be directed to SCS printers, but there will likely be a loss of function.
3. PSF/400 converts SCS or IPDS to AFPDS, but the AFPDS file is not available to the user. It is immediately converted to printer-specific IPDS and sent to the device.
4. The Infoprint Server CRTAFPDTA command can have an AFPDS spooled file as input. It creates a resource file and a merged file and places the results in the IFS.
5. Image Print Transform works with PostScript Level 1 only.

F.5 Product packaging

Part of the decision making process comes down to the price of the various choices. It is not within the scope of this document to provide prices since they are subject to change and are different for each country or region. This section is intended to provide enough information about the product order numbers and features so that you could go to your local country (region) support to obtain current and accurate prices.

Table 41 lists product names and numbers are current as of OS/400 V5R1. Only the feature numbers pertaining to prices are listed in the table. Other features may be required to complete your software order. This table does not include release-to-release or tier-to-tier upgrade prices.

Table 41. Product number and ordering features

Product name	Product number for V5R1	Feature numbers	Feature description	Comments
Print Services Facility for OS/400	5722-SS1	0672	PSF/400 1-45 pages per minute	PSF/400 is a chargeable feature of the operating system. It includes font objects, and the price is based on the speed of the fastest printer installed. It is required to print to any IPDS printer configured as AFP(*YES). For more information, see 1.3.2, “Print Services Facility/400” on page 21.
		0766	PSF/400 1-100 pages per minute	
		1234	PSF/400 Anyspeed.	
Infoprint Server for iSeries	5722-IP1	0494 0495 0496 0497 0498 0499	P05 P10 P20 P30 P40 P50	Tier priced based on iSeries processor group.
Infoprint Designer for iSeries	5733-ID1	A2JK	Basic feature	Includes overlay creation, data mapping, and image editor functions. Single license per iSeries. One user allowed at a time. For more information, see 1.2.1, “Infoprint Designer for iSeries” on page 15.
	5799-GPW		Additional Client PRPQ[1]	5733-ID1 is a prerequisite.

Product name	Product number for V5R1	Feature numbers	Feature description	Comments
AFP Printsuite for iSeries	5798-AF3	2311 2312 2313 2314 2315 2648	APU P05 APU P10 APU P20 APU P30 APU P40 APU P50	End-user tool to convert simple SCS spooled files to highly formatted AFPDS. Separately orderable feature of Printsuite. Priced by system group. For more information, see 1.1.4, “AFP Printsuite – Advanced Print Utility (APU)” on page 9.
		2304 2305 2306 2307 2308 2642	PPFA P05 PPFA P10 PPFA P20 PPFA P30 PPFA P40 PPFA P50	Programming tool to create form definitions and page definitions. Separately orderable feature of Printsuite. Priced by system group. For more information, see section 1.1.3, “AFP Printsuite – Page Printer Formatting Aid/400” on page 7.
		2385 2386 2387 2388 2389 2654	SAP P05 SAP P10 SAP P20 SAP P30 SAP P40 SAP P50	Convert output from SAP R3 to AFPDS. Separately orderable feature of Printsuite. Priced by system group. See 1.3.6, “SAP R/3 AFP PrintSuite feature” on page 26.
		2138	AFP Toolbox	Programming APIs to generate fully composed, high function documents. Separately orderable feature of Printsuite. One price for all systems. Additional information is provided in 1.1.5, “AFP PrintSuite – AFP Toolbox” on page 11.
AFP Utilities for iSeries	5722-AF1	0001 0002 0003 0004 0005 0006	P05 P10 P20 P30 P40 P50	Create AFP Overlays. Convert IOCA to page segments. Print information directly from database records. Priced based on system group. Additional information is provided1.1.6, “AFP Utilities for iSeries – Print Format Utility” on page 13, and 1.2.2, “AFP Utilities for iSeries” on page 16.
AFP Font Collection for Workstations and OS/400	5648-B45	B6RP	Basic feature	Basic features included with new orders of PSF/400 as of V4R5, or with Infoprint Designer for iSeries. Each feature is priced separately. Additional information is provided in 1.2.3, “AFP Font Collection for Workstations and OS/400” on page 17.
		C0EV	Japanese DBCS	
		C0EW	Korean DBCS	
		C0FA	Simplified Chinese DBCS	
		C0FB	Tradition Chinese DBCS	
		C9RT	Type transformer and utilities for Windows	

Note:

1. Infoprint Designer for iSeries - Additional Client PRPQ provides the ability to license additional concurrent design clients for Infoprint Designer for iSeries (5733-ID1). This licensing is provided by means of a physical hardware key that is attached to the parallel port of the client PC.

Appendix G. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

G.1 IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 358.

