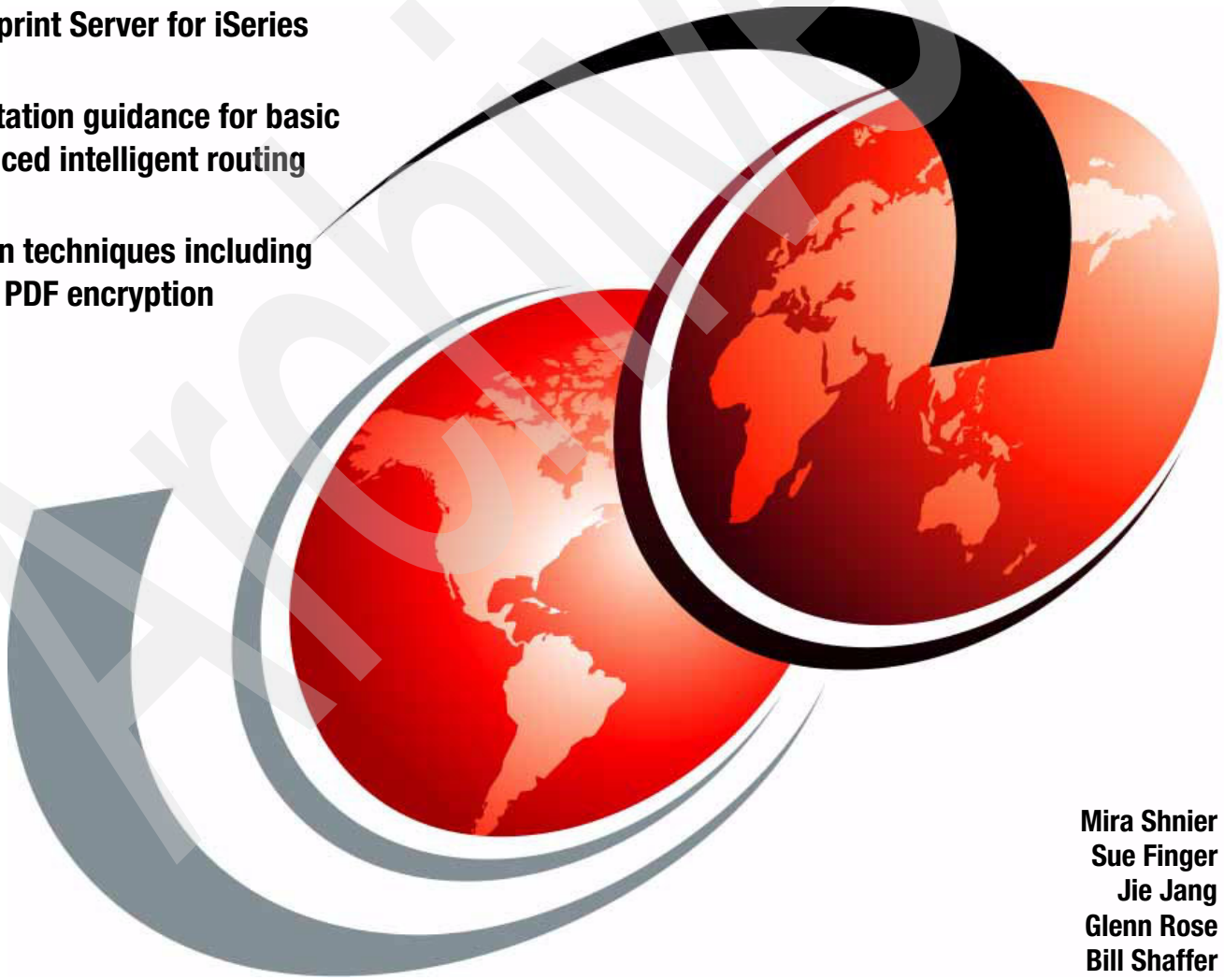
IBM

IBM @server iSeries Printing VII

Infoprint Server Implementation

- High-return e-business output applications using Infoprint Server for iSeries
- Implementation guidance for basic and advanced intelligent routing
- Application techniques including color, fax, PDF encryption



Mira Shnier
Sue Finger
Jie Jang
Glenn Rose
Bill Shaffer

Redpaper

ibm.com/redbooks

International Technical Support Organization

IBM @server iSeries Printing VII: Infoprint Server Implementation

May 2004

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (May 2004)

This edition applies to Version 5 Release 2 Modification 0 and Version 5 Release 3 Modification 0 of Infoprint Server for iSeries (5722-IP1).

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.

Trademarks

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

@server®	Infoprint®	OS/400®
ibm.com®	Intelligent Printer Data Stream™	Print Services Facility™
iSeries™	IBM®	Redbooks(logo) ™
Advanced Function Presentation™	IPDS™	Redbooks™
AFP™	Lotus Notes®	System/36™
AS/400e™	Lotus®	System/38™
AS/400®	Notes®	1-2-3®
Domino®	OfficeVision/400™	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium 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.

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

This IBM® Redpaper builds on the extensive coverage of the IBM Redbook *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250. It expands the implementation guidance for output delivery applications and includes new support available in V5R3. The target audience of this paper is iSeries™ Clients, IBM Business Partners, and IBM IT specialists who need to implement e-business communications solutions with Infoprint® Server.

The principal focus of the paper is intelligent routing. Intelligent routing is a set of capabilities that enable the profiled delivery of business documents and reports to customers and users in the most productive format (print, Web, e-mail, fax). The intelligent routing functions are delivered with two different enabling interfaces. You'll find step-by-step coverage of both interfaces as well as sample programming to get you up and running quickly.

In addition, you'll find related tips and application techniques. These include use of color in iSeries documents, interfacing intelligent routing with fax functions, PDF encryption, and performance considerations.

The team that wrote this Redpaper

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

Mira Shnier is a Senior IT Specialist with the IBM Printing Systems Division in Markham, Ontario, Canada. She provides education and consulting services to IBM Clients, Business Partners, and IBM professionals across Canada and internationally. Mira has 25 years of experience at IBM, of which the last 13 years have been with the IBM Printing Systems Division. She coauthored *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, and made various contributions to AS/400® Printing Redbooks™ II through V. She holds a degree in Chemical Engineering from the University of Toronto.

Sue Finger is a Microcode Test Engineer for IBM U.S. with five years of experience. She has written extensively on e-mail encryption, TCP/IP, and Systems Network Architecture (SNA). Sue has a bachelor's degree in engineering and is currently working on her master's degree from the University of Colorado in electrical engineering-telecommunications.

Jie Jiang is a Software Engineer and Performance Analyst in IBM Printing Systems Division in Boulder, Colorado. She has one and a half years of experience in performance and three years of experience in software programming at IBM. She holds Bachelor of Science and Master of Science degrees in electrical and computer engineering from the University of Colorado at Boulder. Her areas of expertise include performance analysis and different programming languages.

Glenn Rose is the iSeries Program Manager for the IBM Printing Systems Boulder Infoprint Center. He joined IBM in 1968. As Advisory Systems Engineer for government accounts, he became involved with the AS/400 before the announcement in 1989. In 1993, he joined IBM Printing Systems with a responsibility for AS/400 printer sales and technical support. In 1997, he transferred to IBM Printing Systems Division in Boulder, assisting IBM AS/400 clients worldwide with connectivity issues involving IBM network attached printers. Since 1998, he has been involved with several key printing initiatives, teaching iSeries printing, and providing proof of concept assistance for the IBM Printing Systems Division.

Bill Shaffer is the iSeries product manager for output solutions. Based in Boulder, Colorado, Bill is responsible for iSeries tools to manage printing and to re-engineer output for e-business applications. You can contact Bill by sending e-mail to mailto:wcshaff@us.ibm.com.

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

Tim Aden
Rebecca Dimock
Patricia Godfrey
Jack Klarfeld
Miranda Reeves
IBM Printing Systems Division, Boulder, CO

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper 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

1

A look at Infoprint Server for iSeries

This Redpaper provides detailed guidance about the implementation of output management re-engineering using Infoprint Server for iSeries (IBM product 5722-IP1). The major focus is application deployment using the two interfaces provided with Infoprint Server:

- The mapping object interface
- The user exit program interface

It is important to understand the business and application context in which these interfaces are used. This chapter begins with an overview of e-business communications, how Infoprint Server fits in to these types of applications, and specific application examples. It also introduces concepts used by Infoprint Server for iSeries and provides information about the product, the user environment, and a summary of the remaining chapters.

The implementation in this Redpaper assumes that you are at Version 5 Release 2 of Infoprint Server and have applied the major enhancement program temporary fixes (PTFs) delivered in 2003 or later. For additional information about the enhancements available with Version 5 Release 2 and the supplemental PTFs, see Appendix F, “Version 5 enhancements to Infoprint Server” on page 139.

1.1 e-business communications

A central focus of Infoprint Server is e-business communications. The documents and reports in a traditional business communications model use a *print and distribute* methodology. The back-end of business processes is hardcopy production and manual distribution. Many documents and reports use preprinted forms which locked the communication process into hardcopy.

The availability of intranets and the Internet, coupled with new capabilities to produce fully electronic documents, opened rich, new options in how business output is delivered. The fully electronic document breaks the tie to preprinted forms and offers unparalleled flexibility in both document content and document delivery. Electronic pages mean business processes can be re-engineered to enable both profiling of recipients and multi-channel options for delivery. The multi-channel options include Web, client view, e-mail, and fax, and print. The business communications methodology is called *distribute and optionally print*.

As an example, consider an application that produces a monthly sales report. In the print and distribute methodology. The entire report is printed, decollated, and then bursted into logical regional reports. If the report has separate sections for sales regions, then those sections are physically delivered to regional recipients. Contrast that flow with e-business communications. The monthly sales report is produced electronically. At the same time, triggers are inserted in the data stream to delineate or segment the report sections for each sales region. The system separates the report pages by region, producing electronic sub-reports in Adobe Portable Document Format (PDF) or Advanced Function Presentation™ (AFP™) format. Each regional recipient of the reports is profiled as to how they want to receive their report. Individual regional reports can be printed, faxed, e-mailed, archived, or written to the file system for client or Web access.

An important aspect of the transformation made in this example is the business benefits that it provides. If preprinted forms are eliminated from the process, then all the costs associated with those forms—design, acquisition, inventory, handling, and obsolescence—are also eliminated. If printing is eliminated or reduced, then the costs involved in printing—printer hardware, maintenance, and operational costs—are reduced. If hardcopy distribution is eliminated or reduced, then those costs—labor and mailing costs—also go down. These are just the hard cost savings. Electronic delivery provides additional business value such as increased timeliness, greater flexibility, competitive advantage, and higher client or user satisfaction.

1.2 Application examples

Let's look at several additional examples of business communications re-engineering.

1.2.1 Publish or archive output to database

A simple implementation routes documents or reports in PDF format to the integrated file system (IFS). As the spooled files are placed in a PDF writer queue, Infoprint Server converts the print files or print file segments to PDF format and then writes them to a specified directory in the IFS. These PDF files are then available for access via iSeries clients or Web clients. This change also provides an archive for reports. Recent enhancements to Infoprint Server enable full control over PDF file names and directory locations. This simple routing can be accomplished without using the mapping object or user exit programming interfaces.

1.2.2 Electronic report or document distribution via e-mail

This is the example cited earlier with monthly sales reports. Hardcopy distribution can be replaced with e-mail distribution. Virtually every aspect of the e-mail (addressees, body, and attachments) can be customized.

1.2.3 Segmentation of print files (electronic burst and bind)

The ability to segment documents or reports is an essential feature in any business communications deployment. In a hardcopy environment, *burst and bind* refers to the manual process of pulling continuous form printouts off the printer, separating multi-part copies, separating logical report sections, and binding each final segment.

There are two ways to implement this segmentation function electronically. First, printer file data description specification (DDS) contains keywords (Start Page Group and End Page Group) where you can explicitly define start and end of segments during program execution. Second, the Create AFP Data (CRTAFPDTA) command can insert these electronic triggers after the spooled file is created. It keys off predictable data in the file itself. For example, region number is on line 6, column 10 of each page. The unique data element that defines a new segment is called a *routing ID* or *mapping ID*.

1.2.4 Re-queue print files or segments as PDF for printing

An available option after each PDF file is created is to place the PDF back on an iSeries output queue. Why do this? It gives you an additional printing alternative. Many printers now support direct printing of PDF. This means that you can route a hard copy to Printer Control Language (PCL), Intelligent Printer Data Stream™ (IPDS™), and PDF printers.

1.2.5 Intelligent routing of output

The term *intelligent routing* refers to the set of capabilities in the Infoprint Server processing subsystem to select and recognize a print file or print file segment and take distribution action on it based on recipient profiles. The actions can be print, e-mail, fax, Web, and additional routing functions. Much of this Redpaper focuses on the techniques to deploy intelligent routing type applications.

1.2.6 Re-queue print segments as AFP for downstream functions

You can also queue segments back to an output queue in AFP format. This is useful if the action desired on that segment is print, fax, or a similar queue-based function. This Redpaper shows how you can automatically interface print files or print file segments with Facsimile Support for AS/400.

1.2.7 Ad hoc PDF and electronic output functions via iSeries Access

In many cases, the need for electronic delivery of a document or report is not pre-planned but is a real-time (ad hoc) request. iSeries Access (formerly Client Access/400) provides an interactive interface to PDF and delivery functions. You can select one or more print files and direct them to a print queue, to the IFS, or e-mail.

1.3 iSeries output architecture

Let's take a look at the basic iSeries output architecture. Refer to Figure 1-1 and let's walk through the basic print flow. Any OS/400 application that creates print uses a printer file to provide job-level control information as the data is written to the output queue. DDS keywords can define how application and static information are placed on each page. With these instructions, the application places print data into a spooled file and writes the spooled file onto an OS/400 output queue. The print data on the queue is usually either Systems Network Architecture (SNA) Character Set (SCS) format for simple line-mode output or AFP format for any document with graphical elements (for example, images, barcoding, overlays, fonts). With AFP print data, the graphical elements are frequently stored external to the spooled file—in font, overlay, and page segment objects.

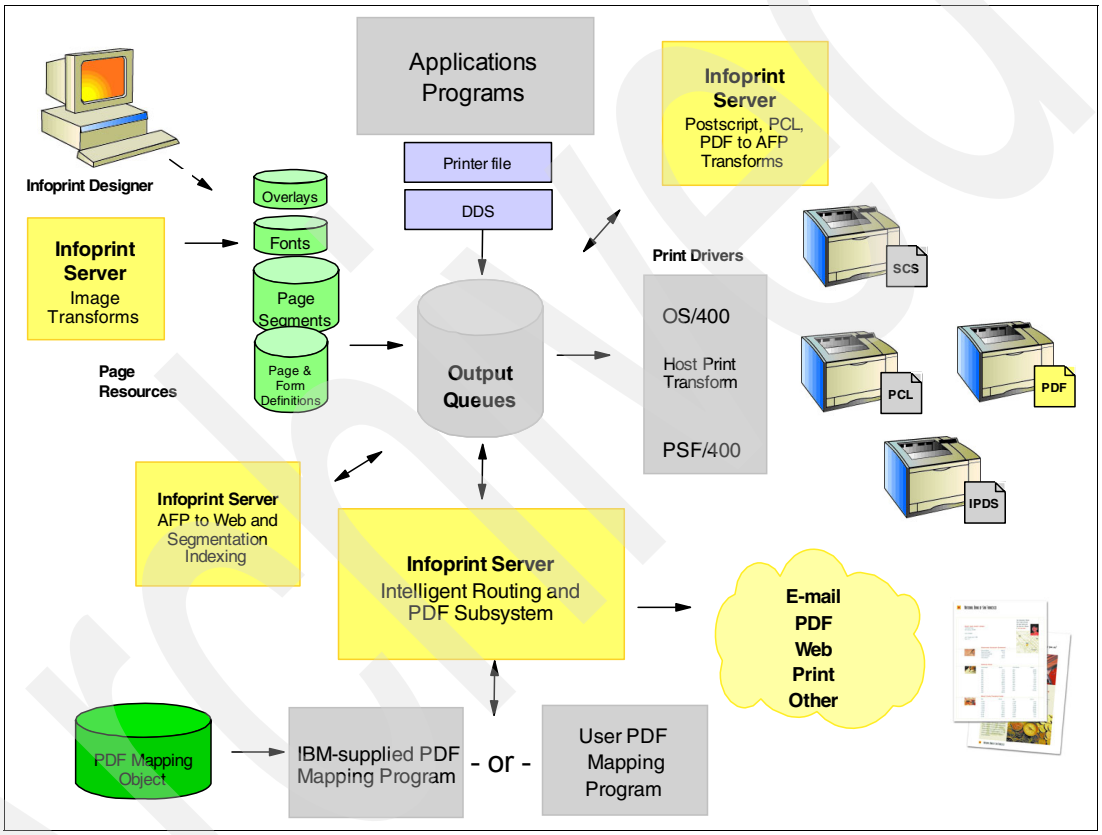


Figure 1-1 iSeries output architecture

When the print data resides on the OS/400 output queue, three drivers can route the print data to a printer. The driver is automatically selected based on the type of target printer. Base OS/400 print management includes the driver for printing to SCS printers. Host Print Transform is the driver for ASCII printers, principally PCL printers. When a PCL printer is selected by the print writer, the print file (either SCS or AFP) is passed to Host Print Transform. Host Print Transform then transforms the print data into ASCII and sends it to the printer.

When an IPDS printer is started, Print Services Facility™ (PSF) for OS/400 is automatically invoked and interactively manages the entire printing process with the printer. When PSF for OS/400 manages AFP print data, it ensures that any external resources required, such as fonts, overlays, and images, are in printer memory when needed.

Two additional object types can be involved in page formatting:

- Page definitions
- Form definitions

These resources are a standard part of the AFP architecture and enable the pages to be formatted independently of the application program. Infoprint Designer for iSeries (IBM program product 5733-ID1) is a fully graphical output composition system that uses these resources to design new applications or re-engineer existing ones.

1.4 What is Infoprint Server for iSeries

Infoprint Server is an iSeries licensed program product (LPP), 5722-IP1. The product is pre-loaded on the OS/400 release CD set and can be installed for 70-day evaluation. Infoprint Server has five functional components:

- PDF services for iSeries
- Intelligent routing output distribution, including e-mail, Web, and file re-queue
- Print file segmentation, AFP indexing, and portable AFP (CRTAFPDTA command)
- PDF, PostScript, and PCL to AFP data stream transforms (priced feature number 5101)
- Image transforms for GIF, TIFF, and JPEG to iSeries format

Again referring to Figure 1-1, let's cover each of these components.

1.4.1 PDF services

A central component of Infoprint Server is the PDF subsystem. This subsystem enables the conversion of any standard OS/400 output data stream (SCS, AFP, IPDS, mixed data, line data, or even OfficeVision/400™ extended formats) to Adobe PDF. The transform creates a native, text-based ASCII PDF data file (assuming the input to the transform is a non-image print file). Highly integrated, the PDF subsystem is deployed as a virtual printer device. This facilitates seamless flow and the handling of the widest variety of input print formats.

Besides producing high-fidelity, text-based PDF, the PDF subsystem supports many advanced PDF characteristics. PDF security functions, such as encryption, password protection, and change protection, can be selected. Advanced AFP page features, such as 2D barcoding and full color image, are supported and can be converted into PDF. iSeries print file indexing, available via DDS or the CRTAFPDTA command, can be set up to flow into the created PDF as bookmarks.

iSeries print file indexing is used in another significant manner with *segmentation*. Segmentation uses the electronic “triggers” placed in the print file by DDS or the CRTAFPDTA command to segment the print file into logical segments. For example, a monthly sales report can be electronically segmented into regional report segments. In a hardcopy context, this is called *burst and bind*. Each segment is handled separately by the PDF subsystem, resulting in a separate PDF file being produced. The segmentation triggers can also contain key variable data that determines the routing or disposition of the segment. See the following section for more information.

There are several interface points with PDF services. PDF processing options can be defined in the printer file, the PDF mapping object, or a PDF exit program. This is the normal, batch application flow. In addition, both iSeries Access and Web Access have interfaces to PDF services. iSeries Access (Navigator) provides a set of interactive panels to define “ad hoc” PDF and intelligent routing operations. Web Access uses the PDF subsystem to bring selected print files to the browser within Adobe Acrobat. There is also a copy spooled file Java™ method to build custom access to PDF services.

1.4.2 Intelligent routing

Central to Infoprint Server is its “e-delivery” or routing capabilities. Working in conjunction with the PDF subsystem, print files or print file segments can be profiled, identified, transformed, and routed as required. A print file or print file segment can be converted into PDF and routed to e-mail, an IFS directory, or an output queue. In addition, a print file or print file segment can be re-queued in iSeries format (AFP) for other routing or delivery functions.

With intelligent routing, any combination of these transform and distribution options can be applied to a print file or print file segment. Essential to this flow is the interfaces that provide the “action” information. Those interfaces include the application printer file, the PDF mapping object, and the PDF exit program. With the printer file, the user-defined data parameter can be used to define action.

With the PDF mapping object and the PDF exit program, the target print file or print file segment is identified and the desired action is defined. With segmented print files, the routing ID in each segment is used to drive the routing action or actions. The PDF mapping object is a data structure that is maintained by a set of iSeries commands. With the PDF mapping object, no programming is required.

With the PDF exit interface, the program communicates directly with the intelligent routing subsystem and provides the required action or delivery information.

1.4.3 Indexing, bookmarking, and portable AFP (CRTAFPDTA command)

The CRTAFPDTA command provides two key functions:

- Indexing
- AFP encapsulation

You can define data within a spooled file as index data and the command builds indexes based on that data. The indexes, in turn, can be used for either segmentation or bookmarking. If you define the spooled file to PDF services as having multiple segments, then the indexes are used to delineate each segment. Note that this enables you to segment a spooled file after the application has created it. Another option for segmentation is to use the Start Page Group keyword in DDS (within the application). If you define the spooled file to PDF services as not having segments, then the indexes become bookmarks within the generated PDF file.

The second function of the CRTAFPDTA command is AFP encapsulation. This means the AFP spooled file, which normally includes references to document components such as overlays, fonts, and page segments, is rebuilt with all the external components placed inline. This complete AFP spooled file can be sent system to system, or in intranet environments, it can become the standard document delivery format. The AFP Viewer is built into iSeries Access. An AFP Viewer plug-in is available for Netscape Navigator or Internet Explorer. This is the AFP equivalent of the Adobe Acrobat Viewer.

1.4.4 PDF, Postscript, and PCL to AFP transforms

Infoprint Server is an integrated set of transforms to convert ASCII print data—PDF, Postscript, and PCL—to AFP for management and further processing by OS/400®. These transforms are available with an optional feature 5101 of Infoprint Server. Working with OS/400 Transform Manager, automatic detection of the ASCII print format is done, the required transform is invoked, and the resulting AFP is placed on an OS/400 output queue. This enables most ASCII output created on OS/400 or in the network to be put in native

OS/400 format (AFP), which lets users take advantage of the OS/400 print or output management capabilities.

1.4.5 Image transforms

A final component of Infoprint Server is its image transforms. These transforms, which run on a Windows® PC, convert industry standard image formats (GIF, TIFF, and JPEG) into AFP page segments or overlays on your PC. You can use the AFP Manager component of iSeries Navigator or OS/400 commands to create the page segment or overlay on your OS/400. This lets you use an image in a client or network application and then embed it in an OS/400 application.

1.5 Intelligent routing flow

It's important to understand the PDF and intelligent routing processes in more detail. Refer to Figure 1-2 for a closer look at how output files flow through these functions, and how profiling, PDF processing, and multi-channel distribution functions take place.

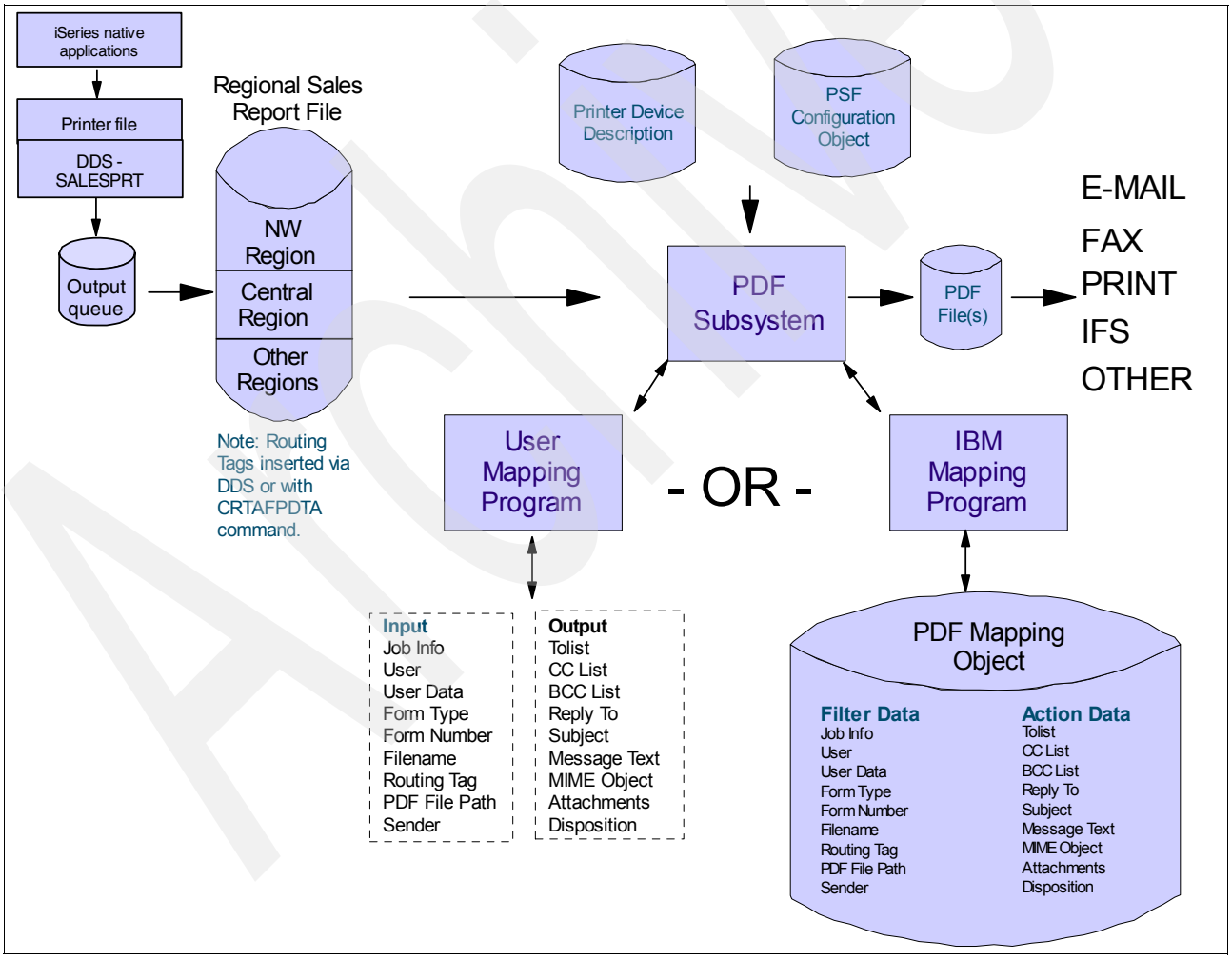


Figure 1-2 PDF and intelligent routing flow

In this example, we are back to the monthly sales report application. We inserted segmentation triggers within the sales report spooled file using either DDS keywords (Start

Page Group, End Page Group) or the CRTAFPDTA command. These triggers delineate the beginning and end of each logical set of pages, in this case, regional subreports.

When the spooled file is placed in a PDF writer queue, then PDF and intelligent routing processing starts. The writer device description (this is a virtual writer, not a real writer) provides general information about processing. The PSF Configuration Object provides more detailed set-up information such as:

- Mail server to be used
- PDF data queue, for communications with exit programming
- AFP respool option
- Administration e-mail ID for error tracking
- PDF mapping program

The last field, PDF mapping program, is key. For simple applications, you can invoke PDF and routing functions without either a PDF mapping object or a custom exit program. However, where data needs to be passed back and forth during processing, then a custom exit program or IBM mapping program must be specified.

As you can see in Figure 1-2, there is a set of input and processing data to be provided. The input or filter data is used to identify the print file or print file segment. The action or output data is the set of processing instructions for that print file or print file segment. The IBM mapping program looks to the PDF mapping object for this information. The PDF mapping object is built using a set of interactive displays. The custom mapping program works directly with the data buffers of the PDF and Routing subsystem, receiving the “input” data on the segment just read and passing back the action data for processing.

1.6 What is in this Redpaper

The remainder of this document is intended to help you take advantage of the new features provided with Infoprint Server for iSeries:

- Chapter 2, “Planning for intelligent routing” on page 11, provides information about configuring the iSeries to take advantage of Infoprint Server’s new intelligent routing features. It also contains background information to assist in the planning process.
- Chapter 3, “Using a PDF mapping object” on page 27, provides detailed information and considerations about using a PDF mapping object.
- Chapter 4, “Encryption for iSeries-generated e-mail” on page 41, provides background information about the encryption technology used by Adobe Acrobat. It follows with information about how the encryption options are selected using Infoprint Server for iSeries.
- Chapter 5, “iSeries color applications” on page 51, explains how an application can be created on the iSeries server and used by Infoprint Server to convert to PDF or be printed on various types of devices.
- Chapter 6, “Using the PDF mapping program” on page 63, provides detailed information and considerations about using a PDF mapping program.

The next three chapters provide information and programming techniques that may be used to extend and enhance the functions provided by Infoprint Server for iSeries:

- Chapter 7, “Using intelligent routing with Facsimile Support/400” on page 99, presents a technique to extend intelligent routing to allow documents to be sent by fax as an alternative or in addition to print, e-mail, or storage in the IFS.

- Chapter 8, “Using APIs to add entries to PDF mapping object” on page 105, provides information about building a program that can be used to add entries to a PDF mapping object as an alternative to adding entities interactively using the user panels.
- Chapter 9, “Saving encrypted PDF files” on page 111, explains how to save the encrypted copies of the PDF file to the IFS. Under normal intelligent routing, only unencrypted PDF files are saved. The encrypted files are normally only created as e-mail attachments.

The appendixes in this Redpaper provide additional reference information:

- Appendix A, “Infoprint Server jobs and job logs” on page 117, provides information to help you understand the different types of jobs used by Infoprint Server and to find different messages in the system job logs.
- Appendix B, “PDF conversion completion data queue” on page 125, provides additional information about the PDF data queue that is not available in the current V5R2 manuals.
- Appendix C, “Using screen prints to test printer file parameters” on page 129, provides a technique to create test spooled files quickly and easily.
- Appendix D, “Spooled file types for PDF conversion” on page 133, summarizes the different spooled file types available on the iSeries, and describes how each is handled by Infoprint Server and intelligent routing.
- Appendix E, “PDF mapping object versus PDF mapping program comparison” on page 137, compares the features available when you are using a PDF mapping object versus a PDF mapping program to implement intelligent routing.
- Appendix F, “Version 5 enhancements to Infoprint Server” on page 139, describes the enhancements made to Infoprint Server with Version 5 Release 2, additional PTFs, and Version 5 Release 3.
- Appendix G, “Infoprint Server performance” on page 145, provides information about times and system utilization for the conversion of different types of spooled files to PDF.
- Appendix H, “Sample programs” on page 163, contains sample program listings that are referenced in other chapters.

2

Planning for intelligent routing

This chapter looks at the following topics relating to the planning and system setup of intelligent routing:

- Configuring for intelligent routing
- Understanding mail sender, reply-to, and Portable Document Format (PDF) administrator
- Building a complex intelligent routing application

2.1 Configuring for intelligent routing

Along with creating the appropriate PDF mapping object or PDF mapping program, you must perform the following additional configuration tasks:

- Create a Print Services Facility (PSF) configuration object
- Create a device description
- Enrolling users in the system directory
- Creating Simple Mail Transfer Protocol (SMTP) addresses for users
- Configuring the iSeries server as an SMTP and Post Office Protocol (POP) server

2.1.1 Creating a PSF configuration object

The following examples show how to create PSF configuration objects that are used for intelligent routing. In all cases, we assume that intelligent routing is used to direct segments of a spooled file to different types of destinations. The Advanced Function Presentation (AFP) respool and the PDF administrator functions are used. The examples include versions for using a PDF mapping object or PDF mapping program with both V5R2 and V5R3.

We recommend that you type the Create PSF Configuration (CRTPSFCFG) command and press F4 for the fully prompted menus. Press F11 to see the keywords associated with each parameter.

Example 2-1 shows the V5R2 command to create a PSF configuration object that uses a PDF mapping object.

Example 2-1 Create PSF configuration object to use a PDF mapping object (V5R2)

```
CRTPSFCFG PSFCFG(pdfmapobj2)
          PDFGEN(*MAIL)
          PDFDEVTYPE(*IP4O3OO)
          PDFPPRDWR1(*LETTER)
          PDFPPRDWR2(*LEGAL)
          PDFMULT(*YES *SPLIT)
          PDFINCFNT(*NO)
          PDFDTAQ(mylib/mypdfdtaq)
          PDFMAILSVR(SMTP mail server)
          PDFSENDER(pdf sender)
          PDFMAPPGM(QSYS/QPQMAPEXIT)
          PSFDFNOPT('PDFMAP(mylib/mymapobj)' 'AFPSAVE(*YES)' 'AFPOUTQ(mylib/afpoutq)'
             'PDFADMIN(pdfadmin@company.com)')
          TEXT('PSF Config for PDF mapping object, V5R2')
          USEOUTLFNT(*YES)
```

Example 2-2 shows the V5R2 command to create a PSF configuration object that uses a PDF mapping program.

Example 2-2 Create PSF configuration object to use a PDF mapping program (V5R2)

```
CRTPSFCFG PSFCFG(pdfmappgm2)
          PDFGEN(*MAIL)
          PDFDEVTYPE(*IP4O3OO)
          PDFPPRDWR1(*LETTER)
          PDFPPRDWR2(*LEGAL)
          PDFMULT(*YES *SPLIT)
```

```
          PDFINCFNT(*NO)
          PDFDTAQ(mylib/mypdfdtaq)
          PDFMAILSVR(SMTP mail server)
          PDFSENDER(pdf sender)
          PDFMAPPGM(mylib/mymappgm)
          PSFDFNOPT('AFPSAVE(*YES)' 'AFPOUTQ(mylib/afpoutq)'
             'PDFADMIN(pdfadmin@company.com)')
          TEXT('PSF Config for user PDF mapping program, V5R2')
          USEOUTLFNT(*YES)
```

Example 2-3 shows the V5R3 command to create a PSF configuration object that uses a PDF mapping object.

Example 2-3 Create PSF configuration object to use a PDF mapping object (V5R3)

```
CRTPSFCFG PSFCFG(pdfmapobj3)
          PDFGEN(*MAIL *SPLF *STMF)
          PDFDEVTYPE(*IP4O3OO)
          PDFPPRDWR1(*LETTER)
          PDFPPRDWR2(*LEGAL)
          PDFMULT(*YES *SPLIT)
          PDFINCFNT(*NO)
          PDFDTAQ(mylib/mypdfdtaq)
          PDFMAILSVR(SMTP mail server))
          PDFSENDER(pdf sender)
          PDFADMIN(pdfadmin@company.com)
          PDFMAPPGM(*IBMPGM)
          PDFMAP(mylib/mymapobj)
          PDFOUTQ(mylib/pdfoutq)
          PDFDIR('/mydir')
          AFPSAVE(*YES)
          AFPOUTQ(mylib/afpoutq)
          TEXT(PSF Config for PDF mapping object, V5R3)
          USEOUTLFNT(*YES)
```

Example 2-4 shows the V5R3 command to create a PSF configuration object that uses the PDF mapping program.

Example 2-4 Create PSF configuration object to use a PDF mapping program (V5R3)

```
CRTPSFCFG PSFCFG(pdfmappgm3)
          PDFGEN(*MAIL *SPLF *STMF)
          PDFDEVTYPE(*IP4O3OO)
          PDFPPRDWR1(*LETTER)
          PDFPPRDWR2(*LEGAL)
          PDFMULT(*YES *SPLIT)
          PDFINCFNT(*NO)
          PDFDTAQ(mylib/mypdfdtaq)
          PDFMAILSVR(SMTP mail server)
          PDFSENDER(pdf sender)
          PDFADMIN(pdfadmin@company.com)
          PDFMAPPGM(mylib/mymappgm)
          PDFOUTQ(mylib/pdfoutq)
          PDFDIR('/mydir')
          AFPSAVE(*YES)
```

```
AFPOUTQ(mylib/afpoutq)
TEXT(PSF Config for user PDF mapping program, V5R3)
USEOUTLFNT(*YES)
```

You can create or modify a PSF configuration object using the AFP Manager options of iSeries Navigator. Figure 2-1 and Figure 2-2 show the windows to create a PSF configuration object for V5R2 that uses a PDF mapping object.

Note: You may require an update to your iSeries Navigator software to use AFP Manager with PSF configuration objects that use the new PSF defined options. A fix is included in iSeries Navigator Service Pack 7. Contact the IBM Support Center for information about how to obtain the latest service pack or an interim fix.

You find most of the parameters used for PDF generation under the PDF Transform tab (Figure 2-1). Some parameters, such as Device emulation type, are common to all types of PDF gnereation. Other prompts are specific to the type of PDF. For example, if you select Send as electronic mail, you are prompted for an Electronic mail sender and Mapping program.

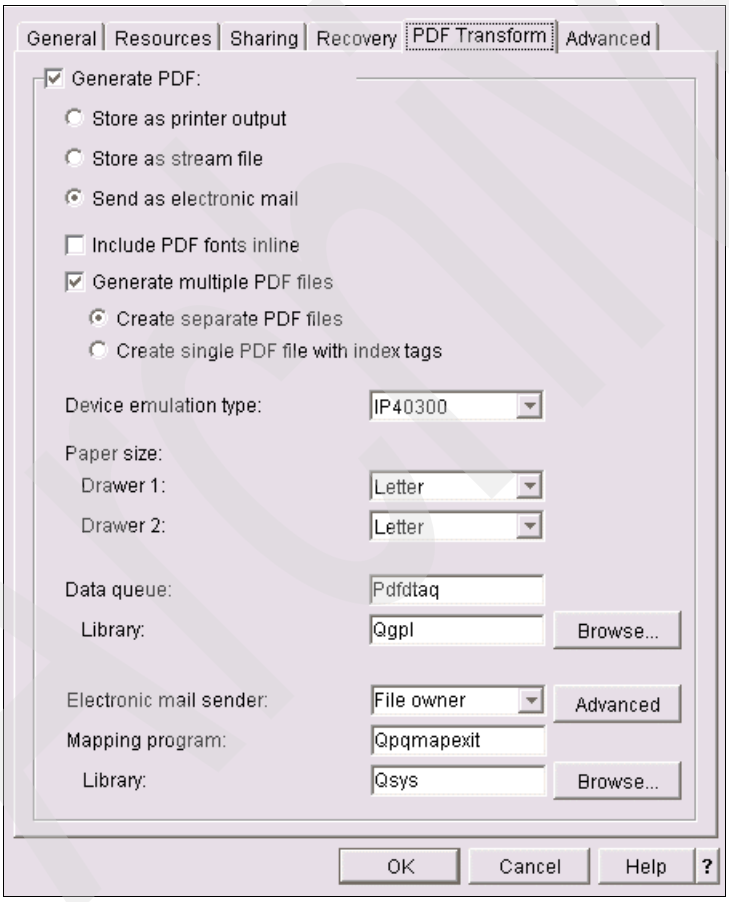


Figure 2-1 PSF configuration using iSeries Navigator: PDF Transform page

In V5R2, you must specify several PDF generation options as PSF defined options. This is done using the Advanced tab (Figure 2-2). Type each option one at a time in the box under PSF defined options, and click Add.

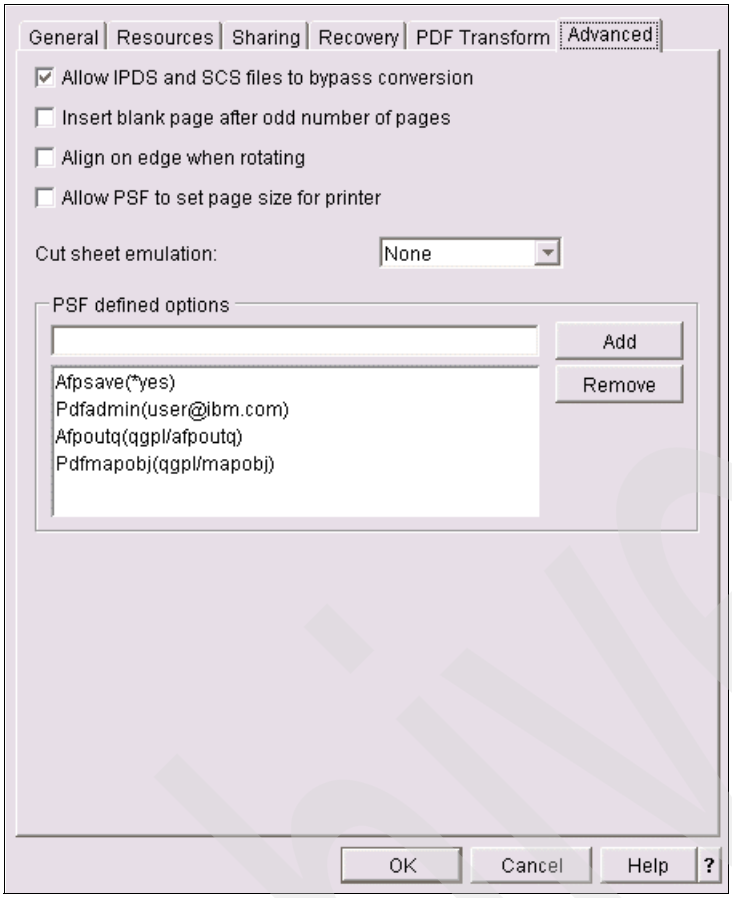


Figure 2-2 PSF configuration using iSeries Navigator: Advanced page

If you want to use outline fonts, select Substitute outline fonts for raster fonts under the Resources tab (not shown).

For a general description of the parameters and options on these windows, see the documentation in *Infoprint Server for iSeries: User's Guide*, G544-5775, or *Infoprint Server for iSeries: User's Guide*, G544-5775. Or refer to the program temporary fixes (PTFs) in *Using Intelligent Routing* or *Using Mapping Objects*.

In addition to the information in these sources, consider these points:

- Be aware of the following limitation for V5R2. Some of the options of the PSF configuration object are only available if the corresponding entry is made for PDFGEN:
  - PDFDIR is only available for PDFGEN(*STMF).
  - PDFOUTQ is only available for PDFGEN(*SPLF).
  - PDFMAILSVR, PDFMAPPGM, and PDFSENDER are only available for PDFGEN(*MAIL) when you are using F4 to prompt to enter the command.

    If you enter the CRTPSFCFG or CHGPSFCFG commands on the command line, you can enter PDFMAPPGM and PDFSENDER even if PDFGEN is not set to *MAIL.

  If you use the command line to enter parameters, you do not see an error if you enter parameters that are not supported for the given value of PDFGEN, but the unsupported values are lost. In the case of PDFMAILSVR, it reverts to the default value of *SNDDST.

This may become an issue if you:

- Expect to use a PDF mapping program or PDF mapping object for e-mail using an SMTP server and want the option to store the PDF stream files using the default naming convention
- Use a default output queue for PDF spooled files

If you use an SMTP server for e-mail, you must specify the stream file path and PDF output queue explicitly in the PDF mapping program or PDF mapping object.

The only allowable combination is that you can send e-mails using *SNDDST and have one PDF spooled file or PDF stream files placed in the default locations. These are the steps to create the PSF configuration object for this function:

a. Create a PSF Configuration object with `PDFGEN(*MAIL)` and specify all the values that are destination independent, such as PDFINCFNT and PDFMULT, along with the PDFSENDER and PDFMAPPGM. If a PDF mapping object is used, specify the PDFMAP parameter using PSFDFNOPT. See Example 2-1 on page 12 or Example 2-2 on page 12. The only difference from these samples is that you can leave the PDFMAILSVR parameter set to *SNDDST.

b. Choose one of the following options:

   - To use the default naming convention for stream files, use the Change PSF Configuration (CHGPSFCFG) command to set the default directory name:

     ```
     CHGPSFCFG mypsfcf PDFGEN(*STMF) PDFDIR('/mydir')
     ```

   - To use a default output queue for PDF spooled files, use the CHGPSFCFG command to set the default output queue name:

     ```
     CHGPSFCFG mypsfcf PDFGEN(*SPLF) PDFOUTQ(mylib/pdfoutq)
     ```

c. The PSF configuration object remembers the values that you specified for PDFSENDER and PDFMAPPGM.

- On V5R3, you can specify multiple values for PDFGEN. This is not necessary if you use a PDF mapping program or PDF mapping object to specify the disposition of the spooled files or segments. The only requirement is that you specify anything other than PDFGEN(*NONE).

  If you are using F4 to prompt while entering the CFRTPSFCFG command, you are prompted for all disposition-related parameters, such as PDFSENDER, PDFMAILSVR, PDFOUTQ, PDFDIR, and AFPOUTQ, for any value of PDFGEN other than *NONE.

  Consequently, you can use a default output queue and a default directory for stream files in your PDF mapping program or PDF mapping object, even if you choose to use an SMTP mail server.

- **PDFMAILSVR:** To use an external SMTP mail server, you can enter the IP address or the domain name of the server.

- **PDFDEVTYPE:** Although the default value for PDF device type is *IP40240, IBM recommends that you use *IP40300 to use higher resolution fonts.

- **PDFPPRDWR1 and PDFPPRDWR2:** Infoprint Server does not look at the PAGESIZE setting in a printer file. It relies on the values specified for PDF paper size for drawer1 and paper size for drawer 2, which it uses in combination with the paper drawer selection in the printer file, DDS, or form definition.

  Only one drawer value can be used in a given spooled file. You cannot change drawers using the DRAWER keyword in DDS, or by changing BINs in a form definition. If a spooled file selects more than one paper source, it is held.

There is no way to override these values in the PDF mapping program or PDF mapping object. The values specified in the PSF configuration object are used for all documents that are processed by all devices that use it. Think of Infoprint Server's PDF generator as being a printer that has two paper drawers, whose sizes are permanently fixed. If you need to generate PDF files using additional paper sizes, you need to create separate PSF configuration object or device description combinations.

If the paper size specified in the PSF configuration object is smaller than the application page, you may lose data. No message is issued.

- **PDFINCFNT:** To reduce the size of the generated PDF files, specify *NO for PDF fonts inline. This tells Infoprint Server not to embed PDF fonts within the document. When the document is viewed, Acrobat Reader maps the core IBM font names to the equivalent Adobe or client environment set of core fonts. For any font name character strings to which Adobe Acrobat does not have an equivalent, Adobe Acrobat uses the Adobe multi-master font substitution program to select the available font that constitutes the "best fit".

  PDF mapping provides acceptable results for fonts from the IBM AFP Font Collection, Infoprint Designer fonts, or fonts from the IBM Infoprint Fonts for Multiplatforms for characters used by the Latin1 language groups. If your documents use any custom fonts or characters from other language groups, you must select PDFINCFNT(*YES) to ensure that Infoprint Server creates an accurate reproduction of the original document.

- **PDFDTAQ:** The PDF data queue parameter is optional. If you name a data queue, it must exist and have a minimum length of 752 characters. For more information about using a the PDF data queue, refer to *Infoprint Server for iSeries: User's Guide*, G544-5775.

- The example PSF configuration objects assume that you are using an SMTP mail server to take advantage of the ability to specify carbon copy (CC) and blind carbon copy (BCC) recipients, a reply-to address, or to add additional stream files to use as body files or attachments.

- **PDFSENDER:** The PDF Mail sender defaults to *SPLFOWN, spooled file owner. Give some thought to using one specific user ID for this function.

  If you specify a value for MAILSENDER in the USRDFNDTA of the printer file or spooled file, that name is used instead of the name specified in the PSF configuration object.

  If you are using a PDF mapping object, you can override the value specified in the PSF configuration object by specifying a value for Mail sender when you are working with the panels for PDF Map Action, Mail = *YES.

  There is no option to override the mail sender if you are using a PDF mapping program.

- **PDFMAPPGM:** If you are using a PDF mapping object, the PDF mapping program name must be set to QSYS/QPQMAPEXIT for V5R2 and *IBMPGM for V5R3.

  If you are using a custom PDF mapping program, the program must exist at the time you create the PSF configuration object.

- **PSFDFNOPT:** In V5R2, the new PSF configuration parameters that are being added using the PTFs must be specified using PSF Defined options. These include:
  - PDFMAPOBJ
  - AFPSAVE
  - AFPOUTQ
  - PDFADMIN

  Each PSF defined option is limited to 30 characters. Because the word PDFADMIN plus the two parentheses takes up 10 characters, this leaves a maximum of 20 characters for the address for the PDF administrator for V5R2.

In V5R3, these same options are specified as standard parameters in the Create or Change PSF configuration object commands. The PDFADMIN field on V5R3 can contain up to 132 characters.

- **PDFMAP:** If you specify a PDF mapping object, it must exist at the time you create the PDF configuration object.
- **AFPSAVE:** If you plan to respool any spooled files or segments as AFP spooled files, you must enable the respool function by specifying AFPSAVE(*YES) in the PSF configuration object. The decision to respool a particular file or segment is made based on settings in the PDF mapping program or PDF mapping object.
- **AFPOUTQ:** If you specify *PSFCFG for the name of the output queue for AFP spooled files in either the PDF mapping program or PDF mapping object, the value specified by the AFPOUTQ parameter in the PSF configuration object is used. Otherwise it is ignored.
- **PDFADMIN:** The PDFADMIN parameter is used to specify an e-mail address to which messages are sent by the PDF subsystem if certain types of errors are encountered. You can also force a message to be sent by specifying a value of 1 for the PDF error disposition value in the PDF mapping program.
- **USEOUTLFNT:** If the spooled file selects an outline font, either directly or via substitution, the PDF uses Adobe Type-1 fonts. If raster fonts are selected, the PDF uses Type-3 fonts. The quality of the PDF as viewed on the target workstation is somewhat better for Type-1 fonts, depending on which version of Adobe reader the client is using. If you select *YES for Use outline fonts in the PSF configuration object, PSF attempts to use outline fonts instead of raster fonts wherever possible.

  Not all requests for raster fonts can be converted to the equivalent outline fonts. For best results, if you are using host font resources, use outline fonts explicitly in your application by selecting coded fonts whose names begin with XZ or character sets whose names begin with CZ.

  The selection of Type-1 versus Type-3 is independent of whether you choose to include the fonts in-line using PDFINCFNT.
- **TEXT:** Use a meaningful description to define the use of this PSF configuration object.

2.1.2 Creating a device description

Set up your device the same way you that set up a TCP/IP IPDS printer. You can have multiple devices active, but each must be configured as a different printer with a different port.

Example 2-5 shows the Create Device Printer command that is used to create a virtual printer for PDF generation and intelligent routing.

Example 2-5 Create a device description for PDF generation

```
CRTDEVPRT DEVD(pdfdevd)
          DEVCLS(*LAN)
          TYPE(*IPDS)
          MODEL(0)
          LANATTACH(*IP)
          AFP(*YES)
          PORT(5013)
          FONT(11)
          RMTLOCNAME('127.0.0.1')
          USRDFNOBJ(mylib/mypsfcfg *PSFCFG)
          TEXT('Convert spooled files to PDF')
```

Consider the following points:

- The device description is the same for either V5R2 or V5R3. It is not affected by the use of a PDF mapping program or PDF mapping object.
- **DEVD and USRDFNOBJ:** Although it is not required, we recommend that you use the same name for the device description and the PSF Configuration object, provided that it also fits in with your company naming conventions.
- The DEVCLS, TYPE, MODEL, LANATTACH, AFP, and RMTLOCNAME parameters must be configured as shown.
- **PORT:** The PORT number must be unique.
- **USRDFNOBJ:** The User-defined object parameter specifies the corresponding PSF configuration object.
- **TEXT:** Use a meaningful description to help define the use of this device description.

2.1.3 Enrolling users in the system directory

Any user designated as a sender for e-mail must be enrolled in the iSeries system directory. If you are using an SMTP mail server, the user must also have an SMTP e-mail address. Finally, for a user to receive incoming e-mails, such as non-delivered mail, the directory entry must have Mail service level set to System message store, and Preferred address set to SMTP name.

There are several ways to specify each of these. You can enter the following commands on a command line or include them in a CL program:

```
ADDDIRE USRID(user system) USRD('e-mail USER') USER(user)
        MSFSRVLVL(*SYSMS) PREFADR(*SMTP)

CHGDIRE USRID(user system) USRDFNFLD((SMTPAUSRID SMTP user) (SMTPDMN SMTP system))
```

In this example, the user's name and domain for external e-mail are the same as the user profile and the iSeries system name. This does not have to be so. This is discussed further in 2.2, "Understanding mail sender, reply-to, and PDF administrator" on page 22.

If the user profile has an SMTP user ID, that user ID is used as the sender, even if you use SNDDST to send the mail. Otherwise, if you use SNDDST to send the mail and there is no SMTP User ID, the profile's User ID is used.

2.1.4 Configuring the iSeries server to send e-mail

To set up your OS/400 to send e-mail with Infoprint Server V5R2 and later, follow these steps:

1. Specify the user ID and address that the mail service uses to route mail that has an Internet address as a recipient.

   This is not the ID of an actual person. For this example, we use INTERNET as the user ID and GATEWAY for the address. We create a directory entry with this user ID and address in step 2. On the command line, enter:

   ```
   CHGDSTA SMTPRTE(INTERNET GATEWAY)
   ```

   Restriction: You need security administrator (*SECADM) authority to use this command.

2. Create a directory entry that mail services uses to route mail that has an Internet address as a recipient.

The User ID and Address must have the same values that you specified for the user ID and address in step 1. On the command line, enter:

```
ADDDIRE USRID(INTERNET GATEWAY) USRD('Generic Internet user') SYSNAME(INTERNET)
PREFADR(NETUSRID *IBM ATCONTXT)
```

3. Specify that outgoing e-mail messages are not split apart and specify that the POP mail servers are started when TCP/IP is started. On the command line, enter:

   ```
   CHGPOPA MSGSPLIT(*NOMAX) AUTOSTART(*YES)
   ```

4. Configure the SMTP server to specify it to start when TCP/IP starts. Also specify the mail router to pass through the firewall, if you have one. If you exchange e-mail with Lotus® Domino® servers, specify *NO for Percent routing character (PCTRTGCHR) along with the following parameters. On the command line, enter:

   ```
   CHGSMTPA AUTOSTART(*YES) MAILROUTER('mailrouter') FIREWALL(*YES)
   ```

   A value for MAILROUTER is required only if you want to control what happens when a name and address cannot be resolved. The mail router name is typically be set up by the system administrator. Specify MAILROUTER(*NONE) if you are unsure.

   Specify a value other than *NONE for ALYRLY if you want to use the local SMTP server.

   To provide for outgoing mail, yet to protect this server from being an open relay, you must prevent it from relaying any e-mail except that which originates on itself (127.0.0.1) with the following commands:

   ```
   CHGSMTPA   ALWRLY(*LIST)
   ADDSMTPLE  TYPE(*ACCEPT) INTNETADR('127.0.0.1')
   ```

5. Create user profiles for any users who want to send e-mail. We create user JOEJ. Enter:

   ```
   CRTUSRPRF USRPRF(JOEJ) PASSWORD(not2bad) TEXT('Joe Johnson')
   ```

6. Add users to the system distribution directory. The value for USER must be the same as the value for USRPRF specified in step 5. To use this ID to send e-mail, specify this user ID in the PDFSENDER parameter of the PSF configuration object used with the PDF subsystem. Alternatively, you can specify it in USRDFNDTA in the spooled file's attributes or the printer file. ISERIES is the system name of the iSeries that Joe works on. Enter:

   ```
   ADDDIRE USRID(JOEJ ISERIES) USRD('Joe Johnson') USER(JOEJ)
   ```

   Important: Complete the following step only if you want to use an SMTP server to send e-mail with Infoprint Server. This step requires *SECADM special authority.

7. Enter the following command to add SMTP addresses for the users in the system distribution directory:

   ```
   CHGDIRE USRID(name address) USRDFNFLD((SMTPAUSRID SMTP 'user-id') (SMTPDMN SMTP
   'smtp-address'))
   ```

   For example, use this command to add an SMTP address for the user ID JOEJ, who's entry specifies the SNADS address as ISERIES. His Internet e-mail address is JOEJ@SUPERSUNSEEDS.COM:

   ```
   CHGDIRE USRID(JOEJ ISERIES) USRDFNFLD((SMTPAUSRID SMTP 'JOEJ') (SMTPDMN SMTP
   'SUPERSUNSEEDS.COM'))
   ```

   Alternatively, follow these steps to add SMTP addresses for the users in the system distribution directory:

   a. To add an SMTP address for the user ID JOEJ, enter the `WRKDIRE` command and press Enter.

   b. Choose option `2` (Change) by the user ID you want to change and press Enter.

c. Press F19 (Add name for SMTP). If no SMTP address is associated with this ID, a message appears on the bottom of the display. Press Enter.

d. Specify either the SMTP user ID and domain or the SMTP route. In Figure 2-3, we specify the user ID and domain of Joe's Internet e-mail address JOEJ@SUPERSUNSEEDS.COM.

```
                              Add Name for SMTP
                                                          System:   ISERIES
 Type choices, press Enter.

   User ID . . . . . .  .   JOEJ       Character value, *ANY, F4 for list
   Address . . . . . .  .   ISERIES    Character value, F4 for list

   SMTP user ID  . . . . .  JOEJ
   SMTP domain . . . . . .  SUPERSUNSEEDS.COM


   SMTP route  . . . . . .

 F3=Exit   F4=Prompt   F12=Cancel
```

Figure 2-3 Adding a name for SMTP

8. Stop the TCP/IP server:

   ```
   ENDTCPSVR SERVER(*SMTP)
   ```

9. Start the TCP/IP server:

   ```
   STRTCPSVR SERVER(*SMTP)
   ```

10. You can now test the new e-mail setup by using the SNDDST command. We recommend that you do this before you test Infoprint Server for the first time.

    Test the e-mail setup by entering this command. You can use the TOUSRID or TOINTNET parameters or both:

    ```
    SNDDST TYPE(*LMSG) TOINTNET((me@mysystem.com)) DSTD('E-mail') LONGMSG('This is a test to
    determine whether my new e-mail setup works.') SUBJECT('E-mail test')
    ```

11. Go to your Internet mail system or OS/400 to see if you received the message.

    For instructions about using SNADS to receive e-mail on your OS/400, refer to the iSeries Information Center at:

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

    From the Information Center, select **Networking ->TCP/IP -> E-mail -> Send and receive e-mail on the iSeries -> Use SNADS to receive e-mail**.

For more information about these commands, refer to the OS/400 online help or the iSeries Information Center.

2.1.5 Setting up OS/400 as a POP e-mail client

You must set up your OS/400 as a Post Office Protocol (POP) e-mail client if you want to:

- Receive e-mail
- Receive error messages when your mail cannot be delivered by another server
- Use your OS/400 as an SMTP server to send e-mail

For instructions, refer to the iSeries Information Center at:

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

To find this topic, in the Information Center, select **Networking -> TCP/IP -> E-Mail -> Send and Receive E-mail on iSeries -> Set up POP e-mail clients**.

For an example of setting up a Netscape Communicator client, see *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250.

2.2 Understanding mail sender, reply-to, and PDF administrator

Depending on circumstances, and how your e-mail function is configured, each of the following users can receive incoming e-mail:

- The user designated as the PDF mail sender
- The user specified using a reply-to address
- The user specified as the PDF administrator

2.2.1 PDF mail sender

The PDF mail sender can be specified in any or all of the following places:

- The *MAIL action in a PDF mapping object
- On the User-defined data parameter of a spooled file attributes
- The PDFSENDER parameter of the PSF Configuration Object

There is no way to change this value in the exit program. The order of precedence is a PDF mapping object, over spooled file attributes, over PSF configuration object.

If you use *SNDDST as the PDFMAILSVR, the user ID needs to be enrolled in the system directory, but an SMTP address does not need to be specified. If you use any other value for PDFMAILSVR, the sender must have an SMTP address. You can add an SMTP address for a user by entering one of the following commands:

- The WRKNAMSMTP command
- The WRKDIRE command for the user (then press F19)
- The CHGDIRE command as shown here:

```
CHGDIRE USRID(user system_name) USRDFNFLD((SMTPAUSRID SMTP email_user) (SMTPDMN SMTP
email_domian))
```

If an SMTP address is not specified, the mail is sent as being from SENDER@Domain, where Domain is the DMNNAME specified by the CHGTCPDMN command. If an SMTP address is specified for the SENDER, the SMTP address is used.

The SMTP address that you specify for the SENDER is used in these cases:

- If a value for Reply-to address is not set by using a PDF mapping program or PDF mapping object, and the recipient of an e-mail clicks the Reply button on their mail program, the SENDER's SMTP address is the destination of the new reply.
- If the outgoing mail was addressed incorrectly, the non-delivery message is sent to the SENDER's SMTP address. This assumes that the address was of the correct format, but that the user name or the domain name was specified incorrectly. For example, instead of sending mail to doraj@improved.ca, you tried to send it to djackson@improved.ca.

The SMTP address for the user does not have to be an iSeries address. If your organization normally uses a different server for e-mail, you can use an address on that server.

If the SMTP address of the PDF mail sender is an address on your iSeries, you need to set up the iSeries as a POP server to receive replies and non-delivery messages. This is discussed in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, and in *Infoprint Server for iSeries: User's Guide*, G544-5775. The user's directory entry must be set up for Mail Service Level: System Message Store and Preferred Address: SMTP name.

In the sample e-mail shown in Figure 6-5 on page 87, the PDF sender was user MIRA. MIRA's SMTP address is specified as mira@SUPERSUNSEEDS.COM.

2.2.2 Reply-to address

You can only specify a reply-to address if you are using a PDF mapping program or a PDF mapping object, and only if the mail server uses SMTP. If you set a value for Reply-to, it is used if the recipient of an e-mail clicks the Reply button on their mail program. Then the new reply is sent to the address designated by the reply-to address.

If the reply-to address is an address on your iSeries, you need to set up the iSeries as a POP server to receive replies and non-delivery messages. This is discussed in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, and in *Infoprint Server for iSeries: User's Guide*, G544-5775. The user's directory entry must be set up for Mail Service Level: System Message Store and Preferred Address: SMTP name.

In the sample PDF mapping program used to send the e-mail illustrated in Figure 6-5 on page 87, the reply-to address was defined as fred@supersunseeds.com. Lotus Notes® only displays the user portion of the address "fred". Other e-mail clients display the full address. In either case, if the recipient chooses to reply, the e-mail is sent to fred@supersunseeds.com.

2.2.3 PDF administrator

You have the option to specify an e-mail address to use as a PDF administrator. We recommend that you use this to assist with any problem determination when using Infoprint Server. If Infoprint Server cannot deliver a document or segment to the designated destinations, the spooled file is held, a message is logged, and a copy of the PDF file is sent as an e-mail to the PDF Administrator.

Configuring for PDF administrator (V5R2)

In V5R2, you specify the e-mail address of the PDF administrator using the PSF-defined options of the PSF configuration object. For example, to change an existing PSF configuration object to add the PDF administrator function, enter the following command:

```
CHGPSFCFG PSFCFG(mylib/mypsfcfg) PSFDFNOPT('PDFADMIN(pdfadmin@company.com)')
```

This example assumes that no other options are specified using PSF-defined options. If PDFMAP, AFPSAVE, or AFPOUTQ is specified, you need to re-enter them.

Configuring for PDF administrator (V5R3)

In V5R3, you specify the e-mail address of the PDF administrator using the PDFADMIN parameter of the PSF configuration object. For example, to change an existing PSF configuration object to add the PDF administrator function, enter the following command:

```
CHGPSFCFG PSFCFG(mylib/mypsfcfg) PDFADMIN(pdfadmin@company.com)
```

Conditions that trigger an e-mail being sent to the PDF administrator

Some of conditions that cause an e-mail to be sent to the PDF Administrator include:

- The address is in an invalid format, such as joe#ibm.com.
- QSPLJOB does not have the correct authority to create stream files in the designated directory path.
- The spooled file or segment does not match any entries in the PDF mapping object.
- The programmer turns on the Disposition of PDF Error flag in the PDF mapping program.

Description of the error e-mail

The e-mails sent to the PDF administrator are sent with the Sender set to QSPLJOB. The PDF attachment is not encrypted, even if the original e-mail request included encryption. The default PDF file name is used. See the example in Figure 6-4 on page 81 for a sample of an error e-mail that is sent to the PDF administrator. In that example, the spooled file name is 276034_000003_OUTPUTMRG_12102003_000004.PDF.

Note the following explanation for this name:

- *276034* is the iSeries job number.
- *000003* is the spooled file number within that job.
- *OUTPUTMRG* is the spooled file name.
- *12102003* is a date stamp.
- *000004* is a sequence number. This is the fourth segment of a file that was split into multiple separate PDF files.

This information can help the administrator evaluate the cause of the error. You can find additional information that may be useful in problem determination in the PDF data queue, and in the job logs for the PDJ and WTR jobs. See Appendix A, “Infoprint Server jobs and job logs” on page 117, and Appendix B, “PDF conversion completion data queue” on page 125, for more information about these topics.