- *AS/400 Printing III*, GG24-4028
- *AS/400 Printing IV*, GG24-4389
- *AS/400 Printing V*, SG24-2160
- *J.D. Edwards OneWorld Implementation for AS/400*, SG24-5195
- *How to Replace OfficeVision/400 in Your Applications: Looking at Domino for AS/400 and AS/400 Alternatives*, SG24-5406

G.2 Other resources

These publications are also relevant as further information sources:

- *IBM AFP Fonts: Font Samples*, G544-3792
- *IBM AFP Fonts: Font Summary*, G544-3810
- *IBM AFP Fonts: Licensed Program Specifications*, G544-5229
- *Ethernet and Token-Ring Configuration Guide*, G544-5240
- *Twinax/Coax Configuration Guide*, G544-5241
- *Infoprint Hi-Lite Color Introduction and Planning Guide,* G544-5420
- *IBM Intelligent Printer Data Stream Reference*, S544-3417
- *IBM AFP Fonts: Technical Reference for Code Pages*, S544-3802
- *IBM Page Printer Formatting Aid: User's Guide*, S544-5284
- *IPDS and SCS Technical Reference*, S544-5312
- *AS/400 Guide to AFP and Print Services Facility*, S544-5319
- *PCL5e and PostScript Technical Reference*, S544-5344
- *Advanced Function Printing Utilities for OS/400*, S544-5349
- *AS/400 Advanced Print Utility Users's Guide*, S544-5351
- *IBM AFP Toolbox for AS/400 User's Guide*, S544-5368
- *SAP R/3 AFP Print for AS/400 User's Guide*, S544-5412
- *Infoprint Manager for AIX Reference*, S544-5475
- *IBM Infoprint Manager for AIX: PSF Direct*, S544-5486
- *Infoprint 21: Ethernet/T-Ring Configuration Guide*, S544-5711
- *AFP Traditional Chinese Font Catalog*, SC18-0124
- *AFP Simplified Chinese Font Catalog*, SC18-0133
- *AFP Thai Font Catalog*, SC18-0137

- *AFP Japanese Font Catalog*, SC18-2332
- *Data Stream & Object Architectures: MO:DCA Reference*, SC31-6802
- *Client Access/400 Personal Communications 5250 Reference Guide*, SC41-3553
- *AS/400 Workstation Customization Programming*, SC41-3605
- *iSeries DDS Reference - Version 3*, SC41-3712
- *iSeries Device Programming - Version 3*, SC41-3713
- *AS/400 System API Reference - Version 3*, SC41-3801
- *OS/400 Printer Device Programming V4R2*, SC41-5713
- *AS/400 System API Reference - Version 4*, SC41-5801
- *OV/400 V3R1 Setting Up & Printing*, SH21-0511

G.3 Referenced Web sites

These Web sites are also relevant as further information sources:

- Latest version of the AFP Printer Driver: `http://www.printers.ibm.com/afpdr.html`
- AS/400 Programming Sampler from the IBM Printing Systems products Web site: `http://www.printers.ibm.com/products.html`
- Uniform Code Council and UCC/EAN-128: `http://www.uc-council.org`
- Printing Systems support IBM Printing Systems Digital Library: `http://www.printers.ibm.com/manuals.html`
- Printing Systems support site for Token-Ring and Ethernet microcode: `http://www.printers.ibm.com/util.html`
- Network Printer Manager: `http://www.printers.ibm.com/npm.html`
- "Java AWT: Printing" written by Sun: `http://java.sun.com/products/jdk/1.1/docs/guide/awt/designspec/printing.html`
- IBM Redbooks home page: `http://www.redbooks.ibm.com`

G.4 How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

G.5 IBM Redbooks collections

Redbooks are also available on CD-ROM. Click the CD-ROM button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

J

K

L

IBM @server iSeries Printing VI: Delivering the Output of e-business

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM eserver™ iSeries Printing VI

Delivering the Output of e-business

Learn how to select iSeries printing software for generating e-business output

Gain detailed technical advice on using Infoprint Server and Infoprint Designer

Learn about the new OS/400 V5R1 printing enhancements

Following on from the successful IBM Redbook *AS/400 Printing V*, SG24-2160, this new printing redbook concentrates on products, methods, and techniques used in delivering the output required by IBM eserver iSeries e-business. It introduces and explains many new printing developments with OS/400 V5R1.

In addition, this book offers detailed technical advice on using two of the most significant iSeries printing products for years: Infoprint Server for iSeries and Infoprint Designer for iSeries. *Infoprint Server for iSeries* extends the capabilities of the iSeries beyond printing, to further enable such e-business requirements as data stream and image transforms, spooled file indexing and segmentation, e-mail and Web-enabling. *Infoprint Designer for iSeries* provides the long-awaited, fully-graphical document composition interface to Advanced Function Presentation on the iSeries.

This book will be of value to iSeries customers, IBM Business Partners, and IT specialists who want to learn about the wealth of new printing and output presentation functions available at OS/400 V5R1. It is the essential guide to delivering the output of e-business.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6250-00 ISBN 0738423971