For more information about the PDF administrator function, see PTF *Using Intelligent Routing*.

2.3 Building a complex intelligent routing application

Due to the inherent complexity of an Infoprint Server implementation, we recommend that the developer segment the tasks involved and test each one before proceeding to the next. Consider the following test case.

Fred at Super Sun Seeds wants to modify the customer invoicing process. Today, invoices are printed from one large spooled file and mailed using traditional methods. Fred wants to take advantage of intelligent routing to send some invoice files as e-mails and some as faxes. Some of his clients have requested that they still receive a hard copy.

Fred has chosen the PDF mapping program method to have more flexibility in customizing the message in the e-mail. This method allows him to use data that is in his existing customer master database.

A recommended plan for development includes the following steps. You can find details for each step in other sections of this publication or in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250.

1. Install the Infoprint Server for iSeries software and the latest PTFs for Infoprint Server for iSeries and Print Services Facility for iSeries.

2. Create a PSF configuration object and device description that converts spooled files to single PDF stream files in a given directory. At this stage, do not split the spooled file or use a PDF mapping program. Test this by using a short version (or use page range) of the sample invoice. This test does two things for Fred:

   a. It verifies that simple PDF generation works.

   b. It allows Fred to see what the PDF looks like. If Fred is not happy with how the page looks, he can go back to the application and change fonts, positions, overlays, etc.

3. Set up a PSF configuration object and device description that generates the PDF files as e-mails. Again, keep it simple. Do not use an exit program and do not split the spooled file. Using any simple spooled file, use the USRDFNOPT parameter to define a MAILTAG with an e-mail address.

   At this stage, Fred also needs to perform the necessary steps to configure for e-mail:

   a. Enroll QSPLJOB and the intended SENDER of the e-mails in the System Directory.

   b. Follow the steps in 2.1, “Configuring for intelligent routing” on page 12, to configure the iSeries for e-mail.

   We recommend that you test the e-mail first by using the default mail server. Select PDFMAILSVR(*SNDDST) in the PSF configuration object. When it is working, create the appropriate SNMP addresses and switch to PDFMAILSVR(*LOCAL) or use an external SNMP server.

4. Fred is now ready to segment the spooled file.

   a. He uses CRTAFPDTA to create an Advanced Function Presentation data stream (AFPDS) merged stream file, which includes the group tags.

   b. PRTAFPDTA is used to create a new spooled file using the stream file.

   c. Fred can verify the content of the mail tags in one of two ways:

      - He can use the iSeries Navigator to view the spooled file using the AFP viewer.
      - He can create a PSF configuration object and device description that uses the PDFMULT(*YES *INDEX) in the PSF configuration object. The resulting bookmarks correspond to the mail tags.

   d. Generate stream files using PDFMULT(*YES *SPLIT) to demonstrate that the spooled files is segmented correctly.

5. Fred is ready to plan for the exit program. We recommend that he use the spreadsheet described in 6.6, “Using a spreadsheet to calculate length and offset values” on page 72, to calculate the offset values for the formats needed for the output data structure.

6. In the first version of the exit program, Fred hard codes each of the desired actions, independent of the results of the lookup program. This helps him verify that his output data structure is designed correctly.

7. Fred modifies the program to include the logic to deal with each invoice segment according to the customer preference fields.

8. To use Facsimile support, Fred creates a monitor program that reads the USRDFNDTA and issues the appropriate SBMFAX command. This is described in Chapter 7, “Using intelligent routing with Facsimile Support/400” on page 99.

9. During testing, Fred entered many of the commands interactively. He now builds a robust business application that automates many of the steps. It also handles various error conditions and provides for appropriate logging.

10. Finally, the job is not complete until the paper work is done. Fred documents his solution so that others at Super Sun Seeds can understand the application and deal with problems or make enhancements in his absence.

3

Using a PDF mapping object

This chapter describes how to use a Portable Document Format (PDF) mapping object to route documents or document segments to any combination of four possible destinations:

- Sending a PDF as an e-mail
- Creating a PDF spooled file
- Creating an Advanced Function Presentation (AFP) spooled file
- Storing the PDF as a stream file in the integrated file system (IFS)

The documentation in this chapter is intended to supplement the information in the product documentation. It assumes that you are familiar with PDF creation and mapping object panes as described in the following manuals:

- For V5R2:
  - *Using Mapping Objects*. This document is loaded on your iSeries as a PDF file when you install the new program temporary fixes (PTFs) that provide the mapping object function. You can find it in the integrated file system as \QIBM\ProdData\InfoprintServer\Transforms\Mapping_Object.pdf.

    Updates to *Using Mapping Objects* are placed on the IBM Printing Systems Division Web site. You can find current versions of this document at:

    http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipserveruser_i_ww
  - *Infoprint Server for iSeries: User's Guide*, G544-5775
- For V5R3, see *Infoprint Server for iSeries: User's Guide*, G544-5775

Important: The functions described in this document are delivered as PTFs for V5R2 for Infoprint Server for iSeries and Print Services Facility for OS/400. For a current list of PTFs, see iSeries Software Knowledge Base document 23381552 at:

http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase/

For further assistance about obtaining PTFs, contact your local IBM Support Center.

3.1 When to use a PDF mapping object

The PDF mapping object opens the door to users who want more freedom in customizing the creation of their PDF and e-mails when using Infoprint Server. Prior to the availability of this feature, users were forced to write a mapping program if they wanted to enhance the e-mail process in several ways. With a PDF mapping object, the user can fill in panels on the iSeries interactively to specify:

- Map a routing tag to a valid e-mail address
- Add reply-to, CC, or BCC addresses to the e-mail
- Assign a custom name to the PDF attachment
- Add additional text and Hypertext Markup Language (HTML) files to the body of an e-mail
- Add additional attachments to the body of an e-mail
- Change the subject line of an e-mail
- Change the default body text of an e-mail
- Change the directory path and name of PDF files that are stored in the IFS
- Change the default authorizations of PDF files that are stored in the IFS
- Change the user data, user-defined data, or form type of PDF or AFP spool generated by Infoprint Server

There are certain situations in which it is still best to use a PDF mapping program. However, using a PDF mapping object is best in other situations. It is important to distinguish which is appropriate for a given set of circumstances. The following scenarios indicate where an PDF mapping object provides the necessary functionality:

- Company X recently purchased Infoprint Server to convert various internal documents to PDF format. Some are placed in the iSeries IFS for viewing and others are sent directly as e-mail attachments to the specific recipients. In this case, there is a limited number of types of documents and they do not want to do any programming. The mapping object's selection criteria gives Company X all the flexibility they need to handle the various types of business documents in the appropriate way.
- Company Y wants to store copies of their invoices as stream files in PDF format in the integrated file system. This is accomplished by having a program that puts the invoice number and intended path into the routing tag using DDS keywords. With that, they build a PDF mapping object that has a value of *SPLF for the path for PDF stream files. They are also changing the authority of the generated PDF files to allow all users to read the files, but not update them.

For a comparison between the PDF mapping program and PDF mapping object, see Table E-1 on page 137.

3.2 Mapping table

The basic concept behind using a mapping table is quite simple. A user enters a series of selection criteria into the object and a set of actions that are to take place if the spooled file or segment that meets those criteria. The selection criteria can apply to the entire spooled file, or a more granular selection can be made based on the routing tags in individual segments.

The actions can be any of the following combinations:

- Convert the spooled file or segment to PDF and send it as an e-mail attachment.
- Convert the spooled file or segment to PDF and store it in the IFS.
- Convert the spooled file or segment to PDF and spool it to an output queue.
- Convert the spooled file or segment to AFP and spool it to an output queue.

3.3 Creating and referencing a mapping object

To create a usable PDF mapping object, you perform these steps:

1. Use the Create PDF Map (CRTPDFMAP) command to create a map object. In V5R2, it is created as an object type *USRIDX with an attribute PDFMAP. In V5R3, it is created as an object type *PDFMAP.
2. Use the Work with PDF Map Entries (WRKPDFMAPE) command to define the selection criteria and desired actions for the spooled files and segments. This is an interactive process and is discussed in detail in 3.4, “Adding entries to the mapping object” on page 30.

As an alternative, you can use the PDF map application programming interfaces (APIs) to create the PDF mapping object and to add, change, or delete the map entries. This is discussed in Chapter 8, “Using APIs to add entries to PDF mapping object” on page 105.

Attention: You cannot update a PDF mapping object while any writers that reference it are started. End those writers before you add or change entries in the mapping object.

To use a mapping object, specify it in the Print Services Facility (PSF) configuration object used for the PDF subsystem. You must enter these parameters:

- The name of the IBM supplied mapping program
- The name of the user-defined mapping object
- Generate PDF output must be set to any value other than *NONE to be prompted for the name of the mapping program

The method to specify those parameters is slightly different for V5R2 and V5R3, as described in the following sections. They describe only those PSF configuration parameters that deal with the mapping object. For a full description of PSF configuration objects, see 2.1.1, “Creating a PSF configuration object” on page 12.

3.3.1 Specifying a mapping object in a PSF configuration object (V5R2)

In V5R2, you must name the IBM-supplied mapping program, QSYS/QPQMAPEXIT, on the PDFMAPPGM parameter exactly as shown. The mapping object is referenced on the PSF defined option, PDFMAP.

This is an example of the command used to change an existing PSF configuration object, PDFMAPOBJ2, to use a new mapping object, MAPOBJ2:

```
CHGPSFCFG PSFCFG(mylib/PDFMAPOBJ2)
   PDFGEN(*MAIL)
   PDFMAPPGM(QSYS/QPQMAPEXIT)
   PSFDFNOPT('PDFMAP(mylib/MAPOBJ2)')
```

This is a subset of the parameters needed to create a PSF configuration object that references a mapping object. For a more complete example, see Example 2-1 on page 12. Note the fact that you must specify PDFGEN(*MAIL), or press F9, to see the PDFMAPPGM parameter if you are using the prompt screens.

3.3.2 Specifying a mapping object in a PSF configuration object (V5R3)

In V5R3, you use a keyword, *IBMPGM, for the PDFMAPPGM to specify that you are using the IBM-supplied mapping program. There is a new parameter to reference the mapping object, PDFMAP.

This is an example of the command used to change an existing PSF Configuration object PDFMAPOBJ3, to use a new mapping object, MAPOBJ3:

```
CHGPSFCFG PSFCFG(mylib/PDFMAPOBJ3)
   PDFGEN(*MAIL *SPLF *STMF)
   PDFMAPPGM(*IBMPGM)
   PDFMAP(mylib/MAPOBJ3)
```

On V5R3, you have the option to include more than one value for PDFGEN. The previous command shows a subset of the parameters needed to create a PSF configuration object that references a mapping object. For a more complete example, see Example 2-3 on page 13. Note the fact that you must specify PDFGEN(*MAIL), or press F9, to see the PDFMAPPGM parameter if you are using the F4 prompt screen.

3.4 Adding entries to the mapping object

Use the Work with PDF Map Entries (WRKPDFMAPE) command to add or change the routing information for your spooled files. This is done in multiple stages:

1. Enter a sequence number to the entry, and specify whether the entry is segmented.
2. Enter the selection criteria for the spooled file or segment.

   If the mapping entry is segmented, the selection of routing tags is entered on a separate panel.
3. Enter the types of actions to be taken on the generated PDF or AFP output.
4. Enter the detailed specifications for each of the action types selected.

Restriction: You must end all writers that reference a PDF mapping object before you make any changes or addition to any entries in that PDF mapping object.

3.4.1 Creating a new entry

To create a new PDF mapping table entry, you start by entering option 1 to add a new entry. Then enter a sequence number and a value for the Segmented field. The other options are further described in the manuals.

Sequence number

Infoprint Server tries to match the spooled file attributes to each mapping entry in order of the sequence number. If a given spooled file or segment matches two different entries, the one with the lower sequence number is used.

We recommend that you number the sequence entry by increments of some number other than one to allow for insertions at a later time.

Restriction: Before you assign sequence numbers, you must do careful planning. Sequence numbers for segmented entries *cannot* be changed by using Option 2 (Change PDF Map Entry). In addition, you cannot copy a segmented entry with the plan of giving the new copy a different sequence number. This point is not mentioned in the standard documentation. However, you can change the sequence number assigned to a non-segmented entry by using Option 2 (Change PDF Map Entry).

Segmented entry

Before you enter any of the selection criteria, you must indicate whether the entry is to be segmented. If you say that an entry is not segmented, you are prompted for one choice of a routing tag to match, which in turn gives you one set of actions for that entry. If entry is segmented, then you can enter multiple routing tags. Each one can have different combinations actions. Think of this field as being used to indicate whether the mapping table entry is further qualified based on different routing tag values.

Either type of spooled file (segmented or non-segmented) can match against either type of PDF mapping table entry. The only restriction is that the PSF configuration object must have PDFMULT(*YES *SPLIT) to have PSF split the spooled file into multiple segments that are each processed according to different entries in the PDF mapping object.

If you specify Segmented = NO, you are prompted to enter one and only one routing tag entry at the same time you enter all the other selection criteria.

If you specify Segmented = YES, do not enter the routing tag on the same panel as the other selection criteria. Instead, you are presented with the Work With PDF Map Segmented Entries panel.

You might say that an entry that has Segmented = NO works the same as an entry that has Segmented = YES and one segment entry. The only exception to this is the fact that you can specify a value of *ALL for the routing tag for a mapping table entry for Segmented = NO. This value is not valid for a segmented entry.

3.4.2 Selection criteria for a PDF map entry

The next step in creating a PDF map entry is to enter the spooled file selection criteria. This is done on the Add PDF Map Entry panel.

The selection criteria is self explanatory. A given spooled file must match all criteria that is specified to be processed by a given entry. For example, consider an entry that specifies a spooled file name of MYFILE and a user of USER1, and a spooled file named MYFILE with a user value of USER2 is encountered. In this case, it is not processed by the actions associated by that mapping table entry.

The values for Output Queue, Spooled file name, Job name, and User are automatically entered in uppercase. These fields allow for a generic name (in the form abc*) that matches the first few characters of the attribute.

The values for User data and Form type are case sensitive. A Form type of LEGAL in the mapping entry does not match a spooled file if its form type is *legal*. These fields do not offer the option to use a generic name. The entire field must match the spooled file attribute value.

The Routing tag is only presented as an option for mapping entries for Segmented=NO. It is also case sensitive and requires an exact match of the full string.

3.4.3 Working with PDF map segmented entries

If the PDF mapping entry is defined with Segmented=YES, you are presented with the work with PDF Map Segmented Entries panel. This is a simple panel that lets you add, change, copy, remove, display, or print entries based on routing tag values.

Tip: If you try to process a spooled file or segment with a map object, but it doesn't match any of the file selection criteria, it is treated as an error. The spooled file is held, and a message is sent to the PDF administrator if one is defined. For more information about the PDF Administrator function, see 2.2.3, "PDF administrator" on page 23.

You can find an example of Creating a PDF Map Entry to Catch Errors in *Using Mapping Objects* and *Infoprint Server for iSeries: User's Guide*, G544-5775. Use this as a basis to create a similar entry in your PDF mapping table. Customize the subject line or message text to indicate that this is a not-found condition. This helps to distinguish this type of error from other messages sent to the PDF administrator.

3.4.4 Defining a PDF map action

After PSF finds a match for spooled file attributes and a routing tag, it uses the information in the Define PDF Map Action For Spooled File, or Change Segment Entry panels, depending on whether the mapping entry is specified as Segmented = NO or YES.

In each case, the entries are similar. It is a matter of specifying *YES or *NO for each of the possible four actions:

- Mail
- PDF spooled file
- AFP spooled file
- Stream file

Based on the responses to the four choices on this panel, the user is presented with panels that are used to define the details of each of the selected actions.

Define Map Action, Mail = *YES

Several panels are used to customize how to create and send an e-mail. The individual fields are described in detail in *Using Mapping Objects* and *Infoprint Server for iSeries: User's Guide*, G544-5775. This section supplements the information in the manual.

SMTP versus SNDDST

If your mail server is set to *SNDDST in your PSF configuration object, some of the e-mail options are ignored. The following options are supported for PDFMAILSVR(*SNDDST) as well as any of the other entries for PDFMAILSVR that use SMTP:

- All of the parameters that relate to encryption
- Mail-sender
- Message text
- PDF file name
- Subject
- To e-mail address and to e-mail stream file

You can only use the remaining parameters if you are using an SMTP mail server. If you specify any of these options and specified PDFMAILSVR(*SNDDST), the spooled file is held:

- Attachments
- BCC e-mail address
- CC e-mail address
- E-mail body
- Reply to e-mail address

Addresses as *FILE

You have the option to enter up to 100 addresses in each of the To e-mail address, CC e-mail address, and BCC e-mail address fields in the PDF mapping object. If you need to enter more than 100 addresses for any of those address types, you must use an external file.

You may choose to use an external file to act as a form of distribution list, even if you do not have as many as 100 names. An advantage is that you can modify it, as an external file, even while the writer associated with the mapping object is still active.

Addresses in an ASCII stream file

Addresses can be entered in an ASCII stream file and stored in the IFS. Use a simple editor, such as Microsoft® Notepad or Microsoft WordPad. Each address can be up to 80 characters long. Addresses must be separated by a carriage control.

Figure 3-1 shows an example of entering addresses into an ASCII file using Microsoft WordPad.

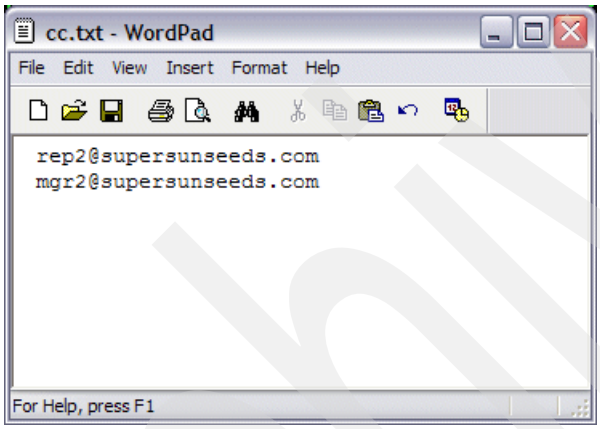


Figure 3-1 Entering addresses in an ASCII file

This file was stored in the folder /mira in the IFS as file cc.txt. It was used in the mapping object to specify the CC addresses, as shown in Example 3-1.

Example 3-1 Using a stream file for addresses

```
...
CC E-MAIL ADDRESSES . . .   *FILE

....
CC E-MAIL STREAM FILE . .   /mira/cc.txt
```

Addresses in an iSeries physical file

You can enter addresses on the iSeries in a source physical file or a data physical file. If you extract the information from other database files, it is likely that you will use a data physical file. If you enter the addresses interactively, consider using a source physical file so that you can use a source entry utility (SEU) to make the updates. In either case, you must reference the file using the qsys.lib naming convention.

In Example 3-2, we use a data physical file for the To e-mail address and a source physical file for the CC addresses.

Example 3-2 Using source and data physical file for addressing

```
TO E-MAIL ADDRESSES . . .   *FILE

TO E-MAIL STREAM FILE . .   /qsys.lib/mira.lib/address.file/address.mbr
...
CC E-MAIL ADDRESSES . . .   *FILE

CC E-MAIL STREAM FILE . .   /qsys.lib/mira.lib/qtxtsrc.file/cc.mbr
```

*Other notes about Action, Mail = *YES*

Consider the following points when adding entries to a PDF mapping object for e-mail:

- If you specify *SPLF for the To e-mail address, Infoprint Server uses the address specified in the routing tag. It is the user's responsibility to ensure that the routing tag is in the proper format for e-mail.
- The subject and message text, if specified, appear exactly as entered. There is no opportunity to build a string based on other spooled file attributes or additional database information. If this is a requirement, you must use a PDF mapping program.
- If you want to use the system-supplied subject or message text, specify *PSFDFT for either of those parameters. The text used for both of these can be found in message PQT4133 in QPQMSGF. Positions 1 to 22 are for the subject line and the remaining characters contain the default message. If you change the contents of this message, it affects all e-mails that use the default.
- You can specify a Mail sender name that is different than the name specified in the PSF Configuration object. This cannot be done if you use a PDF mapping program.

  If you use an SMTP mail server, that is, if PDFMAILSVR(*SNDDST) is not specified, the user specified as Mail sender must have an SMTP user ID.
- Select *PSFDFT for PDF file name to have Infoprint Server generate default names, such as 000001.PDF for your spooled file. If you choose to rename the PDF attachment, you must include the PDF extension. If you do not, the recipient may have difficulties opening the file.

  If you use PDFMAILSVR(*SNDDST), the PDF file name for e-mail must conform to the naming convention used by QDLS. The correct name format is one to eight characters, which can optionally be followed by a period (.) and one to three additional characters. This is not documented correctly in earlier versions of *Using Mapping Objects*.
- When specifying stream files to use as e-mail body files or additional e-mail attachments, you must enter the entire path name for each file. There is no specification in the mapping entry for a default path to use for a group of files such as is available if you use a PDF mapping program.
- There is no option to change the coded character set ID (CCSID) for the subject and message text. If you need to do this, you must use a PDF mapping program.

Figure 3-2 shows an example of an e-mail generated using a PDF.

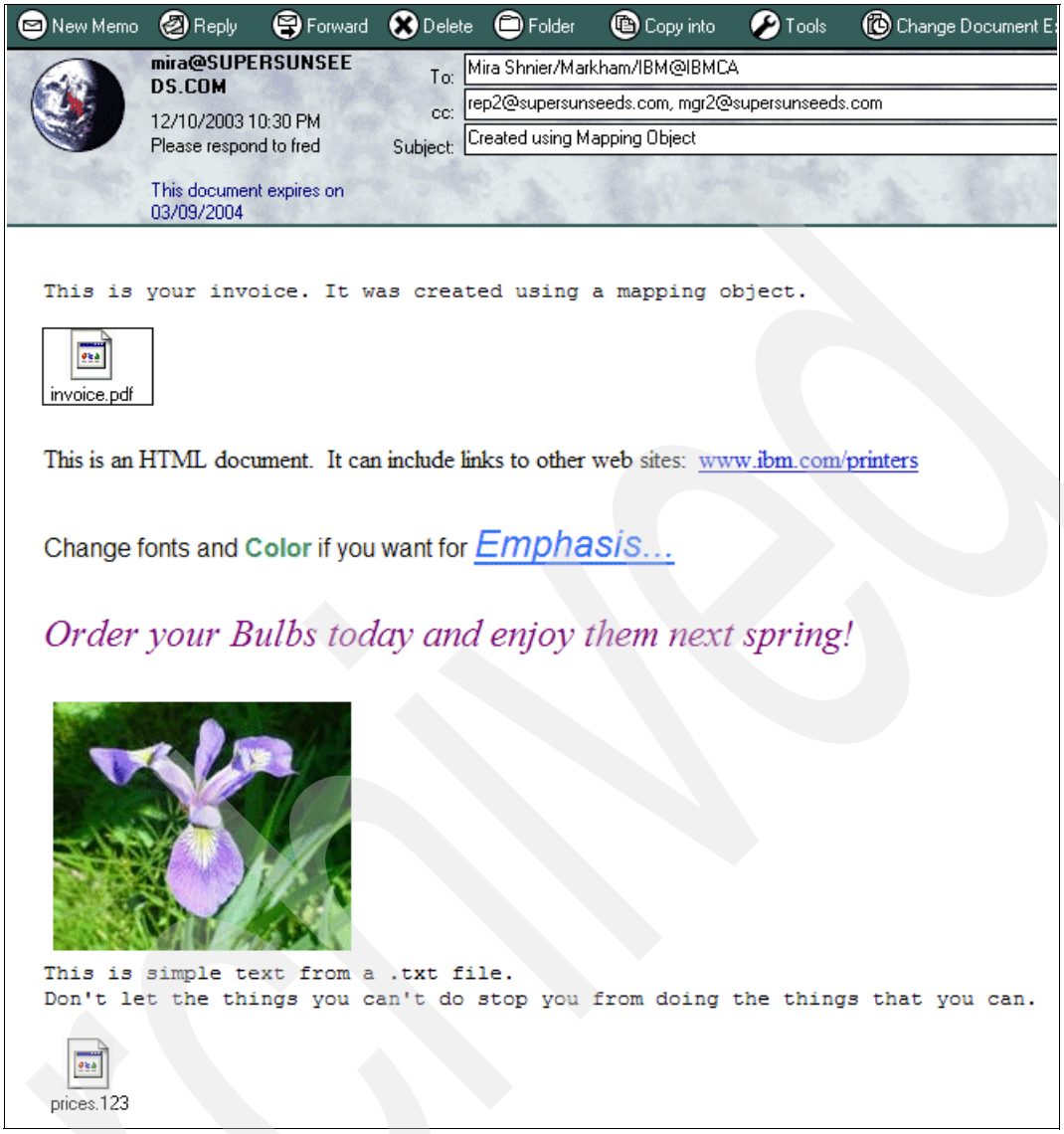


Figure 3-2 E-mail created using a PDF mapping object

Define Map Action, PDF spooled file = *YES

Selecting this option tells Infoprint Server that you want to generate a PDF spooled file from your original spooled file. You can use the given fields to customize the spooled file. Consider the following points when you use this option:

- If you specify *PSFCFG for the name of the output queue, the output queue specified in the PDFOUTQ parameter of the PSF configuration object is used. If no PDFOUTQ is specified, the resulting spooled file is placed in the QGPL/QPRINT output queue.

Restriction: With V5R2, there are limitations to the combination of parameters that you can select when creating or changing a PSF configuration object. Consequently, you can only specify the default output queue name for a PDF spooled file in a mapping object if your e-mail delivery is limited to PDFMAILSVR(*SNDDST). See 2.1.1, "Creating a PSF configuration object" on page 12, for details. Therefore, if you need to use SMTP for your e-mail and choose to use *PSFCFG for the name of the output queue in your PSF mapping object, the PDF output is spooled to QGPL/QPRINT.

This restriction does not exist for V5R3.

- If you specify *SPLF for Spooled file name, User data, Form type or User-defined data, the new spooled file takes on the corresponding value from the original spooled file.
- The new spooled file does not take on the priority from the original spooled file. It uses the default priority of 5.
- The Copies value of the original spooled file attributes is ignored. The new spooled file has copies set to 1.
- If the original spooled file references a form definition, the new spooled file is generated based on its content. If multiple subgroups are in a copy group, multiple PDF pages are generated. If a selection for COPIES is within a subgroup, it is ignored.
- When you enter the output queue and library, spooled file name, user data and form type on the Add or Change AFP Spooled File panel, the values in these fields are automatically converted to upper case. If you want to use a lowercase value for user data or form type, you must enter them in single quotation marks. You can enter the user-defined data in uppercase and lowercase, but you cannot change the format.

Define Map Action, AFP spooled file = *YES

Selecting this option tells Infoprint Server how you want to generate an Advanced Function Presentation data stream (AFPDS) spooled file from your original spooled file. You are given parameters to customize the spooled file. Consider the following points when using this option:

- The PSF configuration object must have AFPSAVE(*YES) specified.
- If you specify *PSFCFG for the name of the output queue, the output queue specified in the AFPOUTQ parameter of the PSF configuration object is used. If no AFPOUTQ is specified, the resulting spooled file is placed in the QGPL/QPRINT output queue.
- If you specify *SPLF for Spooled file name, User data, Form type, or User defined data, the new spooled file takes on the corresponding value from the original spooled file.
- The new spooled file takes on the priority from the original spooled file.
- The Copies value of the original spooled file attributes are ignored. The new spooled file has copies set to 1.
- If the original spooled file references a form definition, the new spooled file references the same form definition. Copies specified within the form definition are honored.
- When you enter the output queue and library, spooled file name, user data, and form type on the Add or Change AFP spooled file panel, the values for these fields are automatically converted to uppercase. If you want to use a lowercase value for user data or form type, you must enter them in single quotation marks. You can enter the user-defined data in uppercase and lower case, but you cannot change the format.

  There is no explicit option for generating FAX. If you want to generate the spooled file or segment as a fax document, refer to the guidelines in Chapter 7, "Using intelligent routing with Facsimile Support/400" on page 99.

Restriction: At the time of writing this Redpaper, the AFP Respool function does not honor page range. It processes the entire spooled file. For example, consider a spooled file that has two segments that are one page long each. Consider that the logic of your PDF mapping object normally selects the first segment to be generated as an AFP spooled file, and the second segment is to be sent as an e-mail only. If you select PAGERANGE(2 2) when you release this spooled file to the PDF subsystem, the AFP spooled segment that corresponds to page one is generated, even though it was not selected as part of the page range.

Define Map Action, Stream file = *YES

There are only two options to set when you select to store a spooled file or segment as a stream file in the IFS:

- PDF stream file name
- Authority

Consider the following points when you specify to store a PDF file as a steam file in the IFS:

- If the directories in the path do not already exist, they are created by Infoprint Server. The owner of any directories created by the transform is QSPLJOB. Their public authority is *RX.
- The PDF stream file object owner is the owner of the original spooled file. The object authority is the value specified on the Authority parameter for the PDF stream file. *EXCLUDE is the default. QSPLJOB has ALLOBJ authority to the object.
- Specify *SPLF for the PDF stream file name to use the routing tag as the PDF file name and path. If this is not a segmented spooled file, Infoprint Server uses the value specified in the MAILTAG option of the user-defined data of the spooled file. If this is a segmented spooled file, Infoprint Server uses the routing tag.

  This is an example of the command to add a stream file name and path to the routing tag for a non-segmented spooled file:

  ```
  CHGSPLFA FILE(FILEME) USRDFNDTA('mailtag(/invoices/1234.pdf)')
  ```

  These are the steps to add a mapping entry that matches spooled file FILEME and creates a PDF file in the path defined in the mail tag:

  a. Add a new PDF map entry for a non-segmented file, as shown in Figure 3-3.

```
                          WORK WITH PDF MAP ENTRIES
PDF MAP . . . :   MAPOBJ                 POSITION TO . .
  LIBRARY . . :     MIRA

TYPE OPTIONS, PRESS ENTER.
  1=ADD   2=CHANGE   3=COPY   4=REMOVE   5=DISPLAY   6=PRINT
  8=WORK WITH SEGMENTS

     SEQUENCE
OPT  NUMBER      SEGMENTED  TEXT
 1   30            *NO
     10            *YES     Map segments
     20            *NO      stream file based on usrdfndta
```

Figure 3-3 Adding a non-segmented entry

b. Define the selection criteria for a spooled file named FILEME, as shown in Figure 3-4.

```
                            ADD PDF MAP ENTRY
SEGMENTED ENTRY . . . . :   *NO
SEQUENCE NUMBER . . . . .   30
TEXT  . . . . . . . . . .   File to *SPLF


TYPE CHOICES, PRESS ENTER.

OUTPUT QUEUE  . . . . . .   *ALL              NAME, GENERIC*, *ALL
  LIBRARY . . . . . . . .                    NAME
SPOOLED FILE  . . . . . .   FILEME            NAME, GENERIC*, *ALL
JOB NAME  . . . . . . . .   *ALL              NAME, GENERIC*, *ALL
USER  . . . . . . . . . .   *ALL              NAME, GENERIC*, *ALL
USER DATA . . . . . . . .   *ALL              CHARACTER VALUE, *ALL
FORM TYPE . . . . . . . .   *ALL              CHARACTER VALUE, *ALL
ROUTING TAG . . . . . . .   *ALL

                                              CHARACTER VALUE, *ALL
```

Figure 3-4 Adding a PDF map entry for spooled file FILEME

c. Specify that the file is to be stored as a PDF stream file, as shown in Figure 3-5.

```
                  DEFINE PDF MAP ACTION FOR SPOOLED FILE
SEQUENCE NUMBER . . . . . :   30
SEGMENTED ENTRY . . . . . :   *NO
TEXT  . . . . . . . . . . :   File to *SPLF



TYPE OPTIONS, PRESS ENTER.

PDF MAP ACTIONS:
  MAIL  . . . . . . . . .   *NO              *YES, *NO
  PDF SPOOLED FILE  . . .   *NO              *YES, *NO
  AFP SPOOLED FILE  . . .   *NO              *YES, *NO
  STREAM FILE . . . . . .   *YES             *YES, *NO
```

*Figure 3-5 Defining a PDF map action for Stream file = *YES*

d. In the Add PDF Stream File panel, enter *SPLF for the name of the stream file, as shown in Figure 3-6.

```
                            ADD PDF STREAM FILE
TYPE CHOICES, PRESS ENTER.

PDF STREAM FILE:
  STREAM FILE  . . . . . .   *SPLF


                                              FILE NAME, *PSFCFG
  PUBLIC AUTHORITY . . . . .   *R             *EXCLUDE, *RWX, *RX, *RW,
                                                *WX, *R, *W, *X
```

Figure 3-6 Adding a PDF stream file

If case of segmented spooled files where the index tags are inserted using the DDS STRPAGGRP/ENDPAGGRP keywords, or by using the AFP Toolbox APIs, we assume that a programmer has control over content of the tag and can build a valid path name.

Routing tags created using the Create AFP Data (CRTAFPDTA) command may work, but there are some practical considerations. If the routing tag is based on some print data such as an invoice number, it is not likely that it is preceded by a path name. In this case, the PDF files are placed in the /root directory of the integrated file system. Similarly, the PDF extension may be missing. A secondary process may be necessary to move the spooled file to a more practical location and add the PDF extension. Information logged in the PDF conversion completion data queue associated with the PSF configuration object can assist with that process. For more information about using the data queue, see Appendix B, “PDF conversion completion data queue” on page 125.

*SPLF is an option that is not documented in the original release of *Using Mapping Objects* that was made available with the V5R2 PTFs. You can download a current version of the manual from:

http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipserveruser_i_ww

- If you specify *PSFCFG for the name of the PDF stream file, the directory that is specified in the PDFDIR parameter of the PSF configuration object is used. If no directory is specified, the spooled file is held and an e-mail is sent to the PDF administrator if one is defined.

Restriction: With V5R2, there is a limitation on the combination of parameters that you can select when creating or changing a PSF configuration object. Consequently, you can only specify the default output queue name for a PDF spooled file in a mapping object if your e-mail delivery is limited to PDFMAILSVR(*SNDDST). See 2.1.1, “Creating a PSF configuration object” on page 12, for details. Consequently, if you need to use SMTP for your e-mail, and you choose to use *PSFCFG for the name of the output queue in your PSF mapping object, the PDF output is spooled to QGPL/QPRINT.

This limitation does not exist for V5R3.

3.5 Considerations for the CRTAFPDTA command

The routing tag in the spooled file must match exactly the routing tag entry in the PDF mapping object for the spooled file or segment to be processed in the ways specified by a map entry. If you are using CRTAFPDTA to create your index tags, specify *NO for generate group names. The default value for this parameter is *YES.

If IDXGFRPNAM(*YES) is specified, either explicitly or by default, an 8-character numeric string is appended to the end of the group name. Thus, if the first segment is for customer number is 1234, a routing tag of 12340000001 is generated. This does not match a PDF mapping object entry for routing tag 1234.

4

Encryption for iSeries-generated e-mail

This chapter discusses:

- History and overview of RC4 encryption
- Why encrypt?
- Encryption levels
- User choices for encryption
- Limitations

4.1 History and overview of RC4 encryption

RC4 encryption is used extensively in software packages. It is seen in IBM Lotus Notes, Microsoft Windows CrytoAPI, Oracle Secure SQL, and Adobe Acrobat. RC4 was developed in 1987 by Ronald Rivest and is now claimed as a trade secret by RSA Security Inc. It is a stream cipher symmetric key algorithm that uses a variable length session key. Stream cipher technology serially encrypts data, one bit at a time. It is symmetric because the encrypted data can by decoded by encrypting it again with the same key. The key is termed a *session key* because the same key is used for both encryption and decrypting within the encryption algorithm.

The session key is up to 256 bytes long and initializes a 256-byte state table. This table is used for the succeeding generation of pseudo-random bytes and to produce a pseudo-random stream. This stream is XORed with the plain text to generate the ciphertext. During this process, each piece of this state table is swapped a minimum of one time.

RC4 has two parts. The first is the key setup, and the second is the ciphering. The original document has the message encrypted with the symmetric key. This key is attached to the encrypted document and transported to the recipient. The recipient uses the symmetric key to decrypt the message and obtain the original document.

The RC4 algorithm is especially strong due to the fact that a particular RC4 key can be used only one time. Additional strengths of RC4 encoding include:

- The algorithm is approximately ten times faster than the Data Encryption Standard.
- The encryption ensures the complexity of perceiving where any value is located within the state table.
- The formula ensures the difficulty of calculating which table location is used to select each sequence value.

The RC4 encoding method passes United States federal regulations. Federal officials can decode this cryptically on a deemed necessary basis. If any issues are encountered as a result of distributing secured Portable Document Format (PDF) files, the resolution of these issues is the responsibility of the author or distributor of the PDF files, not that of Adobe Systems Incorporated or IBM.

Important: To use encryption, users must have the licensed program IBM Cryptographic Access Provider 128-bit for iSeries (5722-AC3) installed on their system. Currently this is a separately orderable item that is free of charge. Users can check for a previous installation or perform an installation of this feature using the GO LICPGM command.

4.2 Why encrypt?

Encryption allows the user to specify options that preserve the integrity of the document. This includes limiting the recipient to a specified amount of editing, printing capabilities, copying, content extraction, and accessibility. This customizing can include the highest restrictions of viewing only, without allowing for changes, printing, or copying.

The PDF files that a user distributes are often routed through dozens of independent systems to reach to their final destination. Any of these can be, and are often, easily monitored. Unless encrypted, the contents of PDF files are viewed as plain text.

Provided that a master password was used at the time of creation, and that the recipient is not given access to that password, users can remain confident that their PDF will retain the

security settings that they specify. Their secure PDF cannot be inserted into another PDF file. If users choose to insert a non-secure PDF inside a secure PDF, the non-secure PDF inherits the full security settings of the outer file.

4.3 Encryption options

Using either a PDF mapping program or PDF mapping object, you can assign several encryption and security options to a PDF file that is generated using Infoprint Server.

The information in this section is intended to supplement the material in other parts of this document, and in *Using Intelligent Routing* and *Using Mapping Objects*. Refer to those sources for information regarding the syntax required to select the desired option within a PDF mapping program or PDF mapping object. You can find additional information in Chapter 3, “Using a PDF mapping object” on page 27, and Chapter 6, “Using the PDF mapping program” on page 63.

4.3.1 Assigning encryption options

The interface to assign encryption to a PDF file is different if you use a PDF mapping object instead of a PDF mapping program. However, the options are similar. One difference is that with a PDF mapping object, you can have a master password automatically generated for you. This option is not available with a PDF mapping program.

Figure 4-1 shows an example of setting encryption options using a PDF mapping object. You are only prompted for this panel if you select PDF Map Actions: Mail = *YES.

```
                              ADD PDF MAP ENTRY
TYPE CHOICES, PRESS ENTER.

MASTER PASSWORD . . . . .   master            PASSWORD, *NONE, *AUTO
USER PASSWORD . . . . . .   *NONE             PASSWORD, *NONE
ENCRYPTION LEVEL  . . . .   *128RC4           *NONE, *40RC4, *128RC4

  PRINT . . . . . . . . .   *YES              *YES, *NO, *IMAGE
  CHANGE  . . . . . . . .   *NO               *YES, *NO
  COPY  . . . . . . . . .   *NO               *YES, *NO
  CONTENT ACCESS  . . . .   *YES              *YES, *NO
  ASSEMBLE  . . . . . . .   *NO               *YES, *NO
  CHANGE COMMENTS . . . .   *YES              *YES, *NO
```

Figure 4-1 Setting encryption options using a PDF mapping object

Example 4-1 shows the RPG calculations used in the sample PDF mapping program to assign encryption values to a PDF file.

Example 4-1 Setting encryption options using a PDF mapping program

```
C*     Encryption of PDF file for e-mail
C                   EVAL      ENCRPTOFF  = 2438
C                   EVAL      ENCRPTLEN  = 71
C*
C                   EVAL      PDFMASTPW = 'master'
C                   EVAL      PDFUSRPW  = ''
C                   EVAL      PDFPRT    = '1'
C                   EVAL      PDFDOCCHG = '0'
C                   EVAL      PDFCOPY   = '0'
C                   EVAL      PDFENCLVL = '2'
C                   EVAL      PDFCNTACC = '1'
C                   EVAL      PDFCHGCMT = '1'
C                   EVAL      PDFDOCASB = '0'
```

4.3.2 Encryption level

RC4 technology has the capability to use keys that are between one and 2048 bits in length. Due to some software limitations and export restrictions, the RC4 key is often limited to 40 bits.

Quickly growing in popularity is the 128-bit encryption, 128RC4, which is the maximum security that federal regulations permit at this time. This is 2^{88} (a 3 followed by 26 digits) times more difficult to decrypt than its 40-bit encryption predecessor. Data that is sent in this manner can be trusted to arrive intact and private to the designated recipient.

PDF files that use the 40-bit 40RC4 encryption are supported by Adobe Acrobat 4.x, where PDF files that use the 128-bit 128RC4 encryption require a version of Adobe of at least 5.x. You will find that some functions, such as a few of the more advanced capabilities of accessibility, are currently limited to Adobe Acrobat 6.x.

4.3.3 Passwords

Using Infoprint Server, and either a PDF mapping object or a PDF mapping program, you may assign a user password, a master password, or both to a given PDF document or segment. The PDF mapping object allows for passwords of up to 10 characters in length. Using a PDF mapping program, you may specify passwords up to 32 characters long. The passwords can consist of any combination of letters A to Z, a to z, and numbers 0 to 9. The passwords are case sensitive.

Master password

The PDF recipient must enter the master password, commonly referred to within Adobe Acrobat as the *owner password*, before changing any security settings for the document. For example, if a print restriction is set to low resolution, and you want to change this, then you must enter the master password, before you can make any changes.

The master password is an all encompassing password in that it allows all the functionality of the user password. It also allows the PDF opener the flexibility of no restrictions. That is, if you know the master password, you simply remove any restrictions after opening the PDF. This is accomplished by using Adobe Acrobat 6.x.

You open the document with either the master or user password. Then right-click the padlock in the lower left corner of the document (or by selecting File-document security). This opens the Document properties panel (Figure 4-2). You click Change Settings to make changes to the security settings.

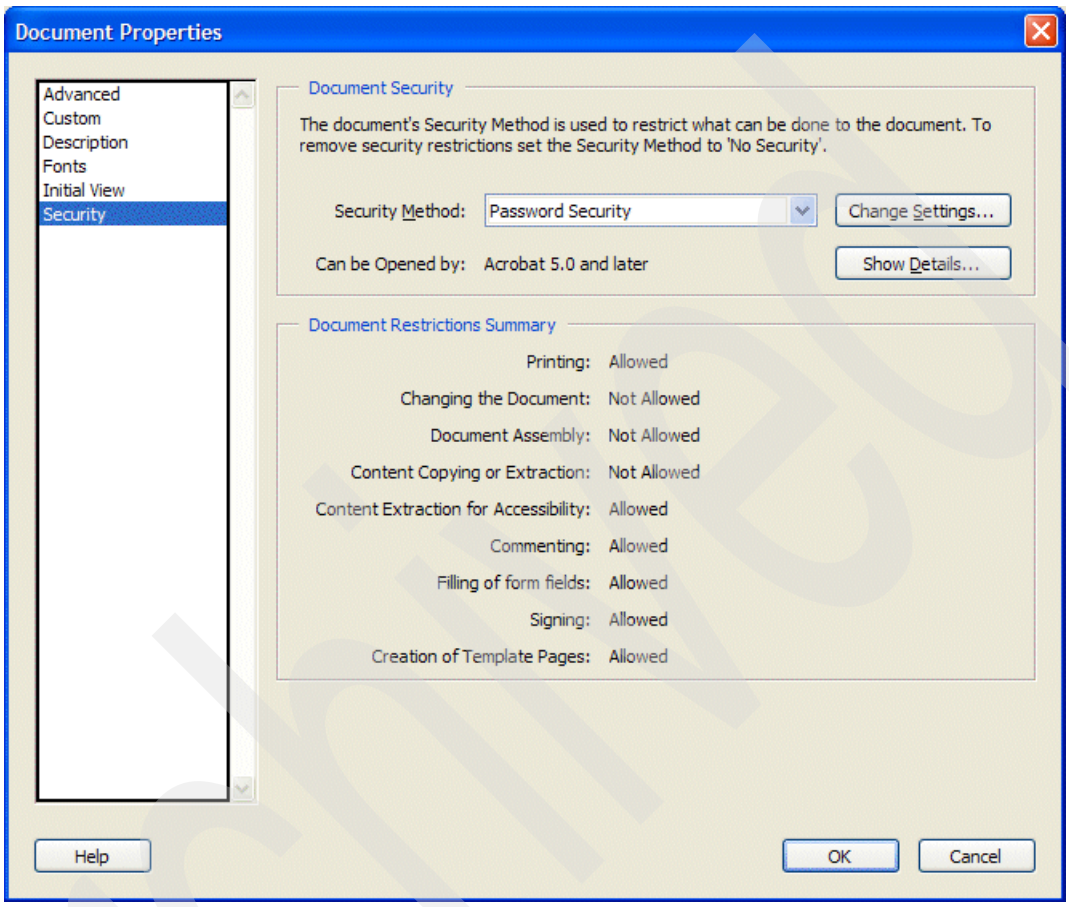


Figure 4-2 Viewing the security settings in a PDF document

With Adobe Acrobat 4.x and 5.x, you accomplish this same task by selecting File -> Document Security and clicking the Change Setting button. The Password Security - Settings window opens as shown in Figure 4-3.

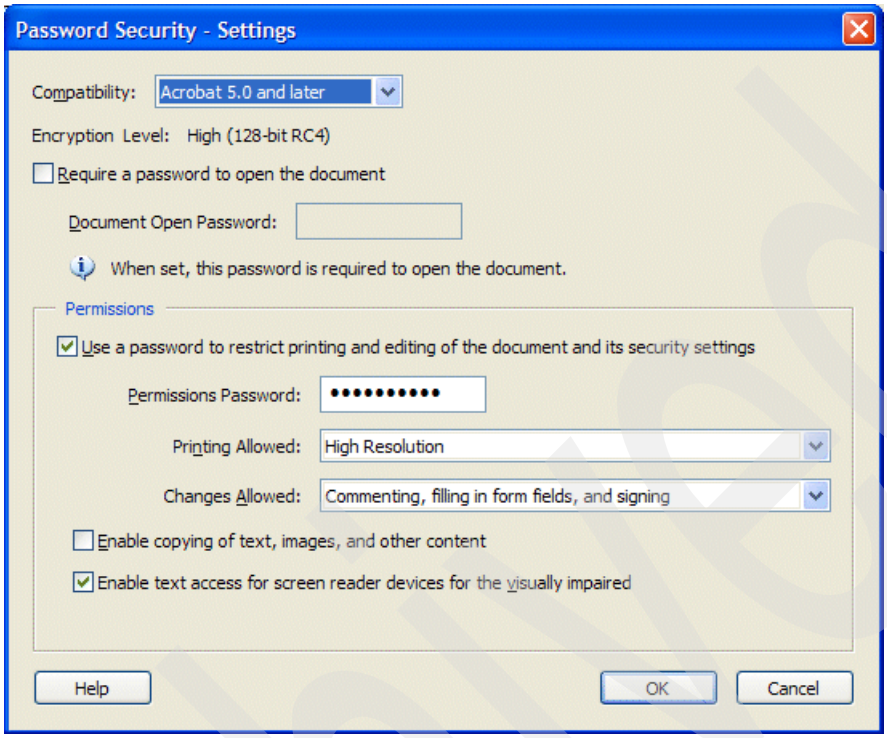


Figure 4-3 Changing security settings

If the original document is opened with a user password, a prompt for the permissions password occurs, as shown in Figure 4-4. This is the PDF master password, so users originally opening the PDF with a master password do not see this additional prompt message. You can then implement changes to the permissions settings.

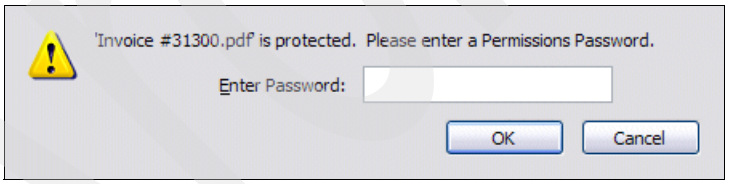


Figure 4-4 Prompt to enter master password

Using a PDF mapping object, you can specify the master password as *AUTO, which removes all options to alter document functionality, from both the opener and creator of the document. This in essence sends a document (with or without a user password required for opening) that makes permanent any restrictions that were set upon creation. This PDF, if set to change *NO, cannot be altered, either by the recipient or the sender of the document. This option is not available when encrypting a PDF document using a user mapping program.

User password

The user password is known within Adobe Acrobat as the *open password*. If the password is set as required upon PDF creation, it must be entered to give permission to open the PDF. Equipped alone with a user password, or with a document not requiring a user password, the

recipient of the PDF may open the PDF and perform all functions that are designated as allowed by the creator of the PDF. This may include printing and changing the file.

Note: Master passwords and user passwords should never be the same. Otherwise anyone who is capable of opening the document can also change the security settings.

Choosing a good password

We recommend that you use the following guidelines to set a good password:

- Make your password something that you can remember, but that is difficult for others to guess.
- Choose a password at least six characters long.
- Use mixed casing.
- Take an uncommon phrase, and use the first, second, or last letter of each word. For example, the phrase “Take me out to the ball game” becomes “tmottbg”, which can be further secured by case changing to “tmottBg”.
- Intersperse letters with numbers. Considering the previous example, you end up with “tm02tBg”.
- Intentionally misspell one or more words to make a password more difficult to decode.
- Use several of these techniques for one password.

Discouraged passwords

Avoid implementing these types of passwords:

- Passwords that are shorter than six characters or with only numbers or alphabetic characters
- Names (including user names, your real name, relative's names, pet's names) or any variation of those
- Dictionary words
- Common password techniques, such as using movies, sports, locations, asteroids, or cartoons (Hacker programs are written to specifically look for those.)

4.3.4 PDF encryption

The PDF mapping object, or a PDF mapping program similar to the one described in Chapter 6, “Using the PDF mapping program” on page 63, empowers the PDF creator to select specific encryption parameters. The following sections provide additional information about these parameters.

Print

When a document is created with PDF encryption for print set to *yes*, the user has permission to print the entire document by the mapping program, or they can submit to print only a portion of it within Adobe Acrobat. To select a page range, the user goes to the Thumbnail section in Acrobat, and uses the Shift-click combination for contiguous pages or Ctrl-click for non-contiguous pages. To choose a specific area of a PDF for printing, the user uses the select text tool. Setting PDF encryption for print to *no* prohibits all printing of the document, in whole, or in portions.

This option is available with both 40-bit and 128-bit encryption.

With 128RC4 only, the additional option of *IMAGE for low resolution printing is available. With this option, the document prints at a maximum resolution of 150 dpi. You may want to use this option if you generate PDF files from check images or other negotiable documents. Printing may be slower because each page is printed as a bit-mapped image.

Copy

When setting the options for PDF encryption for copy, the document creator delegates either an absolute *yes* or *no* in the encryption settings on the iSeries server. Within Adobe Acrobat, the user can choose to select a portion of or the entire PDF to copy. To select a portion, the select text or select image tools are used. The user may find that other utilities, such as Acrobat Catalog, rely on this specified as *yes* to properly perform their functions. The utility may also be constrained to a specific level of encryption. Therefore, the person planning for the creation of the encrypted files on the iSeries and the intended recipient should check with the software manufacturers for documentation. Setting PDF encryption for copy to *no* restricts all users from copying the entire document or any portion of the document.

Visually impaired users that rely on a screen reader may be affected by this setting. If the setting is *no* and 40RC4 encryption is used, these users cannot use screen readers. If 128RC4 is used, the copy setting does not affect the accessibility of the document, since there is a separate setting for the accessibility function for 128RC4.

This option is available with both 40-bit and 128-bit encryption.

Document change

The Change parameter as selectable in the PDF mapping object or PDF mapping program on the iSeries is a simple yes or no response. That bit is passed on to Adobe. If it is specified as a *yes*, within Adobe Acrobat, the user can invoke changes for:

- Insertion, deletion, and rotation of pages
- Adding bookmarks and thumbnail pages (within 128RC4 only)
- Filling in forms and creating form fields
- Adding digital signatures

This option is available with both 40-bit and 128-bit encryption.

Document assembly

This setting is useful in the case where the creator of the PDF file knows that all of the contents are valid, but may not all be necessary or may need reordering. The end user cannot change the verbiage or images on the pages, but they can remove or reassemble a page, chapter, etc. You should note that as the document undergoes changes with this option, if change is set to *YES, then assembly is automatically set to *YES as well.

Document assembly is only available with 128-bit encryption.

Change comments

With encryption enabled, the creator of a PDF file can limit the use of adding or changing comments. If the this option is used, the end user can choose to add or change comments to the PDF within Adobe Acrobat. Comments include highlighting, text, and editing tools (Note tool and Text edit), as well as stamps (such as Approved and Confidential). Later these comments can be edited. Editing capabilities are useful in changing the status, style, or verbiage of a comment, as well as deletion of the comment (although users may find the deletion feature to be more restricted within the Web reviews). This feature is helpful in the case of a company wanting to do a Web edit or an office review of a document, since the history of the changes is logged.

To begin or continue with commenting, the user within Adobe Acrobat, chooses view, toolbars, commenting, or advanced commenting. From the toolbar, they can select a tool. After a comment is in place, the user simply right-clicks the comment and selects Properties to change the attributes, delete the comment, or view the history of the comment. Comments can also be recorded later to reflect accepted, rejected, etc., and are displayed next to the comment in a comments list. After they are recorded as such, this record is permanent. The status display cannot be removed from the comment in the comments list.

This option is available with both 40-bit and 128-bit encryption.

Content access enablement

Content access enablement controls whether the user has permission to access the document with the utilization of screen readers. This permission is strictly to access, not to change, copy, or print. Those permissions are covered specifically within their environments.

Encryption level 128RC4 enables the user to retain accessibility support while adding security. This allows disabled users to access the PDF documents. Visually impaired users can use screen readers or another type of text-to-speech engine for audible support. Within the PDF mapping object, the user selects *YES next to PDF Accessibility. The following options are supported. A similar option is available to use a user mapping program.

- **Natural language specification**: The recipient can stipulate the language in which any text within the PDF is read aloud, for example, English, Spanish, or Japanese. This includes both text strings and text within the content streams.
- **Alternate descriptions**: Images, formulas, and other items that do not readily translate to human-readable text can have textual descriptions included for use with a screen reader.
- **Replacement text**: Support includes using replacement names for nonstandard images, such as illuminated characters, custom characters, and online graphics.
- **Keyboard navigation**: PDF files can be navigated through windows, navigation panels, etc. without a mouse. This requires both Microsoft Windows and a keyboard.
- **Customized text size**: The text size can be enlarged to make it more visually readable.
- **Expansion of abbreviations and acronyms**: Abbreviations and acronyms are often difficult for screen readers. The expansion of both of these can be accomplished through an E property, which is attached to a marked-content text string with a Span tag.

Note: To obtain the highest degree of accessibility support, the receiver should use Adobe Acrobat 6.x.

Content access is selectable only if you use encryption level 128RC4. If you use encryption level 40RC4 and have disabled PDF copy, content access is also disabled.

You can find more extensive information about content accessibility published by the World Wide Web Consortium (W3C) on the Web at:

`http://www.w3.org/WAI/`

Review the link Web Content Accessibility Guidelines 2.0 Working Draft made public on 24 June 2003:

`http://www.w3.org/TR/2003/WD-WCAG20-20030624/`

4.4 Limitations

When planning for encryption, be aware of the following limitations:

- The implementation of encryption to the PDF files is limited to e-mail only if you are using a PDF mapping object as described in Chapter 3, “Using a PDF mapping object” on page 27, or a PDF mapping program similar to the one described in Chapter 6, “Using the PDF mapping program” on page 63. However, we have developed a technique that makes a copy of the temporary version of the generated encrypted PDF file and allows it to be stored in the integrated file system (IFS). This is described in Chapter 9, “Saving encrypted PDF files” on page 111.
- RC4 40-bit encryption is supported by Adobe Acrobat 4.x, 5.x, and 6.x.
- RC4 128-bit encryption is supported by Adobe Acrobat 5.x and 6.x. A document created with 128-bit encryption cannot be opened with Acrobat 4.x or earlier.
- A secure PDF file cannot be inserted into another PDF file.
- The recipient's content accessibility language selection and any specified replacement text may be overridden by a text string encoded in Unicode. These can include an escape sequence to indicate the text language.
- Auto-generated master passwords are supported only by the PDF mapping object. User-supplied PDF mapping programs do not perform this option.

5

iSeries color applications

The demand for color in business documents has been satisfied for years by using preprinted forms. Variable information was added at print time in black and white, providing the most cost-effective method of producing full color business documents. That print environment relied on the fact that there was a limited number of changes in document format and that they could be predicted well in advance.

As the rest of this Redpaper illustrates, the high-return of e-business communications deployments requires fully electronic documents as a prerequisite. If monochrome preprinted forms can be migrated to electronic form, why not color? The cost of color printing has come down significantly. And, the color capabilities of iSeries, which are fairly basic, have been significantly upgraded over the last couple of OS/400 releases.

This chapter contains information about generating and using color in iSeries applications. It introduces you to the iSeries color environment and how you can add color to your output. The output architecture has been enhanced with several full color models. There are several choices for the design of color documents. Infoprint Server supports both color conversion and the transform of iSeries color documents into PDF.

5.1 Definitions

To understand the extent of support on the iSeries for Color as described in this chapter, you need to know the following terms and acronyms:

- **Advanced Function Print Utility (AFPU)**: A green-screen-based product to create overlays and a utility (Print Format Definitions) to merge a database with a form.
- **Advanced Print Utility (APU)**: A spool reformatting application for Systems Network Architecture (SNA) Character Set (SCS) output, to re-engineer output include barcodes, overlays, and fonts.
- **Advanced Function Presentation data stream (AFPDS) driver**: A PC-based driver to print PC applications to Intelligent Printer Data Stream (IPDS) printers and to create Advanced Function Presentation (AFP) overlays and page segment resources.
- **CIELAB**: Color model that specifies the luminance value and two integers to specify the chrominance differences.
- **CMYK**: Color model that uses Cyan, Magenta, Yellow and Black.
- **FS11**: A function set of the Image Object Content Architecture (IOCA) definition for color page segments and images that uses the 15 basic IOCA colors.
- **FS45**: A function set of the IOCA definition for full color page segments and images that includes the RGB and CMYK color models.
- **Highlight**: Hardware dependent model for highlight color hardware.
- **Host Print Transform (HPT)**: A spool writer capability to convert *SCS and *AFPDS EBCDIC spooled files to Printer Control Language (PCL) ASCII.
- **Image Object Content Architecture (IOCA)**: For the definition of graphic objects.
- **Print Services Facility/400 (PSF/400)**: A feature of OS/400 required to print to IPDS printers defined as AFP=*YES.
- **Page Printer Formatting Aid (PPFA)**: A command-based page formatting tool.
- **RGB**: A color model that uses the basic colors of Red, Green, and Blue.

5.2 iSeries color models

The iSeries has supported limited color definition since its inception. The initial support was only eight basic colors. These eight basic IOCA colors provided the capability to support color displays, view print output in color using the AFPDS viewer, and print to color IPDS impact printers. In Version 2 Release 3 (V2R3), support was added to print to color ASCII printers via Host Print Transform for the IOCA colors. The color was specified using Data Description Specifications (DDS) keywords for data, static text, lines, boxes, etc. The color keywords are limited to printer file types *IPDS and *AFPDS.

The Version 5 Release 1 (V5R1) announcement of new DDS color keywords expanded the native iSeries color support adding the *RGB, *CMYK, *Highlight, and *CIELAB color models. Text, lines, boxes, graphics, and even barcodes can have color attributes. Of the four color models added, only two are of interest in creating iSeries Applications. The two that we explore are the *RGB and *CMYK enhanced color models.

The *RGB and *CMYK models are based on the colors that make up their names. *RGB is for red (R), green (G), and blue (B). *CMYK represents cyan (C), magenta (M), yellow (Y), and black (K).

5.2.1 IOCA color model

The color capability that represents the lowest level supported on the iSeries uses the eight basic colors that are defined by the DDS color keywords. These capabilities were introduced in System/38™ and System/36™. It continued with OS/400. The color keywords that are supported are:

- BLK: Black
- BLU: Blue
- BRN: Brown
- GRN: Green
- PNK: Pink
- RED: Red
- TRQ: Turquoise
- YLW: Yellow

These colors can be used for both display files and printer files. The colors are used to format color screens and to allow the viewing of color printer output using the AFP Viewer. The colors can also be printed on the color impact IPDS printers. With the addition of Host Print Transform for color printers, the IOCA colors can be printed on color laser printers that support PCL.

5.2.2 *RGB and *CMYK color models

Additional color support was added with the V5R1 OS/400 release. Four additional color models were added: two were printer hardware related (Highlight and CIELAB) and two provided a full range of color options. Those two models are the *RGB and the *CMYK models.

The *RGB color model is based on the use of integer percentages of the three primary colors: red, green, and blue. The *CYMK model is based on three colors and black used in integer percentages. The three colors are cyan, yellow, and magenta.

5.2.3 Release dependencies

The color support from Version 1 Release 1 (V1R1) to Version 4 Release 5 (V4R5) was for the basic IOCA colors for a limited number of objects. They could be specified for the basic elements within DDS including text fields, lines, and boxes. Single color IOCA page segments were not officially supported, but tools were available to change the color of page segments from black to one of the supported IOCA colors.

Infoprint Designer (announced with V5R1) supported the V5R1 enhanced color models. When announced, it supported all of the extended color attributes for all objects except page segments. Support for color IOCA page segments was added in later releases of Infoprint Designer, delivered via a program temporary fix (PTF). In addition to the eight basic IOCA colors, seven extended IOCA colors are included for a total of 15 IOCA colors.

The latest release of Infoprint Designer, Version 1.15, supports full color, FS45 page segments. The FS45 function is similar in architecture to the JPEG standard. It is a highly-compressed, high-resolution full process color standard. The Infoprint Designer Image Editor can import color objects in TIFF, JPEG, and other supported formats and convert them into full color FS45 AFP objects. These full color page segments can only be used in Version 5 Release 2 (V5R2) applications on systems that are at the now available PTF levels.

The Infoprint Server program product, available for V5R1 and later, supports the new color models provided by both Infoprint Designer and DDS. The only limitation of Infoprint Designer

for V5R1 (at the latest PTF level) is that it does not support the FS45 full color page segments. Only the multi color IOCA page segments are supported in V5R1 applications.

The ability to use the full capabilities of the iSeries colors requires V5R2 at the now available PTF level.

5.3 Creating color objects

There are several tools to create full color documents:

- Infoprint Designer for iSeries
- DDS (of the printer file)
- Advanced Function Presentation (AFPDS) printer driver
- Infoprint Server image transforms (GIF, TIFF, and JPEG to IOCA)

5.3.1 Creating color objects using Infoprint Designer (5733-ID1)

Infoprint Designer supports all of the color models. Color is available in all of the functional components of Infoprint Designer, the Overlay Editor, the Layout Editor, and the Image Editor.

The Image Editor provides the capability to manipulate graphic objects at the individual pel level. The Image Editor can open the following images:

- AFP image
- BMP image
- GIF image
- JPG image
- TIF image
- ICO image
- PTR image

The color for the pel is chosen by using Define Primary and Secondary Colors if you are coloring a black and white image. Imported color images are already color associated with each pel. The color dialog box opens to display all of the color models. To store the color object for creation of a color page segment, it is necessary to be at Image Editor Version 1.15. This allows the color object to be saved as a FS45 IOCA image.

Compression algorithms are available to reduce the size of the resulting AFP image. The compression algorithms that are available depend on the input object type and whether it is grayscale.

The Overlay Editor is used to create color overlays and to import color AFP images that are uploaded to the iSeries as FS45 page segments. The page segment color is not manipulated by the Overlay Editor. All objects with the exception of boxes with rounded corners can be colored using any of the available color models. The rounded corners are rendered as image segments and do not take on the color attributes of the lines used to make the box. The color tab in the attributes for an object is opened to show the color options.

The Layout Editor is used to map the line data from the iSeries print application onto the form designed using the Overlay Editor. Each element can have color attributes. You can use the same window to add color using any of the color models.

5.3.2 Color support using data definition specifications

DDS is the standard page formatting tool when used in conjunction with program logic for building complex print applications. DDS specifications have been enhanced continually

throughout the releases with the addition of many new capabilities, such as the use of variable names for overlays, page segments and fonts, as well as the addition of new keywords (COLOR) and functions.

The valid data types for the color keyword are A, S, and F. Any field with one of these data types can have color attributes using the DDS keyword, COLOR.

The format of the keyword is:

```
COLOR(color-name | *RGB rvalue gvalue bvalue |
      *CMYK cvalue mvalue yvalue kvalue |
      *CIELAB lvalue c1value c2value |
      *HIGHLIGHT hvalue coverage)
```

Color is specified by using one of five methods detailed in the format:

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

For example, the color keyword for the *RGB model is:

```
COLOR (*RGB 10 30 50)
```

These choices result in a color that is a blend of 10% red, 30% green, and 50% blue. White, for example, is one 100% of the colors, and black is zero percent of each color. The color described in previous the DDS example is a medium blue with a hint of gray.

DDS keywords are processed by the iSeries application as pages are written to the iSeries spool. The resulting spooled file with a Print File type of *AFPDS contains the color attributes.

You can view the spooled file in color when it is on the spool by using the AFP Viewer. The viewer is installed as part of the iSeries Access Navigator. AFPDS spooled files are opened using the Viewer when they are selected.

For printing, the color attributes are sent to IPDS printers by PSF/400 where the capabilities of the hardware determine the results. The support is detailed in the following section.

The iSeries output writer can also use Host Print Transform to transform the color attributes (IOCA only) for PCL color printers. All of the color attributes are supported by Infoprint Server for documents converted to PDF. Note the restrictions listed in this chapter on color support by release.

5.3.3 AFPDS Driver

The AFPDS Driver is included in this summary of tools because it can be used in conjunction with PC applications to create iSeries color resources. The resources that can be created are overlays and page segments. We recommend that, if you are planning to use the AFPDS Driver to create color resources, download the latest version from the IBM Printing Systems Division Web site at:

http://www.printers.ibm.com

From that page, follow these steps:

1. Select **Downloads and drivers**.
2. Select **Software downloads**.
3. Select **IBM AFP drivers**.

The process for creating an AFPDS file for an overlay or a page segment is similar. First a PC application that can create or view color resources is used to produce the desired result. The application can manipulate a color JPEG file or create a form with lines and boxes.

You select the Print icon. The printer to use is the IBM AFP 300 (recommended). You can set additional options for the AFP Driver. You simply open the properties, click Output Type, and from the next window, select either overlay or page segment. Then you select Image Options and open the properties to select the color to use. There are multiple options, including 16 color or full color.

The full color page segments created using the AFPDS driver are FS45 objects. The resulting PC file is then imported into the iSeries using the iSeries Navigator's AFP Manager. For information about installing the AFP Manager, see *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250.

Note: Using the same JPEG file, page segments created using the AFPDS driver were equivalent in quality and size to those created using Infoprint Designer's Image Editor.

5.3.4 Infoprint Server image transforms

Infoprint Server image transforms can also transform GIF, JPEG, and TIFF objects into AFP objects. These transforms are batch conversions that run in the Windows environment. The conversion process can be automated by using the Start PC Command function and then uploading the generated IOCA data file. The parameter specified is `-outcolor [Output Color Model]`.

For more information, see:

- *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250
- *Infoprint Server for iSeries: User's Guide*, G544-5775
- *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775

5.3.5 Release dependencies

The color capabilities of the iSeries have matured quickly since the announcement of V5R1. The use of these capabilities can be limited, not by the iSeries, but by the printer hardware that is available to produce applications that use the color attributes. If we ignore the limitations placed on the application by the printer, then the major color dividing line is OS/400 V5R1.

Infoprint Designer (announced with V5R1) supported the enhanced color models with the exception of color page segments that contained IOCA colors. That capability was added through a PTF.

Infoprint Designer color use for OS/400 release V4R5 is limited because the only capability for output from Infoprint Designer applications in release V4R5 is to IPDS printers. There are currently a limited number of color IPDS printers. Since all IOCA colors default to black in many black and white IPDS printers, adding color does not change look of the printed page. Most of IBM cut sheet printers above 60 pages per minute and high end continuous form printers support color to grayscale. These printers can output Infoprint Designer and DDS created IOCA color applications with Release V4R5.

The V5R1 addition of the CVTLINDTA print file parameter allows Infoprint Designer applications to be written to the spool as *AFPDS. Printing of the AFPDS to ASCII color printers is supported, but is limited to the IOCA colors without IOCA color page segments.

The other output option in V5R1 was into color PDF using Infoprint Server. These documents can contain both the *RGB and *CMYK colors with IOCA page segments.

The Infoprint Image Editor function of Infoprint Designer was upgraded early in V5R1 to create multi-color IOCA page segments using the basic eight colors plus the additional seven extended IOCA colors. These color page segments could only be output into color PDF files. Many of the restrictions on the use of color in OS/400 V5R1 remain. The printing option provided by Infoprint Server is the ability to route the PDF to a remote output queue. The resulting PDF file can be printed on both black and white and color PDF capable printers. Color PDF files created by Infoprint Server can also be viewed using Acrobat and printed using PC print drivers.

Infoprint Server produces PDF files that use the extended color attributes provided by the V5R1 announcements and PTF enhancements for color page segments. OS/400 Version V5R2 is required to take advantage of the full color capabilities including FS45 full color images.

5.3.6 PTF levels

We recommend that you always use the latest level of PTF for OS/400, Infoprint Designer, and Infoprint Server. In addition to the color enhancements, many other enhancements and fixes are provided. We also recommend that you use the PTFs that relate to spooling, TCP/IP, and Host Print Transform if physical printing of color objects is planned. To check for the latest listing of PTFs, see the iSeries Knowledge Base on the Web at:

`http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase`

From that page, follow these steps:

1. Select the category **Print**.
2. Select **Search**.
3. Enter the search argument `ptf` or the document IDs from the following list.

All of the categories of PTFs for printing and related software are listed by release. V5R1 and V5R2 are usually listed in the same tables. The Knowledge Base documents that apply are:

- V5RxMx PTF Listing for AFP Printing and Infoprint Server, 23381552
- Infoprint Designer PTF Listing, 26035749
- V5RxMx PTF Listing for Host Print Transform Printing, 24474221
- V5RxMx PTF Listing for Print (Spooling), 325828275
- V5RxMx PTF Listing for TCP/LAN Printing, 23383453

5.3.7 Summary of color support by product

Adding color to iSeries documents is easy. Building a color document that accomplishes the business objectives takes some planning. For color documents that are to be printed, you must consider the technology level of the printer and the iSeries print method used. See 5.4, “Output options for color documents on the iSeries server” on page 58, for additional details.

The tools used to create overlays and applications each have their own set of capabilities. Infoprint Designer supports overlay, page segment, and reformatting using all of the color models available for V5R1 and later. The AFPDS driver is used to generate an overlay or page segment, including FS45 objects, from a PC-based application. Any restrictions on colors are based on the PC applications capabilities. Older tools, such as AFPU, can be used to create IOCA color overlays. APU can be used to process *SCS spooled files and create AFPDS spooled files that contain colored elements. AFPU and APU support the basic IOCA colors only.

The AFP to PDF conversion in Infoprint Server follows the AFP architecture and produces similar results. Table 5-1 lists the capabilities of the IBM products that support color.

Table 5-1 IBM iSeries products that support color

	Basic IOCA colors	Extended IOCA colors	RGB	CMYK	Other color models
Infoprint Designer	Yes	Yes	Yes	Yes	Yes
Infoprint Server	Yes	Yes	Yes	Yes	Yes
DDS	Yes	No	Yes	Yes	Yes
Printer File	Via a color overlay only	Via a color overlay only	Via a color overlay only	Via a color overlay only	Via a color overlay only
AFPDS Driver	Yes [1]	Yes[1]	Yes[1]	Yes[1]	
AFP Utilities (overlays)	Yes (for type 9)	No	No	No	No
AFPU - PFD Applications	Yes	No	No	No	No
APU	Yes	See [2]	See [2]	See [2]	See [2]
PPFA	Yes	Yes	Yes	Yes	Yes

1. Color is specified in a PC application. The creation of the AFP Resource, including the color attributes, is controlled by the driver image options chosen.
2. Overlays and page segments created with Infoprint Designer can be specified.

5.4 Output options for color documents on the iSeries server

The color options prior to OS/400 V5R1 were to print color documents to PCL color printers using the iSeries Host Print Transform. The Host Print Transform objects for the IBM 1220 Color Printer provided support for the 15 IOCA colors. Host Print Transform has many other restrictions. One is the lack of support for color page segments.

The announcement of Infoprint Server with OS/400 V5R1 provided the capability to output color documents to PDF. Those documents can be e-mailed, stored on the integrated file system (IFS), or output to a remote output queue. The PDF files can be directed to the PDF capable printer for printing. The current line of IBM black and white workgroup printers, the Infoprint 1332, 1352 and 1372, have PDF file printing as a native feature. The printer converts the color in the PDF file to grayscale and eliminates the restrictions imposed by the OS/400 Host Print Transform. Infoprint Server can direct PDF output to the queue to print these jobs. The color to grayscale used in the PDF conversion in the printer results in improved readability and lower toner use.

The Workgroup black and white and color printers can be configured as normal print devices using the PJL and Simple Network Management Protocol (SNMP) drivers. These devices must use Host Print Transform. The AFPDS to PCL conversion (Host Print Transform) does not support color page segments, but converts text, lines, and boxes with IOCA color attributes. Color page segments are printed in black and white.

5.4.1 PDF creation for e-output and print

The capability to produce PDF output from iSeries spooled files was announced for V5R1 and later releases. Infoprint Server provides options for output via e-mail, as a stream file to the IFS or to a remote device via a remote output queue.

If the iSeries document is converted to a PDF spooled file, the value for Device Type is *USRASCII. These spooled files may be directed to a PDF capable printer using a remote output queue or the SNDTCPSPLF or LPR commands. The resulting PDF file is identical to the results produced on a full color IPDS printer.

Printers that are PDF capable process the PDF file, rendering color or converting it to grayscale for non-color printers. IBM has color printers that support color PDF printing with the addition of the ImageQuick feature.

If the iSeries document is converted to an e-mail attachment, or stored as a stream file in the integrated file system, a client-based application may be used to send the file to a PDF capable printer.

The color PDF files, whether delivered as an e-mail attachment, or stored in the IFS, may be viewed be the recipient using Adobe Acrobat viewer. We recommend that you use Adobe Acrobat 5.0 at a minimum.

5.4.2 Viewing color AFP spooled files

You can view spooled files in full color using the IBM AFP Viewer. This is available as a component of iSeries Navigator. You simply select Basic Operations-> Printer Output. Selecting the spooled file from this panel automatically invokes the AFP Viewer.

5.4.3 Color PCL and PDF printers

Printing color documents to a PCL color printer using Host Print Transform is supported for IOCA colors. Using *RGB or *CMYK colors with Host Print Transform is not supported and has the same result as printing without any color attributes.

IBM color workgroup printers have an optional color ImageQuick capability that can print color applications. The only application limitations are those imposed by Infoprint Server based on the OS/400 version. V5R1 provides support for full color objects except page segments (FS11 page segments only), and V5R2 supports full color options. Color applications can be formatted into PDF files by Infoprint Server and routed to the printer via a remote output queue.

5.4.4 Black and white PCL and PDF printers

Many of the previous line of workgroup printers from IBM have the capability of process PDF files. Infoprint 1120, 1125, 1130, 1140 and 1145 with an optional ImageQuick feature installed can be defined to the iSeries via remote output queue and print the color PDF files using their color to grayscale algorithms.

5.4.5 IPDS workgroup black and white printers

The current IPDS implementation for the black and white workgroup printers does not support color to grayscale. Printing color objects to an IPDS workgroup printer results in the color attributes being ignored. All objects print in black and white. The currently available IPDS

capability for the color workgroup printers is for black and white only. Printing to the color workgroup printer is equivalent to printing to any black and white workgroup printer.

Since the color attributes are ignored, blue text printed on a wide yellow line results in a wide black line. All color objects are printed in the same intensity of black.

5.4.6 Production black and white printers with grayscale

The IBM production cut sheet (60 pages per minute and above) and continuous form printers have robust controllers that support color to grayscale for AFP printing. Some of these printers also have native capabilities to print PDF files. These printers can provide quality black and white output from the color applications, eliminating the need to create two different overlays and additional page segments, one set for printing and the other for output to PDF. Color can be an integral part of your iSeries application development.

5.4.7 Full color IPDS printers

The only IBM printer that supports full FS45 color is the IBM Color 130 Plus. This printer has not been formally tested with iSeries output, but supports the full color AFP data stream.

5.5 Other considerations

Using color in the iSeries is not unlike creating a graphic color application on a PC. Objects, page segments, overlays, and text may overlap each other. The results can often be predicted. A common technique for document creation is to build the background first and add the items like the variable data last to the foreground. In general, the order in which these elements are sent to the printer determines what is hidden.

A color background created using a wide line, printed first with a color attribute of yellow, followed by a text message in a *CMYK color of dark blue, results in PDF file where the dark blue text on a yellow background appears. If the reverse order is used, only a wide yellow line is seen in the PDF file.

Control of the order of printing of the elements is done best in DDS applications.

5.5.1 Color elements and the print order

Overlays can be added to each page using native iSeries print files (front and back overlays keywords). Since the overlay is added to the page, the lines, boxes, etc., of the overlay may be sent to the printer last and cover the spooled file variable data. Therefore, it is important to understand the order in which objects are sent to the printer if we design applications that do not result in hidden data.

The use of DDS to create the document gives the greatest level of control. The order in which the DDS record formats are written determines what is on the top (in the foreground) when the page is output to PDF or printed. Writing text and then adding a wide color line or an overlay can cause the text to be in the background and covered. Controlling the order, overlays first, page segments next, and variable data last creates documents that are equivalent to those printed on preprinted paper.

5.5.2 Full color image and page segment size

The greater the number of colors used in a graphic object is, the larger that object is. Scaling the objects to the size required and the resolution that best matches our expected output

device results in smaller objects. The resulting size of the AFP resources and the PDF output is directly related to the size of the PC file.

Full color JPEG, TIFF, GIF, and other file types can be turned into full color FS45 AFP objects. It is possible to convert digital color photographs into page segments, to be output along with data on the iSeries to produce a PDF document that contains that picture. The addition of FS45 color support to the iSeries in V5R2 has greatly expanded the graphics capability for e-output. Corporate logos can be produced in full color and incorporated into documents. The high end of the printer hardware line (60 page per minute cut sheet and above) also supports the FS45 specification and converts the color image to grayscale. The primary use of the FS45 page segments today is in AFP documents to be converted from iSeries AFPDS documents into PDF documents.

Full color objects by nature require larger amounts of storage. Depending on the density of the graphic object (pixels), a full color picture can easily require 1 MB of space. Converting one of these objects into iSeries page segments, while maintaining the same quality, results in an iSeries object with larger storage requirements. The use of the full color objects can also affect the AFP to PDF conversion performance.

To illustrate the variations that can exist in image objects, two graphic objects were chosen for size comparisons: one digital photograph with a low resolution and the other a high resolution. Other JPEG and TIFF images were also tested. Table 5-2 shows the results using the two JPEG objects described. A simple application printed the page segment to PDF.

Table 5-2 Color image size comparison

	Resolution	JPEG size	Image Editor AFP size	Page segment size	PDF size
Sample 1	640 x 480	42.6 KB	1.17 MB	270 KB	99 KB
JPEG compression			46.9 KB		
TIFF compression			377 KB		
Sample 2	1171 x 2250	835 KB	15.2 MB	15.83 MB	6.84 MB
JPEG compression			718 KB		
TIFF compression			6.72 Mb		
Sample 2 sized with Image Editor 2 pels to 1 pel; no compression		835 KB		4 MB	1.96 MB

The conclusion is that the use of Color images without compression or resizing is highly discouraged. Infoprint Designer has the tools to size and compress the images to produce efficient iSeries objects. Such objects as corporate logos are usually smaller and contain fewer colors than a photograph. Their conversion to iSeries objects does not significantly increase their size or the size of the resulting PDF files.

5.5.3 PDF file sizes

The AFP architecture is designed to cache resources in the printer, reusing them. The resulting PDF can be structured to reuse the overlay.

Applications that use the same or small number of resources (page segments and overlays) result in efficient PDF files that grow by the amount of variable data used on the new pages plus page controls. A 20-page green-bar report, with an overlay containing a graphic page

segment, increases in size just over 100,000 characters. Each page contains about 3,000 characters. Therefore, the data increase is over 60,000 characters. This is expected. However, when the same page segment is placed dynamically by an iSeries application on each page, the AFP to PDF conversion cannot cache the resource. The resulting PDF is over 20 times the size of a one page PDF.

The resulting multi-page PDF file size varies significantly depending on the way the resources are used. Application design should take into consideration how the pages are being built and the size of the resources created. Infoprint Designer provides the tools to control the size of graphic objects, and Infoprint Server attempts to build PDF files that take advantage of the AFP architecture.

6

Using the PDF mapping program

This chapter contains information that a programmer needs when writing a custom Portable Document Format (PDF) mapping program to use with Infoprint Server for iSeries.

Important: The ability to use a PDF mapping table for intelligent routing is delivered as program temporary fixes (PTFs) for V5R2 for Infoprint Server for iSeries and Print Services Facility for OS/400. This function is included in the base support of V5R3. For a current list of PTFs, refer to iSeries Software Knowledge Base document 23381552 at:

http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase/

For further assistance obtaining PTFs, contact your local IBM Support Center.

Prior to reading this chapter, consider the following notes:

- This chapter refers often to the document *Using Intelligent Routing*, which is loaded on your iSeries as a PDF file when you install the new PTFs. You can find it in the integrated file system (IFS) as \QIBM\ProdData\InfoprintServer\Transforms\Intelligent_Routing.pdf. You can find updates to the *Using Intelligent Routing* document on the IBM Printing Systems Division Web site. Current versions of this document are located at:

  http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipserveruser_i_ww

- The program discussed in this chapter has several names in different publications. In this document, it is referred to as *PDF mapping program*. This is synonymous with references to mapping program, mapping exit program, e-mail exit program, and the Print Services Facility (PSF) user program found in other documents. These names correspond to the program referenced by the PDFMAPPGM parameter of the PSF configuration object.
- The sample PDF mapping program, MAPPGM, in this chapter is listed in “Sample PDF mapping program in RPGLE” on page 164. Line number references in the text of this chapter refer to that program.
- The processing that is done on the iSeries server to convert spooled files to PDF files and route them to the desired destination is performed by a combination of Print Services Facility for iSeries and Infoprint Server. For simplicity, this chapter references the combined process as Infoprint Server, unless noted otherwise.

6.1 When to use the PDF mapping program

The PDF mapping program provides almost all of the same functions that are available through the IBM-supplied PDF mapping program, which uses a PDF mapping object. The primary difference lies in the fact that the programmer can incorporate additional logic and flexibility to the selection criteria for the files and to customize the resulting action. See Table E-1 on page 137 for a comparison summary between using a PDF mapping program and using the PDF mapping object.

Some scenarios in which you can benefit from using a custom PDF mapping program are:

- Company A is implementing Infoprint Server for the purpose of distributing customer documents, such as invoices and statements, according to the wishes of each individual recipient. The customer delivery preference, contact information, e-mail address, and FAX information is in the customer contact database. By using the PDF mapping program, Company A does not have to transcribe the information to the PDF Mapping object.
- Company B wants to use contact information from the customer data file to customize the message that appears as part of their e-mail. This is illustrated in the sample exit program in Appendix E of *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250.
- Company C wants to add discount offers in the form of PDF files as an extra attachment to e-mail. Customers receive different offers based on the previous month's sales, which is determined dynamically at the time the e-mail is generated.

6.2 Other steps to prepare for using intelligent routing

Writing the e-mail exit program is one of many steps needed to set up Infoprint Server for intelligent routing. You can find a detailed description of all the steps in 2.1, "Configuring for intelligent routing" on page 12.

6.3 Programming concepts you need to know

This section discusses a few important concepts that are used in the PDF mapping program:

- Reserved fields
- Binary fields
- Offsets

6.3.1 Reserved fields

All fields designated as reserved fields must be assigned values of hexadecimal 00s. This can be done by assigning the string '00' for each character in the field or by using the reserved word *ALLx'00'.

Important: In V5R1, this was not enforced. If you are using a PDF mapping program that was written for V5R1, such as the sample in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, check that the reserved fields are set correctly before you migrate it to V5R2 or V5R3.

6.3.2 Binary fields

Many of the fields being passed to and from the PDF mapping program have a type of BINARY(4). That means that 4 bytes of data are used in the buffer to represent a given number. In the Data Definition Specifications (DDS) of the sample program, we use two different methods to define these BINARY(4) fields. You can find examples in the sample PDF mapping program in “Sample PDF mapping program in RPGLE” on page 164.

Absolute (positional) notation

Absolute notation defines the fields of the input information data structure. This is essentially an arbitrary decision. The sample PDF mapping program used for this document is loosely based on the sample from *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, which uses absolute positioning. Since few changes are to be made to this part of the program, we continue to use absolute positioning for the fields in the input information data structure.

An example of a binary field that is specified using absolute positioning is SPLNO. It is found at exactly position 37 through 40 from the beginning of the structure, clearly taking up four positions. It is a binary field that contains zero decimals. The compiler assigns a length of 9 to this field because the equivalent packed decimal number that this field could hold can be up to 9 digits long.

Length notation

We use the length notation to define the fields in the output information data structure. This method is chosen for the output information data structure to make it easier to make modifications, such as changing the length of variable length field without recalculating and retyping the absolute positions of all fields that follow it in the structure.

MSGLEN and ADDRLEN are two binary fields of length 4 that are part of the PDF mapping program output information data structure, OUTDS. The ADDRLEN field begins in the position immediately following the MSGLEN field. Note that a length of 9 is used for these two fields. The compiler needs to know that the longest number that this field holds can be 9 digits long. But it is also smart enough to realize that since the data is going to be stored as binary. It only needs to reserve four spaces for that field. For example, the DISPOSTN, CALLAGIN, and RES2 fields consume a total of four characters, so MSGLEN begins in position 5. ADDRLEN begins four positions past it at position 9 (lines 21 through 26). The stand-alone fields, such as INPUTLEN, are also defined using the Length method (lines 147 through 149).

For more information about coding data structures and binary fields in RPGLE, refer to *WebSphere Development Studio, ILE RPG Reference, Version 5*, SC09-2508.

6.3.3 Offsets

The key to building the mapping program output data structure lies in understanding the concept of offsets. The term offset can be interpreted as “how far it is from the beginning”. An offset of 0 means that you are 0 positions from the beginning or at position 1. An offset of 99 takes you to position 100. The concept of offsets may be new to traditional RPG programmers, but other languages, such as C or Java, use offsets as a matter of course.

The use of offsets is necessary since not all data passed between the PDF mapping program and Infoprint Server has fixed or predetermined sizes. For example, the e-mail address can be up to 16 M characters in length, but in most cases, that is not a practical length to work with. Since different programmers can use different lengths for the address to suit their own needs, offsets act as pointers that tell Infoprint Server where the variable length field ends and where the next field of information begins.

For example, the e-mail address must begin at offset 287 because all of the fields that precede it are fixed length. In the sample PDF mapping program, we allow for a maximum of 255 characters to specify e-mail addresses. The next field, EXTLEN, begins at offset 287 + 255 = 542. Another user may need 1000 characters for e-mail addresses. The user places EXTLEN at an offset of 287 + 1000 = 1287.

6.4 PDF mapping program input and output parameters

The PDF mapping program is used by Infoprint Server to determine what should be done with the PDF files built by Infoprint Server and whether they should be encrypted. Infoprint Server provides information to the PDF mapping program via the input information data structure. The PDF mapping program responds with a string of data as defined by the output information data structure. Table 6-1 lists the parameters passed between Infoprint Server and the PDF mapping program.

Table 6-1 PDF mapping program required parameter group

Parameter	Input or output	Type
Mapping program input information	Input	Char(*)
Length of input information	Input	Binary(4)
Mapping program output information	Output	Char(*)
Length of output information buffer	Input	Binary(4)
Length of output information available	Output	Binary(4)

See *Infoprint Server for iSeries: User's Guide*, G544-5775, or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, for full descriptions of the required parameters.

6.4.1 PDF mapping program input information

Infoprint Server passes several fields (see Table 6-2) to the PDF mapping program as part of the input information data structure.

Table 6-2 PDF mapping program input information structure

Field name	From position	To position	Type	Description
JOBNAM	1	26	CHAR(26)	Qualified job name
SPLFID	27	36	CHAR(10)	Spooled file name
SPLNO	37	40	BINARY(4)	Spooled file number
RTGTAG	41	290	CHAR(250)	Routing tag
PDFFILE	291	630	CHAR(340)	Path and name of PDF file
SVRTYPE	631	631	CHAR(1)	Mail server type
RES1	632	632	CHAR(1)	Reserved
PATHCCSID	633	636	BINARY(4)	Path and name CCSID
SENDER	637	646	CHAR(10)	Mail Sender
USRDTA	647	656	CHAR(10)	User data
SYSNAME	657	664	CHAR(8)	Job system name

Field name	From position	To position	Type	Description
TIMESTMP	655	672	CHAR(8)	Creation time stamp
OUTQ	673	682	CHAR(10)	Output queue on which the spooled file is located
OUTQLIB	683	692	CHAR(10)	Output queue library
RES5	693	712	CHAR(20)	Reserved
FORMTYPE	712	722	CHAR(10)	Formtype

The PDF mapping program can use any of this information to determine the ultimate disposition of the spooled file or spooled segment. In the sample PDF mapping program, we use the routing tag to look up the routing information. Other implementations can use other information, such as the spooled file name, job name, or user data, to determine the disposition of the spooled file or segment.

The Char(*) designation for the type of field for the input information indicates that the parameter can be any length. The actual length that Infoprint Server is using is passed to the PDF mapping program in the Length of input information parameter. It is up to the programmer to build a data structure that is long enough to receive all the information that Infoprint Server is providing in the input information data structure.

With V5R2 and the PTFs for intelligent routing, and in V5R3, all data in the PDF mapping input information is represented by fixed length fields. The total length is 722 characters.

Table 6-2 is a variation of the chart found in *Using Intelligent Routing*. We removed the Decimal and Hex Offset columns and inserted columns representing the field names used in the sample PDF mapping program and the begin and end positions of each field. This was done to calculate the absolute (positional) notation used in the sample RPGLE PDF mapping program.

See *Using Intelligent Routing*, or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, for full descriptions of the required parameters.

6.4.2 Length of input information

This field is passed from Infoprint Server to the PDF mapping program. It indicates how much input information it is supplying in the PDF mapping input information structure. For V5R2 with the intelligent routing PTFs and V5R3, this field has a value of 722.

6.4.3 PDF mapping output information

PDF mapping output information is also variable in length. Not all fields used are mandatory, and some of the fields are variable in length. This variability is handled in two ways:

- Infoprint Server tells the PDF mapping program how much room it has reserved in its buffer for the output information. This is the Length of output information buffer parameter. In many cases, this buffer is not long enough to suit the needs of the program. The PDF mapping program should respond by assigning the length that it requires for the output information buffer to the Length of output information available parameter.

  See 6.8, "How Infoprint Server and the PDF mapping program communicate" on page 92, for further explanation.

- Many of the fields and structures within the PDF mapping output information are referenced by other fields that specify their length and their location in the overall

structure. This is described in further detail in 6.5, “Structure of the output data” on page 68.

6.4.4 Length of output information buffer

This field is provided by Infoprint Server to the PDF mapping program to indicate to it how much buffer space is available for the output information. See 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92, for more information about how to use this value in your program.

6.4.5 Length of output information available

This field is passed from the PDF mapping program to Infoprint Server. It specifies the total length of the output information buffer. See 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92, to understand how Infoprint Server uses this value.

6.5 Structure of the output data

For V5R2, the fields used in the output information data structure are described in a combination of the following two locations:

- The original information that was available before the release of the intelligent routing PTFs is in *Infoprint Server for iSeries: User’s Guide*, G544-5775.
- New and changed information is found in the *Using Intelligent Routing* document. This information is not complete in that the sections that did not change were not included. Specifically, if you include files as body files, or as additional attachment, you must refer to the User’s Guide for information about the required structures.

For V5R3, you can find all the field information in *Infoprint Server for iSeries, User’s Guide Version 5, Release 3.0*, G544-5775.

The key to understanding the PDF mapping program lies in understanding the different types of information that you can placed in the PDF mapping program output information data structure. You can consider this in three parts:

- **Base structure**: This area contains fields that provide the minimal support for e-mail plus an indication of whether other types of distribution are requested.
- **Extension area**: This area contains the length and offset values for the other structures and fields.
- **Other formats and fields**: This area contains individual fields and formats that further define e-mail and the other distribution options.

6.5.1 Base structure

The first group of fields are at fixed offsets from the beginning of output information data structure of the PDF mapping program. Using these fields, you define the minimum information that is needed to specify that the file is to be sent by e-mail. You also specify if any of the other actions are to take place. However, the details for those actions and additional e-mail information are specified elsewhere.

Table 6-3 lists the fields in the base structure of the output information of the PDF mapping program. The information is based on the chart in *Using Intelligent Routing* or *Infoprint Server for iSeries, User’s Guide Version 5, Release 3.0*, G544-5775, with the following changes:

- Removed the Hex Offset column
- Added the field names used in the sample program

Table 6-3 reflects the actual length of the ADDRESS field that was used in the sample program, CHAR(255).

Table 6-3 Base structure

Decimal offset from beginning	Field name	Type	Description
0	DISPOSTN	CHAR(1)	Disposition of PDF e-mail
1	CALLAGIN	CHAR(1)	More processing
2	RES2	CHAR(2)	Reserved
4	MSGLEN	BINARY(4)	Length of message text
8	ADDRLEN	BINARY(4)	Length of mail address
12	MSGTEXT	CHAR(255)	Message Text
267	RES3	CHAR(1)	Reserved
268	EXTOFF	BINARY(4)	Offset to extension area
272	CCSID	BINARY(4)	CCSID of message text and subject
276	DSPPDFSTMF	CHAR(1)	Disposition of PDF Stream file
277	DSPPDFSPLF	CHAR(1)	Disposition of PDF Spooled file
278	DSPPDFERR	CHAR(1)	Disposition of PDF Error
279	DSPAFPSPLF	CHAR(1)	Disposition of AFPDS Spooled file
280	RES4	CHAR(7)	Reserved
287	ADDRESS	CHAR(255)	E-mail address

You can learn more about these fields in “Setting required and constant fields” on page 79 and in “E-mail fields in the base structure” on page 81.

6.5.2 Extension area

This group of fields consists entirely of fields that are used to specify the offset and length to other individual or groups of fields that further define the routing of the PDF file. The routing information itself is not contained in this structure. For example, if you want to create a PDF spooled file, you code the parameters, such as output queue and form type for the spooled file, elsewhere in the output information data structure of the PDF mapping program. In the extension area, you define the offset to that information and its length.

Table 6-4 lists the fields in the extension area format. The information is based on the chart in *Using Intelligent Routing Infoprint Server for iSeries, User’s Guide Version 5, Release 3.0*, G544-5775, with the following changes:

- Removed the Hex Offset column
- Added the field names used in the sample PDF mapping program
- Added the offsets from the beginning of the output Information structure of PDF mapping program, based on the assumption that the ADDRESS field is 255 characters long and the extension area begins immediately after at position 542.

The offsets were calculated using the spreadsheet as explained in 6.6, “Using a spreadsheet to calculate length and offset values” on page 72.

Table 6-4 Extension area format

Decimal offset from beginning of extension area	Decimal offset from beginning of output data structure	Field name	Type	Description
0	542	EXTLEN	BIN(4)	Length of Extension area format
4	546	SUBOFF	BIN(4)	Offset to subject
8	550	SUBLEN	BIN(4)	Length of subject
12	554	RPLYOFF	BIN(4)	Offset to Reply to e-mail address
16	558	RPLYLEN	BIN(4)	Length of Reply to e-mail address
20	562	CCOFF	BIN(4)	Offset to CC e-mail address
24	566	CCLEN	BIN(4)	Length of CC e-mail address
28	570	BCCOFF	BIN(4)	Offset to BCC e-mail address
32	574	BCCLEN	BIN(4)	Length of BCC e-mail address
36	578	BDYPTHOFF	BIN(4)	Offset to list of path names for body of e-mail
40	582	DIRPTHOFF	BIN(4)	Offset to path name for directory for files
44	586	DIRPTHLEN	BIN(4)	Length of path name for directory for files
48	590	ATTPTHOFF	BIN(4)	Offset to list of path names of attachments
52	594	STMFNAMOFF	BIN(4)	Offset to PDF stream file path and file name
56	598	STMFNAMLEN	BIN(4)	Length of PDF stream file path and file name
60	602	EATTNAMOFF	BIN(4)	Offset to file name for e-mailed PDF attachment
64	606	EATTNAMLEN	BIN(4)	Length of file name for e-mailed PDF attachment
68	610	PUBAUTOFF	BIN(4)	Offset to PDF file public authority
72	614	PUBAUTLEN	BIN(4)	Length of PDF file public authority
76	618	PDFSPLOFF	BIN(4)	Offset to spooled file PDF distribution
80	622	PDFSPLLEN	BIN(4)	Length of spooled file PDF distribution
84	626	AFPSPLOFF	BIN(4)	Offset to spooled file AFP distribution
88	630	AFPSPLLEN	BIN(4)	Length of spooled file AFP distribution
92	634	ENCRPTOFF	BIN(4)	Offset to PDF encryption information
96	638	ENCRPTLEN	BIN(4)	Length of PDF encryption information

6.5.3 All other formats and fields

The remaining information needed to define the routing of the spooled file is defined as individual fields in the PDF mapping output structure or as formats containing multiple fields. An example of an individual field is the name you want to use for the PDF file when it is attached to the e-mail, instead of 000001.pdf. An example of a format is the six fields needed

if you want to create a PDF spooled file. These six fields must be adjacent to each other in a given order.

Detailed descriptions of these formats and fields are provided in the various sections that describe the calculation specifications for each action.

6.5.4 Relation between the base, extension area, and other formats

Figure 6-1 illustrates the relationship between the base structure, the extension area format, the spooled file PDF distribution format, and a standalone field. These fields represent the case where we choose to send the PDF as an e-mail attachment with a new name and to generate a PDF spooled file. Not all fields are shown in the example.

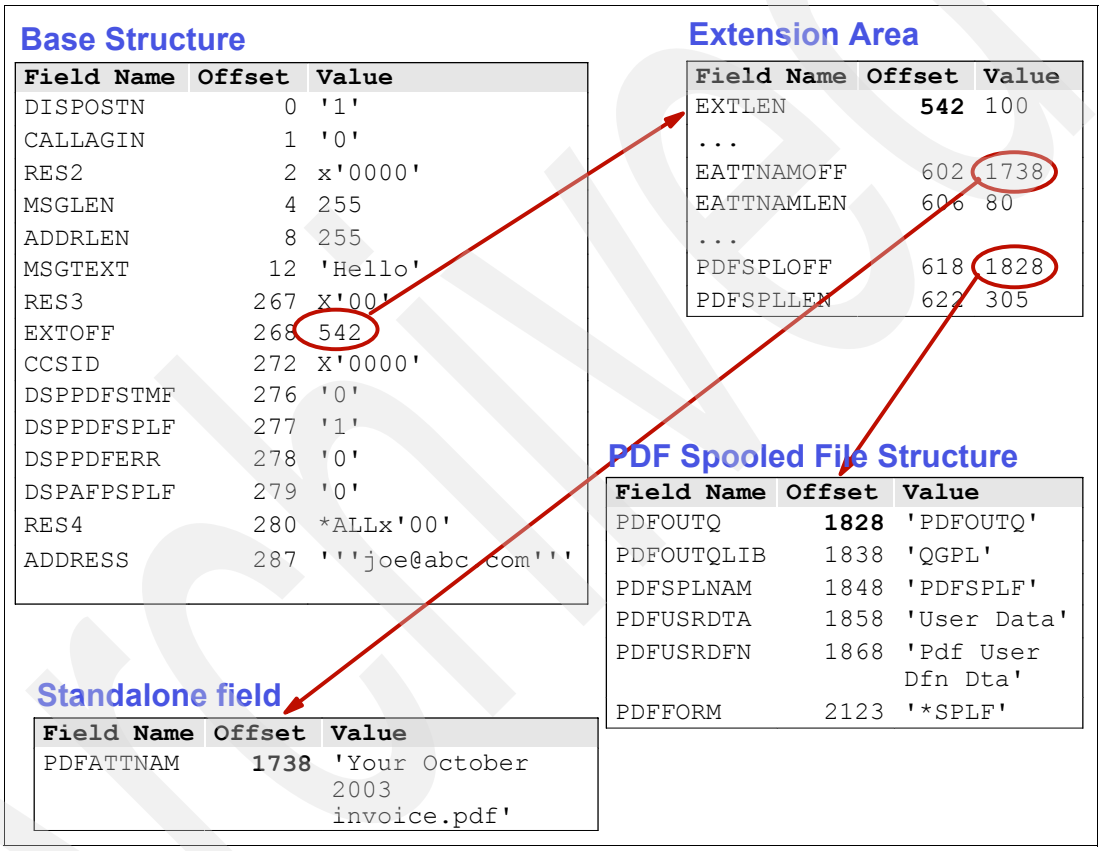


Figure 6-1 Relationship between fields within PDF mapping output information

The EXTOFF field in the base structure points to the beginning of the extension area format. We begin the extension area format at offset 542, immediately following the ADDRESS field, which has a length of 255 in this example.

The first field of the extension area defines its length. For V5R2 with the intelligent routing PTFs or V5R3, you specify a length of 100. The new name of the PDF attachment is at offset 1738 of the PDF mapping output information data structure. The field length is 80 characters. This is reflected by EATTNAMOFF and EATTNAMLEN. Another pair of fields, PDFSPLOFF and PDFSPLLEN, are used to specify the offset and length for the spooled file PDF distribution format, which consists of six fields used to define the PDF spooled file. All six fields must be presented in sequence using predefined lengths.

6.6 Using a spreadsheet to calculate length and offset values

The sample PDF mapping program used in developing this document has over 100 fields defined for the output information of the PDF mapping program. This number varies from installation to installation. However, if you require the flexibility to produce your PDF files in a variety of formats, you may find that you have a similarly large number of fields. It was found that a spreadsheet became an invaluable tool to calculate the offset values and length of the various fields and formats used in the output data structure.

You can find the entire spreadsheet in “Spreadsheet for mapping program calculations” on page 171. A copy of the spreadsheet is available for download as explained in Appendix I, “Additional material” on page 193.

Note that the colors and shading represent different categories of fields:

- **Black-on-gray**: Required value; do not change.
- **Blue-medium**: Calculated value based on the content of other fields; do not change.
- **Red-bold**: User variables; you can make changes to only these fields.

The spreadsheet includes the following columns:

- **Description**: The description as found in *Using Intelligent Routing* and *Infoprint Server for iSeries, User’s Guide Version 5, Release 3.0*, G544-5775.
- **Offset within section**: If a field is in a predefined segment of the output information data structure of the PDF mapping program, the offset from the beginning of that section is included here. For example, SUBOFF is always at offset 4 from the beginning of the extension area, regardless of where the extension area begins. Do not change these values.
- **Final offset**: This is the offset from the beginning of the output information data structure of the PDF mapping program. For example, if the extension area begins at offset 542, SUBOFF is at final offset 546. Do not change these fields. They are calculated based on the position and length of the previous field.
- **Type**: As defined in the documentation. All fields are either Character or Binary.
- **Allowance or Length**: For fields that have a fixed length in the documentation, that length is used here. If the length of the field is listed as variable, CHAR(*), the value used here is the amount of space to reserve for the field. For example, on a given instance the e-mail ADDRESS can only be 20 characters long. However, if you want to allow for an ADDRESS of up to 255 without changing your PDF mapping program, use 255 as the length. You can change any of the variable length fields that are in a bold red color. Do not change any of the predefined fields that are in black on gray or the calculated values that are in a medium blue color.

  Think of this field as the length of the allowance for a given variable length character field.
- **Start position**: This field is used for reference. It is always calculated as Final Offset plus one. Do not change any of these fields.
- **End position**: This field is used for reference. It is always calculated as Start Position plus length minus one. Do not change any of these fields.
- **Sample value**: These can be divided into a few categories.
  - *Reserved fields*: Must be coded as all X’00’ as indicated by the black-on-gray fields. Some length fields are also predefined and shown in black on gray.
  - *Length and offset fields*: These are calculated based on the position and length of the referenced fields. These are in blue (not bold or shaded) and should not be changed.

In some cases, the length field can have the value equal to the overall size of a variable length character field. In other cases, the length value used in the extension area must match the corresponding string length. The single quotation marks are not counted. For example, the length of the name for the body file 'ascii.txt' is calculated using either of the following methods:

- Lotus 1-2-3®: `@LENGTH(I71)-2`
- Microsoft Excel: `=LEN(I71)-2`

This results in the correct value of 9.

– *Variable fields*: Most of these are character fields. Since RPGLE requires that character fields be delimited by single quotation marks, those quotation marks are included in the spreadsheet as a reminder.

- **Field name**: This is the name that is used in the sample PDF mapping program.

The sample spreadsheet has allowances for all of the fields and formats that are used for the different types of distribution of the PDF files or segments. There is no harm in omitting unused fields or structures in a given implementation. This decision is up to the individual programmer.

All variable length character fields are given an allowance of 80, except for the e-mail address, which has an allowance for a length of 255. You can change any of the length fields that are in red. When you make such a change, watch how the other calculated blue fields that follow are adjusted.

We make allowances to include up to three body files within the e-mail and up to three attachments. In this example, even though we don't use the second and third attachment, the field DIRPTH is still placed 308 spaces past the beginning of the attachment format. The attachment format begins at offset 1270. DIRPTH is at offset 1578.

If you need more than three body files or three attachment files, you must insert the corresponding format immediately following the first three of the same type of format. You must manually adjust the equations used for BDYLEN, BDYNUM, ATTLEN, and ATTNUM.

After you complete any changes to the spreadsheet that you require for your implementation, use the length and offset values that are calculated in your version of the PDF mapping program instead of the values that you see in the sample PDF mapping program included with this document.

6.7 Sample program description

This section describes the coding used to in the sample PDF mapping program. For a complete listing of the PDF mapping program, see "Sample PDF mapping program in RPGLE" on page 164.

6.7.1 General comments

This sample program was written in RPGLE. Other programming languages can be used, as long as it can accept the information that is provided in the input information data structure and then generate the output in the correct format as determined by the output information data structure of the PDF mapping program.

Depending on the functions that a given client chooses to implement using intelligent routing, the PDF mapping program can become fairly lengthy. To help you understand the various concepts, we found that it helps to break it up into smaller components. In this chapter, you

see small segments of the sample program that pertain to the discussion points within a given section. You can find a full listing of the sample PDF mapping program in “Sample PDF mapping program in RPGLE” on page 164.

Programming techniques vary from individual to individual. The sample PDF mapping program is intended as a guide for developers. It may not cover the business needs of all users. Many values for lengths and offsets are hard coded based on the calculations that were done using a spreadsheet. These same calculations work equally well if done as part of the program. This may be a preferred method for a user's production program. However, for the sake of this document, we felt that it is easier for you to understand the program if the offset and length values were hard coded wherever possible.

Note: It is the responsibility of each programmer to customize their own PDF mapping program to suit their own specific needs.

The sample PDF mapping program includes corrections and enhancements to the programs that were published in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, and in *Infoprint Server for iSeries, User's Guide Version 5, Release 2.0,* G544-5775.

6.7.2 Program scenario and overview

The customer scenario and sample input spooled file used by the sample program is based on the sample used in Chapter 14, “End-to-end example” found in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250. The spooled file contains four separate invoices that the Super Sun Seeds is routing to different destinations. The overall process to generate the Super Sun Seeds invoice spooled file for use with intelligent routing is shown in Figure 6-2.

The process shown is outlined in the following sequence. Note that the numbered steps correspond to those in Figure 6-2:

1. The spooled file is generated as DEVTYPE(*LINE). It uses a form definition and page definition that were created using Infoprint Designer for iSeries.
2. The Create AFP Data (CRTAFPDTA) command is run against the spooled file. This adds tag fields for each separate invoice. The output from this step is a stream file that is routed to a physical file member.
3. The Print AFP Data (PRTAFPDTA) command is run against the physical file member. The new spooled file is now in *AFPDS format and contains routing information in the index tags.
4. The spooled file is placed in an output queue associated with a device description and PSF configuration object that are set up to use the intelligent routing PDF mapping program.
5. The PDF mapping program uses the information in the tag fields to determine how the individual segments from the spooled file are routed. In the earlier example, all valid invoices were directed to e-mail. In this new version, some invoices are directed to other destination types.

In an individual user environment, routing tags may be added using DDS keywords or the AFP Toolbox application programming interfaces (APIs) instead of using the CRTAFPDTA command. This bypasses steps 1 through 3. These other methods are described in *Infoprint Server for iSeries: User's Guide*, G544-5775, and *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775.

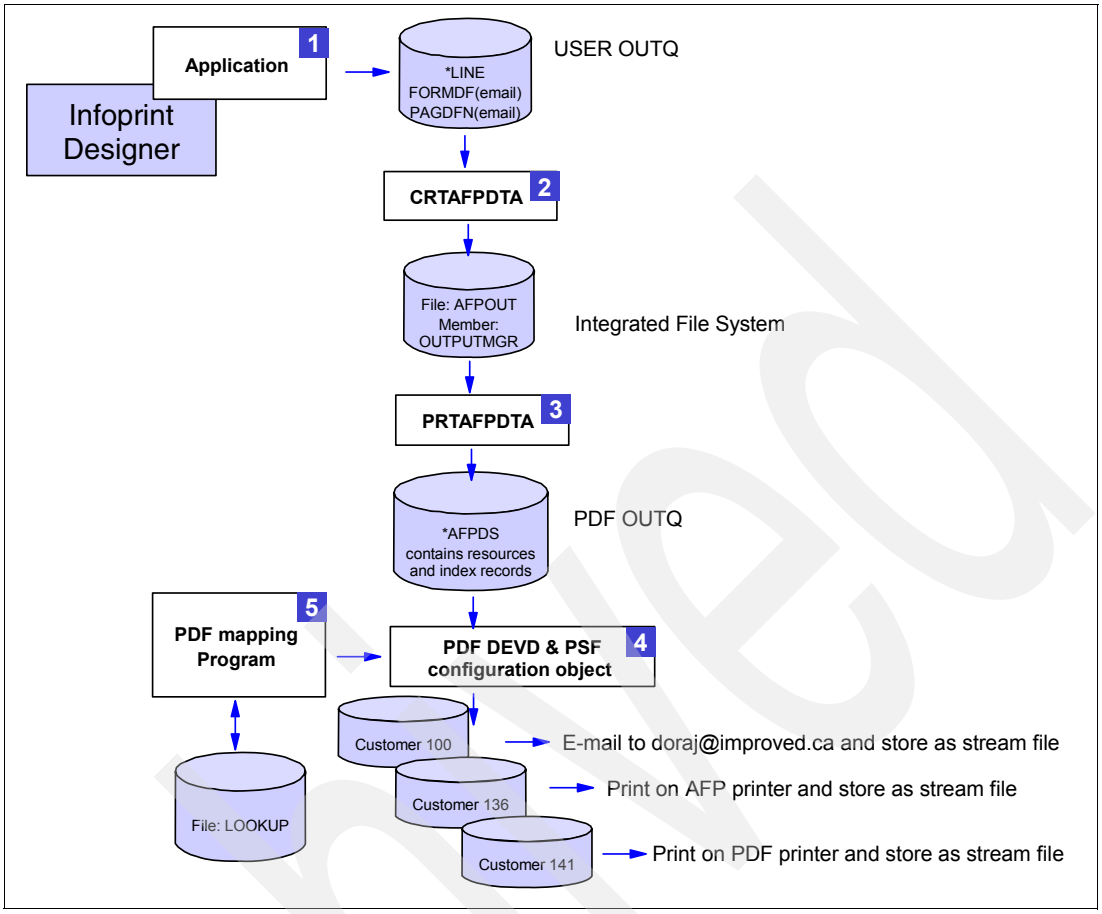


Figure 6-2 Generating spooled file for intelligent routing

6.7.3 Building routing tags using the CRTAFPDTA command

Routing tags are inserted into the spooled file using the CRTAFPDTA command. In the sample created for the purposes of this document, we changed the information included in the routing tags from the example used in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250. Super Sun Seeds now wants to store copies of all invoices in their IFS based on the invoice number. The CRTAFPDTA command was changed so that it builds the routing tag based on the customer number plus the invoice number:

```
CRTAFPDTA FROMSPLF(EMINVSCS) JOB(*) SPLNBR(*LAST)
  FORMDF(MIRA/EMBASI) PAGDFN(MIRA/EMBASI) TOIDXSTMF(*DFT)
  TORSCSTMF(*DFT) IDXTRG((* 3 'NEW')) IDXTAGFLD((4 15 6) (4 34 6))
  IDXTAG(('routing data' (*IDXTAGFLD1 *IDXTAGFLD2))) IDXGRPNAM(*NO)
  RSCDTA(*FORMDF) TOMRGSTMF(*DFT)
```

The significant differences from the CRTAFPDTA command used in the *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, are:

- A second index tag field, (4 34 6) was defined to pick up the invoice number from position 34 of record 6, for a length of 6.
- The new index tag field was added to the index tag by specifying *IDXTAGFLD2 as part of IDXTAG.

- In the IBM Redbook sample, eight-digit sequence numbers were automatically added to the group name. This example avoids this in the data that is created by specifying IDXGRPNAM(*NO).
- A slightly different form definition and page definition are used.

Figure 6-3 shows a sample page of the spooled file with the index trigger and index tag fields highlighted.

```
                        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 6-3 Spooled file showing trigger and index tags

6.7.4 File specifications

This section describes the file specifications of the sample PDF mapping program. One file is used to look up the distribution information.

For testing purposes, optional log files were added to capture copies of the input and output data structures. This is described in 6.10, “Logging the input and output data structures” on page 97.

The file specification for the lookup file is found on line 1 of the sample PDF mapping program.

LOOKUP is simple keyed physical file. The K in position 34 indicates that the file is to be processed by Key. The EXTFILE keyword allows us to specify the exact path for the physical file.

QSPLJOB must have *USE authority to the lookup file.

The file is based on the physical file used in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, with the addition of a few extra fields to specify destinations other than e-mail.

You can find the DDS for the LOOKUP file in “Data description specifications for LOOKUP file” on page 171.

Table 6-5 shows the sample data used for testing. Super Sun Seeds company is storing a copy of all invoices in their IFS on their iSeries server. Improved Printing Corp has requested that two e-mails be sent, one to Dora Jackson and one to her backup. The Invoices for Organic Garden Supplies are printed on Super Sun Seeds' production AFP printer. Los Arboles is a preferred customer. Their invoice is printed in color on a PDF printer. The last invoice in the spooled file is for customer 154. We purposely created the record for The Last Leaf in error to demonstrate the use of the new PDF administrator capability.

The company name and contact names are used to build a message that becomes part of the e-mailed document. These fields are ignored if no e-mail is sent.

Table 6-5 Sample data used for testing program

Cust number	Company name	Contact	E-mail address	Send as e-mail	Store in IFS	Print as PDF	Print as AFP
100	Improved Printing Corp	Dora Jackson	‘doraj@improved.ca’ ‘backup@improved.ca’	Y	Y	N	N
136	Organic Garden Supplies	Erin Paull	‘epaull@organicgarden.uk’	N	Y	N	Y
141	Los Arboles Del Mundo	Amy Lyndhurst	‘amyl@los_arboles.sp’	N	Y	Y	N
155	The Last Leaf	Mark Allen	‘mark.allen@last_leaf.com’	Y	Y	Y	Y

Important: Each e-mail address must be delimited separately by single quotation marks.

Example 6-1 illustrates the data as it resides in the lookup file.

Example 6-1 Sample data in the lookup file

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0./....+....4
  100Improved Printing Corp   Dora Jackson            'doraj@improved.ca''backup@improved.ca'      \      YYNN
  136Organic Garden Supplies  Erin Paull              'epaull@organicgarden.uk'                    /      NYNY
  141Los Arboles Del Mundo    Amy Lyndhurst           'amyl@los_arboles.sp'                        \      NYYN
  155The Last Leaf            Mark Allen              'mark.allen@last_leaf.com'                   /      YYYY
                      ****** END OF DATA ******
```

Note the following points:

- Improved Printing Corp receives two copies of the e-mail. Note that each e-mail address is delimited by single quotation marks, with no spaces between.
- The company number field is six characters long. Only three digits are used for these customers. The fields is right aligned with spaces in the first three positions.
- The correct customer number for The Last Leaf is 154. We entered an incorrect number into the Lookup file to test the error handling processes.

6.7.5 Definition specifications

The RPGLE definition specifications are used to define the field types and lengths used in the input information data structure and output information data structure of the PDF mapping program, plus a few other independent fields. For a complete listing off all fields, see “Sample PDF mapping program in RPGLE” on page 164. Many of the fields encoded in this section are binary fields. See 6.3.2, “Binary fields” on page 65, for more information.

Input information data structure

The PDF mapping program input information data structure, INPUTDS, is coded as a program described data structure using absolute (positional) notation. See 6.4.1, “PDF mapping program input information” on page 66, for more information about the PDF mapping program input information data structure.

Output information data structure

The PDF mapping program output information data structure, OUTDS, is coded using length notation. This was done to allow for changes to the lengths of any of the variable fields. If this is not done and you want to change the length of a field, such as ADDRESS, you must adjust all the to and from positions for all of the following fields.

When using length notation, it is important to keep the sequence of the fields intact, since the position of any given field depends on the positions and lengths of the preceding fields.

The binary fields are coded with a length of 9. The resulting data structure uses 4 bytes for these fields.

Stand-alone fields

The INPUTLEN, OUTPUTLEN, and OUTINFO fields are coded as stand-alone fields. These fields must be BINARY(4), so they are coded with a length of 9.

6.7.6 Calculation specifications

The steps to perform in the program’s calculation specifications are outlined here:

1. The program receives the input information from Infoprint Server.
2. The program determines whether the output buffer is long enough. If not, it returns processing back to Infoprint Server with a request for a longer buffer.
3. The buffer is cleared, and some required fields are set.
4. The program extracts the customer number from the routing tag. It does a lookup to the LOOKUP file based on the value of customer number.
5. If the LOOKUP fails, the program turns on an error condition and returns processing back to Infoprint Server. Unlike the version of the PDF mapping program in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, this version does not call an external program for error handling.
6. If the LOOKUP is successful, the program sets the appropriate fields in the PDF mapping program output information data structure and returns processing back to Infoprint Server.

Assigning values to variables

Most of the coding in the calculation specifications in the sample program is simply the assignment of values to fields. Therefore, it is worth spending a few moments reviewing the specific fields types and their requirements.

Character strings

In RPGLE, these strings are delimited by single quotation marks. Be aware that many fields, such as DISPOSTN, appear to be numeric in that the allowable values are ‘0’ or ‘1’. Despite appearances, these fields are designated as character fields. Make sure that these are coded correctly with quotation marks.

E-mail addresses

Fields used for the e-mail address, as well as the CC, BCC, and Reply to addresses, must have single quotation marks as part of the data. The resulting RPGLE code requires that you use three single quotation marks on each end of the field.

Numeric fields

Any numeric fields are assigned by using the standard decimal number, without any quotation marks. This holds true even for fields that are defined as binary.

Reserved fields

These fields must be set to hexadecimal 00s in all positions. For example, RES2 has a type of CHAR(2), so it must be assigned a value of X'0000'. An alternate method is to use the system supplied figurative constant *ALLX'00'.

Passing parameters to and from Infoprint Server

The five parameters used to pass data to and from Infoprint Server are defined in the *ENTRY PLIST at the beginning of the calculation specifications. These fields are described in 6.4, “PDF mapping program input and output parameters” on page 66. See lines 151 through 156.

Checking the length of the output buffer

The program first checks to see if the output buffer length specified by Infoprint Server is large enough to hold all of the data we want to pass to it. See lines 163 through 166.

Note: The length of OUTINFO used in this test is based on the ending position of the last field in the PDF mapping program output information data structure as calculated by the spreadsheet.

For more information about the logic used by Infoprint Server and the PDF mapping program to set the buffer length, see 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92.

Setting required and constant fields

Before you assign fields that are specific to one routing destination or another, you must set some fields that are independent of the PDF routing:

- We recommend that you assign hexadecimal '00' to the entire PDF mapping program output information data structure at the beginning of the program. This ensures that unused fields and reserved fields have a value of X'00's.
- The CALLAGIN field should almost always be set to a value of ‘0’. See 6.9, “More processing (CALLAGIN)” on page 96.
- The reserved fields must be set to strings of X'00's. This is done implicitly when the entire data structure is set to X'00's. However, for illustration purposes, we do this explicitly for each reserved field.
- The pointer to the extension area, EXTOFF, and the length of the extension area, EXTLEN, are defined. These values are based on the spreadsheet calculations.

- The PDF error code is set to ‘0’. It is changed only if a “not found” condition appears when looking up the routing information.

Lines 171 through 180 in the sample PDF mapping program show the RPGLE instructions used to set the required and constant fields.

Looking up routing information

The customer number is contained in the first six characters of the RTGTAG field. The RPGLE substring operation is used to extract it and place it in the CUSTNO field. The CHAIN command is used to retrieve the customer preferences record from the LOOKUP file. See lines 182 through 184.

Customer record not found

Immediately after the chain operation to the LOOKUP file, use the RPGLE built in function, %FOUND, to determine if a valid customer record has been found. If a “not found” condition occurs, set up the variables used to take advantage of the PDF administrator function of Infoprint Server. See 2.2, “Understanding mail sender, reply-to, and PDF administrator” on page 22, for more information.

After you set the values, the RETURN operation passes control back to Infoprint Server. No other distribution options are used for segments that fail the lookup operation. The ENDIF operation indicates the end of the block of statements used for these segments.

If the customer is found, the program sets the error condition, DSPPDFERR, to ‘0’ and continues to evaluate the various routing actions.

The RPGLE code used to check for an error in the lookup process is in lines 186 through 198.

Consider the following points when you use the PDF error function:

- The e-mail is sent using the default style shown in Figure 6-4. You cannot change the subject line, message text, or PDF file name. The PDF file name can help to find the original spooled file on the iSeries because it contains the job number, spooled file name and number, a date stamp, and in the case of segmented file, a sequence number indicating which segment was in error.
- The e-mail is only sent to the user defined in the PDFADMIN parameter in the PSF configuration object. If no PDF administrator is defined, this value is ignored. In either case, the spooled file is held and a message is logged in a job log for the PDJ job.
- Settings for CC, BCC, and Reply-to are ignored.
- The e-mail is sent by QSPLJOB, as opposed to the user specified in the PDFSENDER parameter of the PSF configuration object.
- Other action types, if selected, are ignored.
- The pointer to the extension area is set to 0. No fields beyond the base of the output information data structure are used.

Figure 6-4 shows an example of an e-mail sent to the PDF administrator. The recipient uses Lotus Notes to receive e-mail.

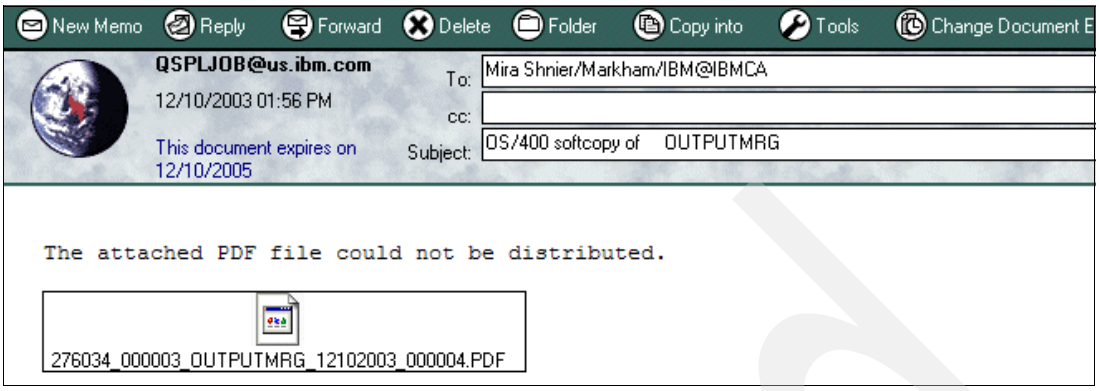


Figure 6-4 Sample e-mail generated by using Disposition of PDF Error

6.7.7 Calculation specifications for sending an e-mail

The program checks the customer preference for e-mail. If this is one of the desired distribution methods for the customer, several fields are available to customize the output. They are divided into three groups:

- E-mail definition fields in the base structure.
- Simple fields in the remaining area of the PDF mapping program output information data structure

  The length and offset to these fields are defined in the extension area. This includes fields used to define the subject, reply-to address, CC address, BCC address, and to rename the PDF attachment.
- Formats used to define body files and additional attachments, plus one stand-alone field used to define a default directory for body files and attachments

  The offset to the two formats and the offset and length for the stand-alone field is defined in the extension area.

E-mail fields in the base structure

Lines 202 through 213 of the sample PDF mapping program show calculation specifications used to define the fields from the base segment of the PDF mapping program output that are used for e-mail.

Consider the following fields:

- **DISPOSTN:** The e-mail disposition field for the PDF mapping program output information data structure is a character field that must be set to ‘0’ or ‘1’.
- **EADDR:** If the routing tag is in the correct format to use as an e-mail address, you must move it to the ADDRESS field in the output data structure. This does not happen automatically.
- **MSGLEN:** If you choose not to create a custom message, set the MSGLEN field to zero. If you do this, the system-supplied message `Mail was sent by OS/400 process. Please do not reply to this address` is used.

  This text is found in message PQT4133 in message file QPQMSGF, starting in position 23. You can modify this text. The modified text is used as the default for all e-mails that do not have a custom message.

- **MSGTEXT:** In this example, we use field information from the customer database file to enhance the message text field.

  Tip: Building a dynamic custom message based on variables is an example of a function available with the PDF mapping program that is not available with the PDF mapping object.

- **ADDRESS:** The e-mail addresses from the customer preference field, EADDR, is assigned to the output information data structure field, ADDRESS, of the PDF mapping program.
- **ADDRLEN and MSGLEN:** The values of the ADDRLEN and MSGLEN fields can be greater than or equal to the length of the strings in the corresponding ADDRESS or MSGTEXT fields. The maximum length for the message is 255. There is no imposed length for the e-mail address field. We used 255 in the sample PDF mapping program.
- **ADDRESS:** If you are coding the address or addresses directly in your program, use the syntax as described for CC and BCC, as illustrated in line 222 and 226 through 227 of the sample PDF mapping program.
- **ADDRESS:** Unlike the PDF mapping object, the PDF mapping program does not offer the option to use a separate stream file to name a group of address for the e-mail function. It is the programmer's responsibility to concatenate all the valid addresses into the one ADDRESS field.
- **CCSID:** Specify the CCSID if you are using an Simple Mail Transfer Protocol (SMTP) server. All other fields described in this section can be used for SMTP or *SNDDST.

Additional e-mail fields in the output information data structure

Several optional fields can be used to enhance the look and function of the e-mail. The Reply-to, CC, and BCC fields are used to provide additional addressing information. The ability to change the subject line and the name of the PDF attachment increases the understanding that the recipient has of what the content of the PDF is and subsequently the ability to act on it sooner.

The calculation specifications used for this step are found in lines 214 through 236.

Consider the following fields:

- **SUBOFF and SUBLEN**: If you choose not to use any of these options, set the corresponding offset and field length, such as SUBOFF and SUBLEN to 0.
- **EXTSUBJ**: If you do not specify a subject, the default system supplied subject is used. It is made up of the string found in positions 1 through 22 of message PQT4133 in message file QPQMSGF, plus the name of the original spooled file. You can modify the 22 characters of text. The name of the original spooled file continues to be displayed as part of the subject. The modified text is used as the default for all e-mails that do not have a custom subject defined.
- **EXTRPLY, EXTCC, and EXTBCC**: If you hard code an e-mail addresses for the Reply-to, CC, or BCC fields, you must delimit them by using three single quotation marks, as illustrated in the preceding example. You can specify multiple addresses for CC and BCC. Use this same syntax if you hard code multiple e-mail addresses in the ADDRESS field.
- **RPLYLEN, CCLEN, and BCCLEN**: The length field for the Subject, Reply-to, CC, and BCC addresses does not have to match the actual length of information passed in those fields. The maximum length for the subject is 256. There is no imposed length for the Reply-to, CC, and BCC address fields. We chose to use 80 for the length of each of these fields for the sample PDF mapping program.

- **RPLYLEN, CCLEN and BCCLEN**: The Reply-to, CC, and BCC addresses can only be used if you are using an SMTP mail server.
- **EATTNAMLEN**: There is no imposed maximum for the length of the name of the attachment. A maximum length of 80 is allowed in the output information data structure.
- **PDFATTNAM**: In the sample program, we build the name of the PDF attachment by concatenating some constant text plus the invoice number. The invoice number is obtained by performing a substring operation against the RTGTAG field and extracting five characters starting in position 7.

> **Tip:** Building a dynamic PDF file name based on variables is an example of a function that is available with the PDF mapping program that is not available with the PDF mapping object.

- **PDFATTNAM**: Make sure you add the ".pdf" string to the name of the PDF attachment. If you do not, the recipient may have difficulties opening the file.
- **PDFATTNAM**: If you are using *SNDDST, you can change the subject or the name of the PDF attachment. The PDF attachment is stored temporarily in QDLS if you use *SNDDST. Consequently, you must use the QDLS naming convention, nnnnnnnn.ext, if you choose to rename the PDF attachment. This allows for up to eight characters followed by an optional period and an optional extension of up to three characters.

Setting encryption fields

In the case of the Super Sun Seeds sample, we assign a master password to the file, but no user password. This allows any user to open the file and read it, but they cannot change the security settings without knowing the master password. An encryption of 128 bits is used. The user can print the PDF file and make comments (annotations), but they cannot change the document, copy it, or perform document assembly. The file is assigned content access for the visually impaired. For a further clarification about what these various permissions allow the user to do, see Chapter 4, "Encryption for iSeries-generated e-mail" on page 41.

Table 6-6 lists the fields used for encryption. It is based on the charts in *Using Intelligent Routing* or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, with these changes:

- Removed the Hex Offset column
- Added the field names used in the sample PDF mapping program
- Added the offsets from the beginning of the PDF mapping program output information structure based on the values calculated using the spreadsheet

Table 6-6 PDF encryption format

Decimal offset from beginning of encryption area	Decimal offset from beginning of output data structure	Field name	Type	Description
0	2438	PDFMASTPW	CHAR(32)	PDF master password
32	2470	PDFUSRPW	CHAR(32)	PDF user password
64	2502	PDFPRT	CHAR(1)	PDF print
65	2503	PDFDOCCHG	CHAR(1)	PDF document change
66	2504	PDFCOPY	CHAR(1)	PDF copy
67	2505	PDFENCLVL	CHAR(1)	PDF encryption level
68	2506	PDFCNTACC	CHAR(1)	PDF content access enablement

Decimal offset from beginning of encryption area	Decimal offset from beginning of output data structure	Field name	Type	Description
69	2507	PDFCHGCMT	CHAR(1)	PDF change comments
70	2508	PDFDOCASB	CHAR(1)	PDF document assembly

For complete field descriptions, refer to the *Using Intelligent Routing* document. You can find the RPGLE instructions used to define encryption in lines 238 through 250 in the sample PDF mapping program.

Consider the following additional points when programming for encryption:

- **ENCRYPTOFF:** The encryption offset varies from program to program depending on the other fields that are used in the output information data structure of the PDF mapping program. In the case of the sample PDF mapping program, the value was calculated using the spreadsheet.
- **ENCRYPTLEN:** The length of the encryption format is 71. This is the sum of the lengths of the nine fields used to define encryption.
- **ENCRYPTOFF and ENCRYPTLEN:** If encryption is not to be used, you must specify a value of 0 for ENCRYPTOFF and for ENCRYPTLEN.
- **PDFMASTPW and PDFUSRPW:** The PDF master password and PDF user password, if used, must be padded with blanks, X'40', or null characters, X'00', to fill the 32 characters allowed. This is done implicitly in the sample PDF mapping program when the entire data structure is set to all X'00's. Each of these strings can be up 32 characters in length.
- **PDFMASTPW and PDFUSRPW:** The PDF master password and PDF user password fields are case sensitive.
- The remaining fields are character fields with a length of 1.

Fields used to define body files and additional attachments

With the release of V5R2, you can add additional files to an e-mail. You can include files that are text (file type .txt) or Hypertext Markup Language (HTML) (file type .htm or .html) as body files. These means that with most browsers, they appear in the body portion of e-mail. Other file types, such as spreadsheet, images, sound files, or word processing documents can be added to the e-mail as additional attachments.

See *Infoprint Server for iSeries: User's Guide*, G544-5775, or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, for further information about body files and additional attachments.

The format of the received e-mail depends on the e-mail client being used by the recipient. We had the following results when testing:

- Lotus Notes, Microsoft Outlook Express 6, Microsoft Hotmail, and Netscape Communicator: Text files and HTML files appeared as body files as expected. In addition, pictures in the form of JPEG files appeared as part of the body when specified as body file, as shown in Figure 6-4 on page 81.
- Microsoft Outlook: All body files were delivered as attachments.

The overall structure for defining each type of file, body, or attachment is the same. The format is actually a format within a format. The outer format, known as the *stream file format*, defines the overall length of the format and the number of subformats that follow. Immediately following the stream file format, you have formats for each individual body or attachment file. If

you specify stream files for the body and as additional attachments, use a separate stream file format for each type.

Table 6-7 and Table 6-8 list the fields used to define a stream file format that is used for body files plus the individual file format for the first body file. It is based on the charts in *Infoprint Server for iSeries: User's Guide*, G544-5775, or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, with the following changes:

- Removed the Hex Offset column.
- Added the field names used in the sample program for the first body file.
- Added values used for the first body file for the offsets from beginning of the output information structure of the PDF mapping program. These values are based on the values calculated using the spreadsheet.
- These charts reflect the actual length of the BDYPTH1 field that was used in the sample program, CHAR(80).

Table 6-7 Stream file format

Decimal offset from beginning of stream file format	Decimal offset from beginning of output data structure	Field name	Type	Description
0	962	BDYLEN	BIN(4)	Total length of stream file information for body
4	966	BDYNUM	BIN(4)	Number of stream files specified

Table 6-8 Individual stream file format

Decimal Offset from beginning individual file format	Decimal offset from beginning of output data structure	Field name	Type	Description
0	970	BDYX1	BIN(4)	Total length of individual stream file format
4	974	BDYY1	BIN(4)	Length of individual stream file format
8	978	BDYOFF1	BIN(4)	Offset to path name
12	982	BDYPLEN1	BIN(4)	Length of path name
16	986	BDYUSE1	CHAR(1)	Use specified directory
17	987	BDYRES1	CHAR(3)	Reserved
20	990	BDYPTH1	CHAR(*)	Path name of body file

The Path name of body file BDYPTH1 is not part of the stream file format. It is included here because it makes sense to include it in the output structure of the PDF mapping program immediately following the other fields used to define the body file. The stand-alone field, DIRPTH, is optional. If most of the attachments are located in one directory path, it simplifies the construction of the individual file names. For a complete listing of the calculation specifications required for up to three body files and three attachments, see lines 252 through 315 of the sample PDF mapping program.

Note the following fields:

- **DIRPTH:** The directory for body and attachment files is used for body or attachment files that have the field for Use specified directory, such as BDYUSE1 set to '1'. This field is

ignored if nnnUSEn is set to '0'. In the sample PDF mapping program, most files reside in the /most directory in the iSeries IFS.

- **DIRPTHOFF:** The value for directory path offset was calculated using the spreadsheet.
- **BDYPTHOFF and ATTPTHOFF:** The offsets to list of body paths and attachments are calculated using the spreadsheet.
- **DIRPTHLEN:** There is no imposed limit on the length of the names for the body files or attachment files. We chose a length of 80 for DIRPTHLEN, which is more than sufficient for our needs. The length of this field does not have to match the actual length of the string used for the path name.
- **DIRPTHLEN:** We recommend that this length be divisible by 4. This ensures that the total length of the individual stream file format, BDYX1, is divisible by 4, which is required.
- **BDYLEN:** The total length of the stream file information for the body files is the sum of the of the lengths of BDYLEN, BDYNUM, plus the lengths of all three body files' formats. In the case of the sample PDF mapping program, the total is 4 + 4 + 100 + 100 + 100 = 308.
- **BDYX1:** The total length of the individual stream file format equals the sum of the six fields that are part of the format, plus the allowance for the stream file name. Infoprint Server uses this value to determine where the next stream file information format begins. In the sample PDF mapping program, the format has a length of 20. We allow up to 80 bytes for the path name. This gives a value of 100. If you choose a longer or shorter allowance for the path file name, that is reflected in this field.
- **BDYY1:** The length of the individual stream file format is equal to the sum of the six fields that are part of the format alone. This value must be 20.
- **BDYOFF1:** The offset to path name is based on the beginning of the individual stream file format. In the sample program, the path name is placed immediately following the end of the individual stream file information format, so this value is 20.
- **BDYPLEN1:** The length of path name must exactly match the length of string used in the path name. The RPGLE built-in function, %len, is used to calculate the path length. If you calculate this value manually, do not count the single quotation marks.
- **BDYUSE1:** The '1' for BDYUSE1 indicates that the first body file, sample.htm, resides in the /most directory.
- **BDYPTH2 and BDYPTH2:** The second body file does not reside in the /most directory. Consequently, BDYPTH2 includes the full path name, and BDYUSE2 is set to '0'. The third body file, ascii.txt, and the one attachment file, prices.123, reside in the /most directory.
- According to *Infoprint Server for iSeries: User's Guide*, G544-5775, and *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775, the second body file, iris.jpg, should not be included as part of the body files. It should be defined as an attachment. These documents state that if the body file is not a text file (.txt) or an HTML file (.htm or .html), it is delivered as an attachment. We found initially that the picture of the iris was resolved in the body when using some mail programs.

  This is a function of Multipurpose Internet Mail Extension (MIME) encoding and the receiver's e-mail program. Infoprint Server builds the MIME header with the attachment and the body file information with different MIME structures for each. The e-mail program decided if the type should be rendered or attached.

  Important: SMTP MIME is case sensitive. Infoprint Server inserts the file name as it is supplied by the mapping program. Many e-mail client programs properly print the body file in the e-mail if the extension is .txt or .html, but not if it's .TXT or .HTML

- **ATTNUMB:** The sample program only uses one attachment file, although the code allows for up to three. If you want to use two or three attachments, change the value of ATTNUMB and add the appropriate information for the other files. Use the spreadsheet as a guide.
- If you want to add four or more body files or attachments, you may want to start by making the changes in the spreadsheet. This makes it easier to calculate the offset values for any formats or fields that are placed in the PDF mapping program output information structure after the body and attachment files definitions.
- No limit is imposed by Infoprint Server regarding the number and size of body files and attachments that can be included in an e-mail when you use a PDF mapping program. However, consider the band width for the recipient's connection to the Internet or limits that may be imposed by their Internet service provider (ISP).
- You can only use body files and attachments if you are using an SMTP mail server.

Figure 6-5 shows e-mail generated from the sample program and received using Lotus Notes.

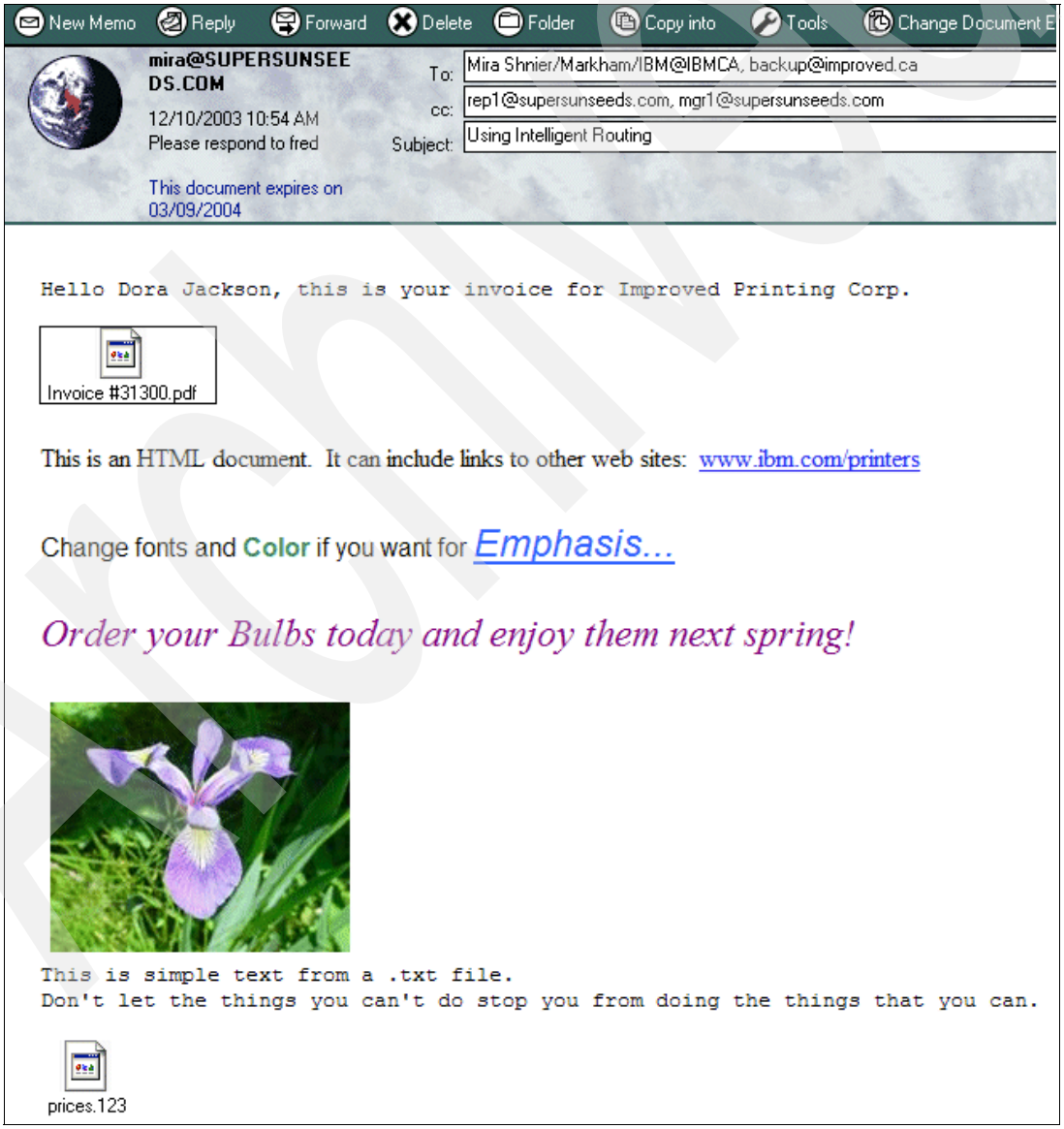


Figure 6-5 Sample e-mail including message, body files, and additional attachments

Not sending an e-mail

In the case where the value for DISPEMAIL is not 'Y', processing is transferred to the ELSE section of the logic. It is important that some fields are set correctly if an e-mail is not to be sent.

Lines 317 through 326 of the sample PDF mapping program contain the logic used if an e-mail is not to be sent.

Note the following points:

- **DISPOSTN and ADDRLEN:** It is not sufficient to set the e-mail disposition field, DISPOSTN, to `'0'`. You must also set the address length, ADDRLEN, to `'0'`. It is incorrect to specify don't mail and then provide e-mail address information.
- **MSGLEN, MSGTEXT and ADDRESS:** We recommend that you set MSGLEN to `'0'` and clear the MSGTEXT and ADDRESS fields.

6.7.8 Calculations specifications for storing the file as a stream file

The next block of code in the sample PDF mapping program illustrates the variables that must be assigned based on your decision to store the spooled file or segment as a PDF stream file in the Integrated file system. Only two stand-alone fields are specified when storing the PDF file as a stream file in the IFS. No multi-field formats are used for this action.

- PDF stream file path and file name, PDFSTMFNAM
- PDF file public authority, PDFPUBAUT

For detailed descriptions about these fields, see "Extension Area Format" in *Using Intelligent Routing* or *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775.

Along with these two fields, you must specify the correct values for the corresponding length and offset fields that are defined in the extension area.

Lines 328 through 349 of the sample PDF mapping program illustrate the calculations specification used to define how the PDF is stored as a stream file.

Note the following points:

- **STMFNAMOFF and PUBAUTOFF:** The offsets to the stream file name and public authority were calculated using the spreadsheet.
- **PUBAUTOFF and PUBAUTLEN:** If you want to use the default public authority for the stream file, *EXCLUDE, specify a value of 0 for both PUBAUTOFF and PUBAUTLEN.
- **PUBAUTLEN:** The values specified for the length of the public authority of the stream file must match the length of the string being used. In the sample PDF mapping program, the RPGLE built-in function, %len, is used to calculate the length. If you calculate it manually, do not include the single quotation marks. For example, '*R' has a length of 2.
- **STMFNAMLEN:** The value specified for the Length of the stream file path and name parameter does not have to be equal to the actual string length.
- **STMFNAMLEN:** There is no specified limit to the length of the PDF stream file path and file name.
- If the directories in the path do not already exist, they are created by Infoprint Server. The owner of any directories created by the transform is be QSPLJOB, and their public authority is *RX.
- QSPLJOB needs *WX data authorities to the folders in the path if they exist.

- The PDF stream file object owner is the owner of the original spooled file. The object authority is the value specified on the Authority parameter for the PDF stream file. *EXCLUDE is the default. QSPLJOB has ALLOBJ authority to the object.
- **PDFSTMFNAM:** If the path name is blank or invalid, the file is not stored. If a user is specified for PDFADMIN in the PSF configuration object, an e-mail is sent to that user.

  Remember to give the PDF file name a .pdf extension. Otherwise the recipient may have difficulty opening it.

  Important: If a file exists in the same path under the same name, it is overwritten without any warning. It is the programmer's responsibility to ensure that unique names are used.

- **PDFSTMFNAM:** To illustrate the flexibility that the PDF mapping program offers, the stream file name and path are built by concatenating some constant text with the invoice number that is extracted from the routing tag. This value was originally part of the print data and was captured using the CRTAFPDTA command. See 6.7.3, "Building routing tags using the CRTAFPDTA command" on page 75, for details on how the CRTAFPDTA command was used to build the routing tags.
- **STMFNAMOFF:** If you want to use the default path and name for the stream file, specify '0' for Offset to stream file path and name. The stream files that are generated follow the naming convention outlined in *Infoprint Server for iSeries: User's Guide*, G544-5775, and *Infoprint Server for iSeries, User's Guide Version 5, Release 3.0*, G544-5775.

  Restriction: With V5R2, there are limitations to the combinations of parameters that you can select when creating or changing a PSF configuration object. Consequently, you can only use specify that you want to use default name for a stream file in a mapping program if your e-mail delivery is limited to PDFMAILSVR(*SNDDST). See 2.1.1, "Creating a PSF configuration object" on page 12, for details. This restriction does not exist for V5R3.

- **DSPPDFSTMF:** If you choose not to store the PDF file as a stream file, specify the character '0' for the stream file disposition. You must also specify a value of '0' for both STMFNAMLEN and PUBAUTLEN. In the sample program, the specifications for setting SMFNAMLEN and PUBAUTLEN equal to '0' are redundant. This is because the entire PDF mapping program output structure is initialized to *ALLx'00' near the beginning of the program. However, the calculation specifications are included here for illustration purposes.

6.7.9 Calculation specifications for generating a PDF spooled file

Prior to installing the intelligent routing PTFs, if you split a large Advanced Function Presentation data stream (AFPDS) spooled file into multiple small PDF spooled files, the entries for each of the PDF spooled files in the destination output queue were virtually indistinguishable. With intelligent routing, you can use the PDF mapping program to customize the spooled file name, user data, and form type to suit your needs. For example, you can assign the invoice number to the user data field.

The six fields used to customize that type of output are contained in the spooled file PDF distribution format.

Table 6-9 lists the fields needed to generate a PDF spooled file. It is based on the chart in the *Using Intelligent Routing* with the following changes:

- Removed the Hex Offset column.
- Added the field names used in the sample program.
- Added values for the offsets from beginning of the PDF mapping program output information structure. These values are calculated using the spreadsheet.

Table 6-9 Spooled file PDF distribution format

Decimal offset from beginning of PDF spooled file format	Decimal offset from beginning of output data structure	Field name	Type	Description
0	1828	PDFOUTQ	CHAR(10)	Name of output queue on which to spool PDF
10	1838	PDFOUTQLIB	CHAR(10)	Output queue library
20	1848	PDFSPLNAM	CHAR(10)	Spooled file name
30	1858	PDFUSRDTA	CHAR(10)	User data
40	1868	PDFUSRDFN	CHAR(255)	User defined data
295	2123	PDFFORM	CHAR(10)	Form type

All fields in this format are fixed length fields.

Lines 351 through 374 of the sample PDF mapping program illustrate the calculations specification used to define how the selected spooled file or segment is generated as a new PDF spooled file.

Consider the following:

- **PDFSPLOFF** and **PDFSPLLEN:** The values for Offset to spooled file PDF distribution and Length of spooled file PDF distribution were calculated using the spreadsheet.
- You can use uppercase or lowercase for any of the previous character fields. One exception is the specification of the reserved words *SPLF or *PSFCFG, which must be specified in uppercase. The output queue, library, and spooled file name are automatically converted to uppercase. User data, user-defined data, and form type remain in the case used in the PDF mapping program.
- **PDFSPLNAM:** If you leave the spooled file name blank, the name of the original spooled file is used. If you specify an invalid spooled file name, Infoprint Server renames the file to QSYSPRT.
- **PDFFORM:** In the example, we used a value of *SPLF for form type for PDF spooled file. This tells Infoprint Server to use the form type of the original spooled file. In a similar manner, you can specify *SPLF for PDFSPLNAM, PDFUSRDTA, and PDFUSRDFN.
- **PDFOUTQ and PDFOUTQLIB:** If the output queue does not exist, Infoprint Server places the spooled file in the QPRINT output queue in library QGPL. A message is logged in the job log for the PDJ job. The original spooled file is not held.
- **PDFOUTQLIB:** If you specify *LIBL for the output queue library name, Infoprint Server searches the library list of the PDF job. *LIBL, if used, must be specified in uppercase.
- **PDFOUTQ:** If you want to store the spooled file in the default output queue as specified in the PSF configuration object, specify *PSFCFG for the value for the PDFOUTQ.

Restriction: With V5R2 there are limitations to the combination of parameters that you can select when creating or changing a PSF configuration object. Consequently, you can only specify that you want to use the default output queue name for a PDF spooled file in a mapping program if your e-mail delivery is limited to PDFMAILSVR(*SNDDST). See 2.1.1, “Creating a PSF configuration object” on page 12, for details. This restriction does not exist for V5R3.

6.7.10 Calculation specifications for generating an AFPDS spooled files

Using intelligent routing, you have the option to generate individual segments of a large spooled file as separate AFPDS spooled files. The six fields used to customize the AFPDF output are contained in the spooled file AFPDS distribution format.

Table 6-10 lists the fields that are used to create an AFPDS spooled file. It is based on the chart in the *Using Intelligent Routing* with the following changes:

- Removed the Hex Offset column.
- Added the field names used in the sample program for the first body file.
- Added values used for the first body file for the offsets from beginning of the PDF mapping program output information structure. These values are based on the values calculated using the spreadsheet.

Table 6-10 Spooled file PDF distribution format

Decimal offset from beginning of PDF spooled file format	Decimal offset from beginning of output data structure	Field name	Type	Description
0	2133	AFPOUTQ	CHAR(10)	Name of output queue on which to spool AFP
10	2143	AFPOUTQLIB	CHAR(10)	Output queue library
20	2153	AFPSPLNAM	CHAR(10)	Spooled file name
30	2163	AFPUSRDTA	CHAR(10)	User data
40	2173	AFPUSRDFN	CHAR(255)	User defined data
295	2428	AFPFORM	CHAR(10)	Form type

All fields in this format are fixed length fields.

Lines 382 through 399 of the sample PDF mapping program illustrate the calculations specification used to define how the selected spooled file or segment is generated as a new AFP spooled file.

Consider the following points:

- **AFPSPLOFF** and **AFPSPLLEN:** The values for Offset to spooled file AFP distribution and Length of spooled file AFP distribution were calculated using the spreadsheet.
- **AFPSPLNAM:** If you leave the spooled file name blank, Infoprint Server uses the name of the original spooled file.
- **AFPSPLNAM:** If you specify an invalid spooled file name, Infoprint Server renames the file to QSYSPRT. A message is written to the job log for the WTR job. The original spooled file is not held.

- In the example, we used a value of *SPLF for AFPFORM. This tells Infoprint Server to use the form type of the original spooled file. In a similar manner, you can specify *SPLF for AFPSPLNAM, AFPUSRDTA, and AFPUSRDFN.
- You must specify 'AFPSAVE(*YES)' and specify an output queue using 'AFPOUTQ' in the PSF configuration object.
- **AFPOUTQ and AFPOUTQLIB:** If you specify the name of an output queue that does not exist, Infoprint Server places the AFP spooled file in the QPRINT output queue in QGPL. Messages to this effect are logged in the WTR job.
- **AFPOUTQ:** If you specify *PSFCFG for AFPOUTQ, Infoprint Server uses the name of the output queue and library specified using AFPOUTQ in the PSF defined option in the PSF configuration object. If an output queue is not specified in either place, the spooled file is placed in QGPL/QPRINT.
- **AFPOUTQLIB:** If you specify *LIBL for the output queue library name, Infoprint Server searches the library list of the WTR job. *LIBL, if used, must be specified in uppercase.
- You can use uppercase or lowercase for any of the previous character fields. One exception is the specification of the reserved words *SPLF, *PSFCFG, or *LIBL, which must be specified in uppercase. The output queue, library, and spooled file name are automatically converted to uppercase. User data, user-defined data, and form type remain in the case used in the PDF mapping program.
- The resulting spooled file has the same priority and owner of the original spooled file.
- If copies greater than one are specified in the spooled file attributes, it is ignored.
- If a form definition that contains multiple subgroups in a copy group is used, the resulting spooled file prints multiple pages for each original page.
- Infoprint Server intelligent routing does not direct spooled files to Facsimile Support for iSeries directly. However, you can place the fax phone number and other customer information in the user-defined data field. Place the resulting spooled file in an output queue that is monitored by a process that issues the Submit Fax command. See Chapter 7, "Using intelligent routing with Facsimile Support/400" on page 99.

Restriction: At time of writing, the AFP Respool function does not honor page range. It processes the entire spooled file. For example, consider a spooled file that has two segments that are one page long each. Assume that the logic of your PDF mapping program normally selects the first segment to be generated as an AFP spooled file, and the second segment is to be sent as an e-mail only. If you select PAGERANGE(2 2) when you release this spooled file to the PDF subsystem, the AFP spooled segment that corresponds to page 1 is generated, even though it was not selected as part of the page range.

6.7.11 Calculation specifications to end a program

When all the processing for the various desired actions takes place, the PDF mapping program returns control to Infoprint Server, as shown in line 403 of the sample.

6.8 How Infoprint Server and the PDF mapping program communicate

In the process of delivering your PDF and AFP output to the correct destination in the correct format, Infoprint Server calls the PDF mapping program multiple times. There is some mention of this in *Infoprint Server for iSeries: User's Guide*, G544-5775, and *Infoprint Server*

for iSeries, User's Guide Version 5, Release 3.0, G544-5775. This section provides additional information and explanation.

Note: Multiple calls to the PDF mapping program described in this section are independent of any extra calls to the PDF mapping program that are caused by setting the More processing field, CALLAGIN, in the Output information data structure to a value of '1'. For details about this field, see 6.9, "More processing (CALLAGIN)" on page 96.

Depending on the tasks that are being performed by your PDF mapping program, you may need to insert logic to track which stage of the overall process the PDF mapping program is being called. There are two different types of tests you may want to include in your program:

- Buffer size test
- Encryption and AFP Respool test

These tests are not always required, especially if you not doing complex processing in your PDF mapping program. The buffer size test is easier to understand and implement and is recommended from a performance perspective. The Encryption and AFP Respool test becomes more important if you are doing custom processing on the PDF file and are not getting the expected results.

6.8.1 Buffer space too small

One condition that causes the PDF mapping program to be called extra times deals with the buffer space. The first time Infoprint Server calls the PDF mapping program, it does not know how much space the PDF mapping program is going to want, so it assumes a relatively small size. It tells you this by passing the Length of output information buffer parameter. This parameter is described in 6.4, "PDF mapping program input and output parameters" on page 66.

This field tells the PDF mapping program how much space Infoprint Server has set aside in its buffer for information that the PDF mapping program is going to provide. The PDF mapping program responds by telling Infoprint Server how much space it really needs. It does this by assigning a value to the "Length of output information available" parameter.

If Infoprint Server realizes that it didn't give the PDF mapping program enough space, it calls the PDF mapping program again and uses the number that the PDF mapping program just requested. Think of the dialogue as shown here:

Infoprint Server: Hello PDF mapping program. I need some information about what to do with the PDF file for customer 100. I have room for 337 bytes of data in the output buffer.

PDF mapping program: Sorry, that is not enough space. I need 2509 bytes.

Infoprint Server: OK, let's try this again. I need some information about what to do with the PDF file for customer 100. I have room for 2509 bytes of data in the output buffer.

PDF mapping program: That is better. Here is the data you requested.

Infoprint Server: Thank you. I can process the PDF file now.

Note: The initial buffer size is different for different stages of the cycle.

Lines 158 through 166 of the sample PDF mapping program illustrate the logic used to handle the buffer size test.

OUTPUTLEN is the buffer size as provided by Infoprint Server, and OUTINFO is the size that the PDF mapping program needs for all of its data. If the Infoprint Server buffer is too small, the program does not need to do any other processing with respect to routing the file or segment. The program only needs to return the desired length the buffer, in this case 2509, in the OUTINFO field. Infoprint Server does not use the information in the output information data structure if the buffer is too small. Infoprint Server calls the PDF mapping program again with OUTPUTLEN = 2509.

We recommend that you do this test to improve performance. There is no sense doing any work, in particular the file lookup, in the main part of the program if the results are not used anyway.

6.8.2 Encryption of the PDF file for e-mail

In addition to the extra calls needed to set the buffer size, Infoprint Server calls the PDF mapping program multiple times per segment with respect to encrypting and routing the PDF file. This is because different pieces of information are needed for different processes within Infoprint Server.

The first time Infoprint Server calls to the PDF mapping program, it determines whether it must build an encrypted version of the PDF for e-mail or non-encrypted versions for e-mail, stream file, or spooled file generation. At that point, the input information data structure does not include the path and name of the PDF file because it is not built yet. Based on the results of that call to the exit program, Infoprint Server uses the encryption information to build the PDF files and store them in a temporary location in the /PSF400 directory in the IFS.

The second time Infoprint Server calls the PDF mapping program, it knows the name of the temporary location of the PDF files. It asks the PDF mapping program where to route the files.

An added complexity is that these two calls do not always happen in sequence. Infoprint Server processes the file based on 32K blocks. If the file is segmented, and there are multiple small segments within the one block, Infoprint Server may do the first call to gather the encryption information multiple times before doing the second set of calls to determine the distribution information.

This conversation represents the logic to handle PDF encryption, assuming that the buffer size logic is already handled:

Infoprint Server: Hello PDF mapping program. I am about to build the PDF file for the segment for customer 100. How do you want it encrypted?

PDF mapping program: Since you didn't tell me the PDF name, I assume that you want encryption information. I want to e-mail the PDF file for customer 100, and I want it encrypted so they can read, but not modify, it.

Infoprint Server: Hello PDF mapping program. I am about to build the PDF file for the segment for customer 101. How do you want it encrypted?

PDF mapping program: Since you didn't tell me the PDF name, I assume that you want encryption information. Do not encrypt the PDF file for customer 101. It is going to be generated as a spooled file.

...

Infoprint Server: I have the PDF file for customer 100 stored in /PSF400/..../000001.PDF. What do you want me to do with it?

PDF mapping program: E-mail it with these body files. Here is the address to send it to and the reply-to address to use.

Infoprint Server: I have the PDF file for customer 101 stored in /PSF400/..../000002.PDF. What do you want me to do with it?

PDF mapping program: Convert it to a spooled file and place it in the output queue for our color PDF printer.

...

6.8.3 AFP Respool function

The AFP Respool function also takes place early in the flow, while Infoprint Server is still processing the original spooled file. If AFPRESPOOL(*YES) is specified in the PSF configuration object, a separate call is made to the PDF mapping program to determine if and where that individual segment (or file) is to be respooled.

The information passed from Infoprint Server to the PDF mapping program is the same for both AFP Respool and encryption. In both of these cases, the PDF path and name in the input information data structure are not defined. There is no way to know which type of information is requested. There is also no strict rule for how these two types of calls are sequenced, because of the way Infoprint Server handles the spooled files in 32K blocks.

We recommend that you perform a test in your program to determine whether the PDF file name is known. If not, you must make sure that both the AFP Respool and the encryption information are defined before you return processing to Infoprint Server.

6.8.4 Logic for encryption and AFP Respool

You need to be concerned about extra calls that are needed for encryption and AFP Respool if your exit program is intended for extra processing on the PDF file while it is in its temporary location in the /PSF400 directory. An example of that type of processing is found in the sample program in Appendix E of *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250. In that program, if the lookup to the customer routing file failed, the PDF file was copied to an error folder. When that program is migrated to a V5R2 server that has the intelligent routing PTFs applied, it fails.

The sample V5R1 PDF mapping program can be modified by adding the lines of code shown in Example 6-2.

Example 6-2 Checking for a blank PDF file name

```
C* Check to see if being called for encryption.  The PDF file name
C* in the input structure is blank if called for encryption.
C* If you don't want to do any encryption, just set the output buffer
C* to hexadecimal zeros and return.
C*
C     EMPDFFILE     IFEQ      *BLANKS
C                   MOVE      *ALLX'00'     OUTDS
C                   RETURN
C                   ENDIF
```

In this particular case, neither encryption nor AFP Respool is requested. If either of the two functions is required, the fields specifying the encryption options and AFP Respool routing must be included between the IFEQ and ENDIF statements. Since there is no way to know if this is a call for encryption or AFP Respool information, you must set the fields that pertain to both functions.

If your PDF mapping program is mainly used to look up the desired routing information and is not doing any external work, you may not need the test for the PDF file name. All you do is fully populate the output information data structure with all the information that it needs for any of the calls and RETURN. It doesn't hurt if you provide the e-mail addressing information when it is looking for encryption or vice versa.

If your PDF mapping program includes logic that manipulates the temporary PDF file, such as the sample used in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250, make sure that you do the copy only in the last cycle, when the PDF spooled file name is known.

6.9 More processing (CALLAGIN)

If you set the More processing field of the output information data structure, CALLAGIN, to '1', the PDF mapping program is called by Infoprint Server again, using the same input information. Subsequent calls to the PDF mapping program that are triggered by the More processing field are made after the PDF file is generated. You cannot change the encryption options or create a second version of the AFP spooled file. You can use the More processing option to create additional stream files, PDF spooled files, or e-mail it additional times.

Important: Use the More processing field with care. If you set it to '1', and do not have a means to set it back to '0', you can cause Infoprint Server and the PDF mapping program to go into an endless loop.

Since the input information is the same for each call, the PDF mapping program must update an external object to trigger a condition that sets the field back to '0'.

6.9.1 Scenario for using more processing

Assume that you want to send a PDF as an e-mail to a customer and their sales representative. You want to add different attachments and use a different subject line for each copy. To do this, you must send it as two separate e-mails, not just one e-mail with multiple addressees or CCs. Assume that all segments are processed twice, once for the customer and once for the sales representative. Here is an example of a process to accomplish this:

1. Create an external data area and initialize it to a value of 'cust'.
2. Check an external data area to see which pass this is. Since it has a value 'cust', we know this is the first pass and we do the processing for the customer.
3. Generate the e-mail for the customer, including a marketing brochure as an additional attachment to the body. The subject line reflects your company name and their invoice number.
4. Assign a value of '1' to the CALLAGIN field. Update the external data area to 'rep.'. Return processing to Infoprint Server.
5. Infoprint Server calls the PDF mapping program a second time with the same input information.
6. Check the external data area. Since it has a value of 'rep.', you build the e-mail for the sales representative, including a spreadsheet chart showing the customer's past sales information. The subject line contains the customer's name.
7. Assign a value of '0' to the CALLAGIN field. Update the external data area to 'cust'. Return processing to Infoprint Server.

6.10 Logging the input and output data structures

A technique that proved invaluable during the initial testing of the PDF mapping program was setting up a pair of physical files to log copies of the input and output information data structures. On the input side, it helped to confirm the values in the mail tags. On the output side, it helped to verify the contents of the various formats used for each type of distribution.

By placing the WRITE instructions at various places in the logic, it helps you to understand the multiple calls between Infoprint Server and the exit program, as described in 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92.

The use of these two files is purely optional. You can remove them once the program is moved into production.

Example 6-3 and Example 6-4 show the DDS for the two files. Each of the two records is at least as long as the corresponding data structure.

Example 6-3 DDS for logging the input information data structure

```
A.........T.Name++++++RLen++TDpB......Functions
A          R LOGINR
A            LOGINDS      722
```

Example 6-4 DDS for logging the output information data structure

```
A.........T.Name++++++RLen++TDpB......Functions
A          R LOGOUTR
A            LOGOUTDS    3000
```

In the program, the files are referenced in the RPGLE file specifications as shown in Example 6-5.

Example 6-5 RPGLE file specifications for log files

```
FFilename++IPEASF.....L.....A.Device+.Keywords+++
FLOGIN     O    E             DISK
FLOGOUT    O    E             DISK
```

Since no value for EXTFILE is used in this sample, the log files must be placed in a library that is part of the job library list for the PDJ job of the writer. QGPL was used during testing of this program. QSPLJOB must have update authority to these files.

Example 6-6 illustrates sample calculation specifications used to capture the log information and write it to the physical files. These statements were usually placed just before a RETURN statement.

Example 6-6 Calculation specifications to capture the log information

```
C                   EVAL      LOGINDS = INPUTDS
C                   EVAL      LOGOUTDS =  OUTDS
C                   WRITE     LOGINR
C                   WRITE     LOGOUTR
C                   RETURN
```

The resulting files can be viewed or printed to assist with problem determination.

6.10.1 Writing a custom log file

In a production environment, a variation of these files can provide a historical record of how different files or segments were handled by Infoprint Server for iSeries.

Consider the example of the Super Sun Seeds scenario used as a basis for the PDF mapping program sample. Super Sun Seeds wanted some way to retrieve the PDF files that were stored as stream files in the IFS. In the current implementation, the files are filed in folders for each month, and the name of the file is based on the invoice number.

The Super Sun Seeds' programmer added a custom log file to the exit program application. It includes other relevant information about the invoice stored, such as customer number, date and time stamp, and the path name of the PDF file. Each time a PDF stream file is generated, a record is written to this file. Super Sun Seeds can then use this information to quickly find and retrieve any stored PDF invoice.

Similar logs can be used to track the e-mail addresses used by the mail function. That way, if you change the master file record for a given company, the log stills reflect who an e-mail was sent to before the change. This may be important for auditing purposes.

7

Using intelligent routing with Facsimile Support/400

Many clients have expressed the need to deliver the output of intelligent routing to the intended recipient as a fax document. This is currently not an option supported directly by Infoprint Server and intelligent routing. This chapter describes a way to use intelligent routing in combination with an output queue monitor to achieve that purpose.

The overall concept is to use the Advanced Function Presentation (AFP) Respool capability of Infoprint Server. Instead of placing the output in an output queue that is associated with a physical device, it is placed in a separate output queue. The information that is needed by Facsimile Support, such as the phone number and other cover page information, is placed in the user-defined data field of the spooled file. A monitor program takes the information about the spooled files and, using the information in the User defined data, submits the request to send the document as a fax.

This chapter describes the steps that a programmer needs to route the output from Infoprint Server for iSeries to Facsimile Support/400:

- Modify the intelligent routing mapping object or mapping program to accommodate information needed for faxing.
- Route the output to a special output queue that has a data queue assigned to it.
- Create an output queue monitor that reads the data queue, accesses the file information, and sends the document as a fax to the intended recipient.

This chapter also provides a basic outline of programming techniques that skilled programmers can use to develop an application that is robust and has complete error handling and maintenance.

Prior to reading this chapter, you must be familiar with the following information:

- The information in other chapters of this Redpaper that describe how to set up Infoprint Server with intelligent routing for iSeries
- The use of a mapping object or mapping program

- RPGLE, CL or other iSeries programming languages
- Facsimile Support/400

System application programming interfaces (APIs) are used to develop this solution. For detailed information about any of the APIs used in this chapter, refer to the iSeries Information Center on the Web at:

http://publib.boulder.ibm.com/html/as400/infocenter.html

When you reach the Information Center, go to **Programming-> APIs-> API finder**.

7.1 Configuring for intelligent routing to fax

In this first scenario, a given file or segment can be printed or faxed, but not both. See 7.2, "Fax, print, or both" on page 103, for an alternative scenario that includes the option to print, fax, or both.

The following steps explain how to set up the iSeries to allow Infoprint Server to route the output to Facsimile Support/400:

1. Set up your Portable Document Format (PDF) subsystem for intelligent routing. Use either a mapping program or a mapping object. The Print Services Facility (PSF) configuration object must specify AFPSAVE(*YES).
2. Create a data queue to be used by the monitor program. For more information about how data queues are used with output queues, see *IBM @server iSeries Printer Device Programming, Version 5*, SC41-5713. Here is an example of the command to create a data queue to be monitored for the purposes of sending a fax:

   ```
   CRTDTAQ DTAQ(mylib/FAXMON) MAXLEN(128) TEXT('Monitor for fax')
   ```

3. Create a user output queue to be used as a temporary holding place for the spooled file. Use the DTAQ parameter of the CRTOUTQ command to associate the data queue you created in step 2 to the output queue. Here is an example of the command to create an output queue with an associated data queue:

   ```
   CRTOUTQ OUTQ(mylib/FAXMON) TEXT('Monitor for fax.') DTAQ(mylib/FAXMON)
   ```

4. When the Submit Fax command was processed, a second entry was generated in the data queue. To prevent this, a second output queue was created as an intermediary step. This output queue does not have a data queue associated with it. Here is an example of the command to create the second output queue:

   ```
   CRTOUTQ OUTQ(mylib/FAXMON2) TEXT('Second output queue for fax monitor.')
   ```

5. Set up intelligent routing to generate an AFP spooled file for the desired file or segment.
   - If you are using a mapping object, create entries for the PDF Map Action for AFP spooled file = *YES for the desired spooled files or segments. Set the value for Output queue to the name of the output queue created in step 3. In the User Defined Data field, enter the fax information, such as phone number and recipient name. Be aware that the information must be in a strict format to be used as parameters in the send fax command.

     Figure 7-1 shows an example of adding an entry to the mapping object. The first 20 characters of the user-defined data field are reserved for the phone number. The next 20 character are used for the addressee name that appears on the cover page.

```
                              Add AFP spooled file
Type choices, press Enter.

AFP spooled file:
  Output queue . . . . . .   FAXMON            Name, *PSFCFG
    Library  . . . . . .       QGPL            Name
  Spooled file . . . . . .   *SPLF             Name, *SPLF
  User data  . . . . . .     *SPLF             Character value, *SPLF
  Form type  . . . . . .     *SPLF             Character value, *SPLF
  User-defined data  . . . .  1 513 555 0314      Dora Jackson


                                               Character value, *SPLF
```

Figure 7-1 Mapping object entry to route output to fax

– If you are using a mapping program, add the database files that contain the fax information to the file specifications of the program. Add logic to the program to check for the request to distribute the file or segment as a fax. Construct the string for the user-defined data in the desired format. Direct the output to the output queue that is being monitored for fax.

  In Example 7-1, the data fields PHONE and USER are combined in a data structure, FAXINFO, which is then assigned to the AFPUSRDFN field in the output data structure.

Example 7-1 Mapping program logic to route output to fax

```
DName++++++++++++ETDsFrom+++To/L+++IDc.Keywords+
D FAXINFO         DS
D  PHONE                   1     20
D  USER                   21     40
......
CLONO1Factor1+++++++Opcode&ExtFactor2+++++++Result++++
C* Check customer preference for creation of FAX
C*    DISPFAX       IFEQ      'Y'
C*
C*   Create an AFP spooled file in the FAX output queue
C                   EVAL      DSPAFPSPLF = '1'
C*
C*     Attributes for Fax Spooled file
C                   EVAL      AFPSPLOFF  = 2133
C                   EVAL      AFPSPLLEN  = 305
C                   EVAL      AFPOUTQ    = 'FAXMON'
C                   EVAL      AFPOUTQLIB = 'QGPL'
C                   EVAL      AFPSPLNAM  = '*SPLF'
C                   EVAL      AFPUSRDTA  = '*SPLF'
C                   EVAL      AFPUSRDFN  = FAXINFO
C                   EVAL      AFPFORM    = '*SPLF  '
```

6. Create an output queue monitor program to receive entries from the data queue. You can find a sample program in “Mainline monitor program” on page 175. The monitor CL program performs these basic steps:
   a. It checks to see if there is a condition that requires it to stop. These are examples of conditions that require the monitor program to stop:
      - A request to end the subsystem in which the program is running.
      - A separate program places the word ‘STOP’ in the data queue using the QSNDDTAQ API. You can find an example of a program that performs this task in “Program to submit STOP to data queue” on page 178.
   b. The program uses the QRCVDTAQ API to receive the information about the spooled file. For more information about the layout of the information received from the data queue, see *IBM @server iSeries Printer Device Programming, Version 5*, SC41-5713.
   c. The spooled file name, qualified job name, and spooled file number are passed to a procedure written as a separate RPGLE module. This procedure uses the QUSRSPLA API to retrieve the spooled file attributes. The data structure for the SPLA0100 format used by QUSRSPLA can be copied into your program from member QUSRSPLA in source physical file QSYSINC/QRPGLESRC. The user-defined data field, QUSUDD, is found in positions 1157 through 1411 of the SPLA0100 format.

      You can find a sample program in “Output queue monitor for fax” on page 175. The user-defined data, along with some error codes, are returned to the calling program.
   d. To prevent duplicate entries from being generated in the data queue, the spooled file is moved to the second output queue, FAXMON2, using the CHGSPLFA command.
   e. The fax telephone number and recipient name are extracted from the user-defined data.
   f. The spooled file is submitted to the fax subsystem using the SBMFAX command.
   g. Some basic error handling is performed.
7. The monitor program is submitted to run in the background as a batch job.

Figure 7-2 illustrates the steps used to route output to FAX using Infoprint Server and intelligent routing.

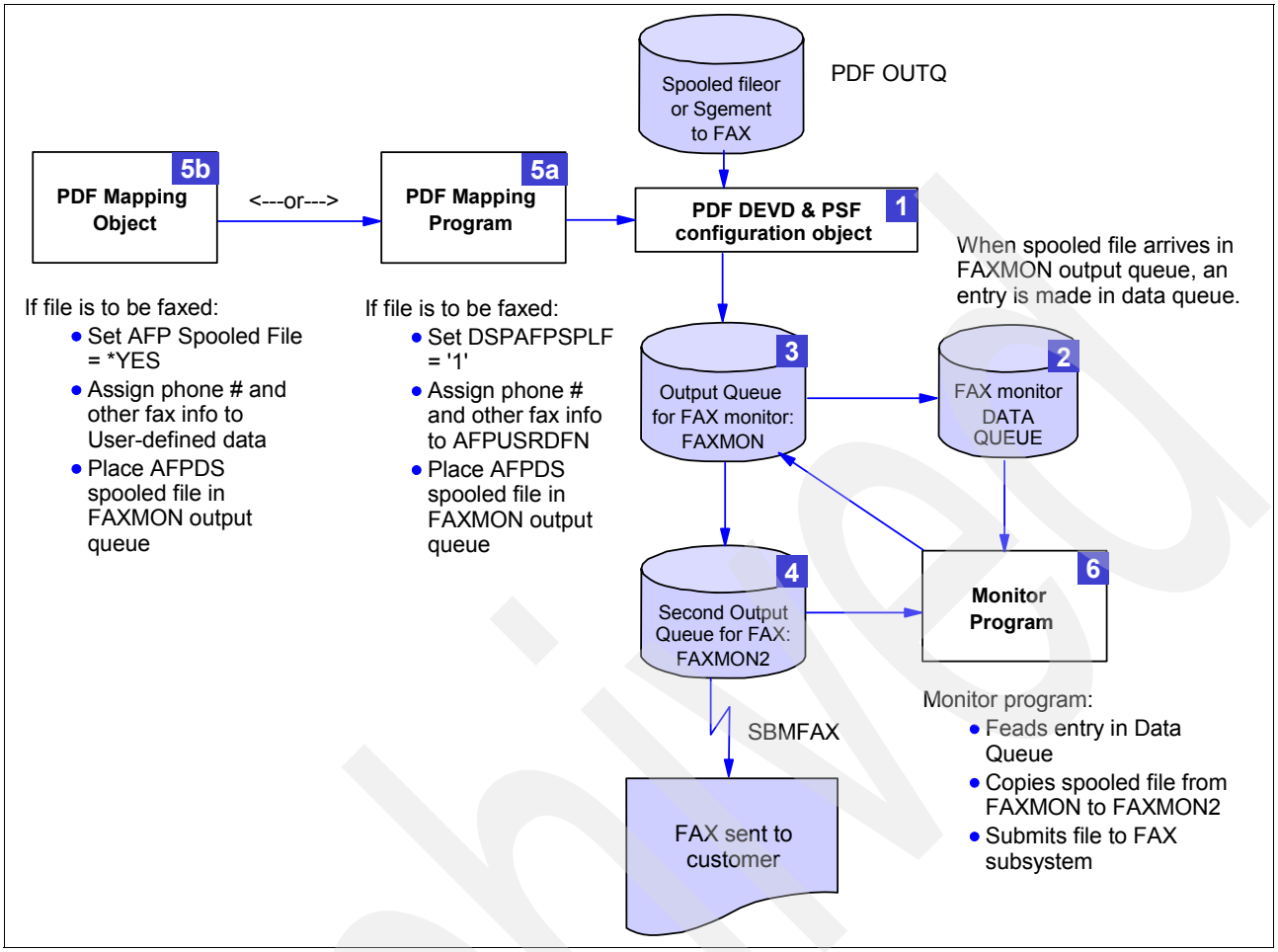


Figure 7-2 Intelligent routing to FAX

7.2 Fax, print, or both

If you need an option to print and fax the same spooled file or segment, consider this scenario. It is based on the previous example, with the following additions and changes:

1. In your mapping program or mapping object, assign the customer number or other key value to the user data field. This field can be up to 10 characters in length. Depending on the combination of fax or print, assign the following value to the AFP spooled file's User defined data field.

   – Fax and print: `AFPRESPOOL(*PRINT)`
   – Fax only: `AFPRESPOOL(*NOPRINT)`
   – Print only: Leave the user-defined data field blank

   If any of these three options are specified for the file or segment, set the disposition value for AFP print to '1' in the PDF mapping program, or *YES if you are using a PDF mapping object. If neither print nor fax is required for the spooled file or segment, set the disposition value for AFP print to '0' or *NO. Direct any file that is either to be faxed or to be printed to the output queue that is associated with the physical printer.

2. Create or change the PSF configuration object for the physical printer to support the AFP Respool function. Direct the respooled files to the fax monitor output queue.

   ```
   CHGPSFCFG PSFCFG(QGPL/myprinter) PSFDFNOPT('AFPSAVE(*YES)''AFPOUTQ(mylib/FAXMON)')
   ```

This output queue does not use the PDF subsystem. It does not need a mapping program or mapping object. Files are processed according to these rules:

- Files that have AFPRESPOOL(*PRINT) in the user-defined data are respooled to the fax output queue. They are printed by the physical printer.
- Files that have AFPRESPOOL(*NOPRINT) in the user-defined data are respooled to the fax output queue. They do not print on the physical printer.
- Files that do not have an entry for AFPRESPOOL in the user-defined data print to the physical device. They are not respooled to the fax monitor output queue.

3. Create the two Fax monitor output queues, FAXMON and FAXMON2, as explained earlier.
4. Create a monitor program similar to the one described in the previous example. In this version of the program, you retrieve the user data field, QUSUD01, instead of the user defined data field. It is in positions 91 through 100 of the SPLA0100 format.
5. Use the key from the user data field to look up the information needed to construct the SNDFAX command, such as the fax telephone number and recipient's name, from a user database.

8

Using APIs to add entries to PDF mapping object

This chapter provides information about how to write a program that uses a system application programming interface (API) to add entries to a Portable Document Format (PDF) mapping object.

Prior to reading this chapter, you must be familiar with using PDF mapping objects, as described in Chapter 3, “Using a PDF mapping object” on page 27. The fields that are added to the PDF mapping object using an API follow the same rules as those that are added using the interactive interface.

You must also be familiar with the following programming topics as discussed in Chapter 6, “Using the PDF mapping program” on page 63:

- Reserved fields (see 6.3.1, “Reserved fields” on page 64)
- Binary fields (see 6.3.2, “Binary fields” on page 65)
- Offsets (see 6.3.3, “Offsets” on page 65)
- Using a spreadsheet (see 6.6, “Using a spreadsheet to calculate length and offset values” on page 72)

8.1 Why use an API to update a PDF mapping object

The PDF mapping program brings the power of intelligent routing to the end user. If you have a relatively small number of entries to make, the interface provided with Infoprint Server may likely suite your needs. However, if you want to create entries to route PDF copies of invoices for each customer with whom you do business, you may find that the interactive approach may not be the best. If that is the case, you may choose to use a PDF mapping program.

The other option is to use a program similar to the one described in this chapter to initially populate the PDF mapping object, using information that is stored in a database file. Further individual updates may be made using the interactive interface.

Some users may not want to end their PDF writers for the time it takes to make manual changes to the PDF mapping object. You can use a standard interactive program to store the update information in a temporary file, in the format used by the API. Then use a program that uses the APIs to transcribe the information from the temporary file to the mapping object. You must end the writer to use the API to update the PDF mapping object, but the time is short.

8.1.1 PDF mapping object APIs

Five different system APIs are available to work with PDF mapping objects:

- **QPQCRTPM**: Create a PDF map
- **QPQDLTPM**: Delete a PDF map object
- **QPQOLPM**: Open a list of PDF map information
- **QPQAPME**: Add a PDF map entry
- **QPQRPME**: Remove PDF map entries

This chapter deals only with the Add PDF map entry API, QPQAPME. For detailed documentation about the PDF mapping object APIs, see the System API Reference information in the iSeries Information Center at:

`http://www.iseries.ibm.com/infocenter`

8.2 Sample program structure

The sample program was written in RPGLE, but any other high level language may be used. It follows a very simple flow. All values are hard coded to best display the purpose of each field. In a production implementation, it is likely that many of the fields will obtain their values from external data base files or from an interactive display file.

The sample program follows this sequence:

1. The sample program uses one file specification for logging and problem determination purposes. A copy of the action data structure, as well as any error codes if there is a problem, is written to the LOGAPI file.

   Example 8-1 shows the data description specifications (DDS) for the LOGAPI file.

Example 8-1 DDS for API logging file

```
A          R LOGAPIR
A            LOGAPIDS    3500
```

In a production environment, add in any file specifications for files that contain the data being used to update the PDF mapping object.

2. Following the file specifications, if any exist, are the definition specifications. This section is used to define individual fields and data structures used in the program. Definition specifications for some of the structures are supplied by IBM. These are copied into the program at compile time by using the /COPY compiler directive. The remaining fields must be included by the programmer.
3. The bulk of the calculation specifications is used to assign values to fields that are used in the various data structures. Most possible PDF actions are included in this sample.
4. After all the data is assigned, the QPQAPME API is called.
5. A copy of the action data structure is written to the LOGAPI file.
6. Finally, a check is done to see if there is an error condition. If the update is successful, the API returns the name of the library where the updated PDF mapping object resides. A second program is called. It sends the name of the library to the system operator if the action was successful. If there was an error, the word “error” is sent.
7. If there is an error, a copy of the error structure is written to the LOGAPI file to use for problem determination.

8.3 Required parameter group

You must supply ten parameters when you call the Add PDF Map Entry API. Table 8-1 describes these parameters. It shows the actual lengths or values used in this instance of a program that calls the API.

Table 8-1 Required parameter group

Field or structure name used in sample program	Description	Input or output	Type	Length	Field value
RLIB	Returned library name	Output	Char	10	
PDFMAP	Qualified PDF map object name	Input	Char	20	PDFMAP MIRA
QPQE0100	PDF map entry spooled file attributes	Input	Char	326	As assigned in program
ATTRIBLEN	Length of PDF map entry spooled file attributes	Input	Binary	4	326
ATTRIBMFT	Format of PDF map entry spooled file attributes	Input	Char	8	APME0100
ACTION	PDF mapping action	Input	Char	3289	As assigned in program
ACTIONLEN	Length of PDF mapping action	Input	Binary	4	3289
ACTIONFMT	Format of PDF mapping action	Input	Char	8	PMAP0100
ADDTYPE	Add type	Input	Binary	4	1
ERRCOD	Error code	I/O	Char	116	

The following sections supplement the information provided in the documentation available from the iSeries Information Center.

8.3.1 Returned library name

This field contains the name of the library that contains the PDF map object if the update is successful. This is useful if the library name was specified as *LIBL or *CURLIB.

8.3.2 Qualified PDF map object name

The first 10 characters of this structure contain the name of the PDF mapping object being updated. The second 10 characters contain the library name *LIBL or *CURLIB.

8.3.3 PDF map entry spooled file attributes

These attributes correspond to the selection criteria for PDF map entry as described in 3.4.2, “Selection criteria for a PDF map entry” on page 31. There is one difference in that the indication of whether a given entry is segmented occurs in a separate structure, as part of the mapping action definition. In the sample program, this is field QPQSE.

The attribute fields are grouped together in a predefined format, APME0100. The data structure corresponding to this format, QPQE0100, is supplied by IBM in member QPQAPME, in source file QRPGLESRC, in library QSYSINC. Similar code that can be included is provided for RPG, COBOL, and ILE COBOL.

8.3.4 Length of PDF map entry spooled file attributes

Since all of the fields in the spooled file attributes format are fixed length, the length of the PDF map entry spooled file always has a value of 326.

8.3.5 Format of PDF map entry spooled file attributes

This field tells the operating system how the data supplied in the attributes data structure is to be used. It must have a value of APME0100.

8.3.6 PDF mapping action

The PDF mapping action is the structure used to indicate which actions are to be taken on the selected spooled file or segment, and for each action, the detailed specifications. The reader may see similarities between this structure and the PDF mapping program output information data structure, as described in 6.5, “Structure of the output data” on page 68.

PMAP0100 format

The mapping action definition starts with a predefined format, PMAP0100. The data structure corresponding to this format, QPQP0100, is supplied by IBM in member QPQAPME in source file QRPGLESRC in library QSYSINC. Similar includable code is provided for RPG, COBOL and ILE COBOL.

The fields defined in PMAP0100 are similar to the fields in the extension area used by the PDF mapping program output information data structure. Several pairs of fields contain offsets and lengths for other structures which appear in other parts of the PDF mapping action structure.

If any of the indicated actions are not to be used, set the corresponding length and offset to 0.

The PMAP0100 format also includes fields to specify if the entry is to be segmented, and to define a description for the entry.

A separate format is used for each of the actions. It is not necessary to include formats for actions that are not used.

Mail information entry

This is likely the most complex of the structures. It contains some fields directly, such as the encryption information, as well as pointers and lengths for most of the other information used to define an e-mail. Some types of information can have a variable number of repeats. For example, you may specify multiple values for the To e-mail address. For these types of entries you specify the number of repeats along with the offset and length.

You have to option to specify the name of a stream file to use for the To e-mail address, CC e-mail address or BCC e-mail address entries. If you use these options do not specify individual names for the corresponding address type.

The entries for individual e-mail addresses, stream files, and PDF file name are separate fields that are defined outside of this structure.

The data structure corresponding to the mail information entry, QPQAPMIE, is supplied by IBM in member QPQAPME in source file QRPGLESRC in library QSYSINC. Similar includable code is provided for RPG, COBOL and ILE COBOL.

Note: At the time this Redpaper was written, we discovered that we defined several the fields at the end of the mail information entry section of the PDF mapping action structure incorrectly in the copy of QPQAPME in QSYSINC. “Corrected fields for main information entry structure” on page 184 lists the corrected versions of the added and changed the fields. A PTF is planned to update the source members in QSYSINC. Check with the IBM Support Center for availability.

PDF stream file information entry

This is a simple structure with only four fields. The first two define the offset and length for the PDF file path and name. This is followed by the field for Public authority and a reserved field. The PDF file path and name is defined as a separate field outside of this structure.

The data structure corresponding to the PDF stream file information entry, QPQDFSFE, is supplied by IBM in member QPQMAP in source file QRPGLESRC in library QSYSINC. Similar includable code is provided for RPG, COBOL and ILE COBOL. This structure was not used in the sample program. A user defined structure, STMFDS, was used instead. STMFDS includes the name of the PDF path and file.

PDF spooled file information entry

This is a simple structure containing the fields used to define a PDF spooled file being generated by Infoprint Server. There are no external fields that require offsets or lengths.

The data structure corresponding to the PDF spooled file information entry, QPQDFSFE00, is supplied by IBM in member QPQMAP in source file QRPGLESRC in library QSYSINC. Similar includable code is provided for RPG, COBOL and ILE COBOL. This structure was not used in the sample program. A user defined structure, PDFSPLFDS, was used instead.

AFPDS spooled file information entry

This is a simple structure containing the fields used to define a Advanced Function Presentation data stream (AFPDS) spooled file being generated by Infoprint Server. There are no external fields that require offsets or lengths.

The data structure corresponding to the AFPDS spooled file information entry, QPQDSSFE, is supplied by IBM in member QPQMAP in source file QRPGLESRC in library QSYSINC. Similar includable code is provided for RPG, COBOL and ILE COBOL. This structure was not used in the sample program. A user defined structure, AFPSPLFDS, was used instead.

8.3.7 Length of PDF mapping action

The information provided in the PDF mapping action my vary in length from installation to installation. This is because some fields or structures are optional, and other fields or structures may be used multiple times. For example, you may choose not to define an action for PDF spooled file, but wish to include three body files.

In the case of the sample program, this value is 3289.

8.3.8 Format of PDF mapping action

This field tells the operating system how the data supplied in the PDF mapping action data structure is to be used. It must have a value of PMAP0100.

8.3.9 Add type

This field defines what type of add is to be performed. If it has a value of 1, an existing entry may be overwritten if one exists. If Add type is set to 2, an error is returned if you try to add a new entry and one already exists with the same attributes.

8.3.10 Error code

The error code is a structure that contains several fields. It follows a standard structure used by most APIs on the iSeries server. The first field is passed from the program to the API to indicate how much error data the program wants to receive. If the program does not want to receive any data back, this value is set to 0. The remaining fields are described in the iSeries Information Center under the topic APIs-> Concepts-> Error code parameter.

If an error condition occurs, the API supplies a value to tell the calling program how many byes of data are passed back to describe the error. It is common practice to check this field after a call to the API. If it has a value other than 0, then we know that some sort of error occurred. An inspection of the remaining fields can help pinpoint the problem. In the sample program, the field tested is BYTAVL.

In the sample program, the entire error code data structure has a length of 116. This allows for the receipt of a seven-byte message ID, plus up to 100 characters of message data, among other things.

8.4 Sample programs

You can find a copy of the sample programs and planning spreadsheet in “Using APIs to add entries to a PDF mapping object” on page 178.

9

Saving encrypted PDF files

The encryption options described in Chapter 4, “Encryption for iSeries-generated e-mail” on page 41, are normally available only for Portable Document Format (PDF) files that are sent as e-mail attachments. The encryption options do not apply to the PDF stream files that are generated using the standard techniques available with a PDF mapping object, or PDF mapping program, as described in Chapter 3, “Using a PDF mapping object” on page 27, and Chapter 6, “Using the PDF mapping program” on page 63.

Some clients require the PDF spooled files, which are stored as stream files in the integrated file system (IFS), to be encrypted. For example, a financial institution chooses not to use e-mail to deliver monthly credit card statements. Instead they convert the statement to a PDF, which is placed on the iSeries IFS, and then deliver it to the end user upon request using an interactive Web-based application. Due to the sensitive nature of the documents, the financial institute uses the encryption options to assign a master password to the document and then selects the option to prevent the user from altering the contents.

This chapter describes a programming technique to save encrypted files to the IFS. This function is not available otherwise with Infoprint Server for iSeries for V5R2 or V5R3.

9.1 Technical background

To implement the technique described in this chapter, you must use a PDF mapping program. It is not possible to achieve the desired results using a PDF mapping object.

To understand the process, it is important to know how Infoprint Server and the PDF mapping program communicate. We recommend that you review 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92, before you continue with this chapter.

The technique relies on having the PDF mapping program tell Infoprint Server different things when it is called before versus after the PDF files are created:

- In the earlier call to the mapping program, before the PDF file is build, the PDF mapping program tells Infoprint Server that it wants to e-mail an encrypted version of the file.
- In the subsequent call, the PDF mapping program tells Infoprint Server that the file is not to be e-mailed. Instead, the mapping program itself copies the encrypted PDF file to a destination determined in the program.

We illustrate this using a modified version of the conversation between Infoprint Server and the PDF mapping program (assumes that the buffer size logic is already handled):

Infoprint Server: Hello PDF mapping program. I am about to build the PDF file for the segment for customer 100. How do you want to encrypt it?

PDF mapping program: Since you didn't tell me the PDF name, I assume you want encryption information. I want to e-mail the PDF file for customer 100, and I want it encrypted so they can read, but not modify, it.

...

Infoprint Server: I have the (non-encrypted) PDF file for customer 100 stored in /PSF400/..../000001.PDF. What do you want me to do with it?

PDF mapping program: I changed my mind. Do not e-mail it or take any other action. Instead, I learned that the encrypted version of the file is named /PSF400/..../X000001.PDF. I am copying it to the /encrypted directory.

Infoprint Server generates temporary copies of the PDF file, based on the values passed to it from the exit program. The exit program knows that the PDF files have yet to be built if the value for the name of the PDF file, which is passed to the program in the mapping program input data structure PDFFILE, does not have a value. Under normal circumstances, Infoprint Server generates an encrypted version of the spooled file only if you selected the e-mail option and have requested that encryption be done. With reference to the sample program, this involves setting DISPOSTN to a value of ‘1’. It also involves having a non-zero value for both the length and offset to the encryption area, ECRYPTOFF and ENCRYPTLEN.

A non-encrypted version of the PDF file is also built, even if it is not explicitly requested. This is done in case an error condition occurs and an e-mail is to be sent to the PDF administrator. For more information about the PDF administrator, see 2.2.3, “PDF administrator” on page 23.

The temporary spooled files that are generated have a path and file name similar to the following examples:

```
/PSF400/332539/QPADEV0003/MIRA/332287_000005_QSYSPRT_01122004_000001.PDF
/PSF400/332539/QPADEV0003/MIRA/332287_000005_QSYSPRT_01122004_X000001.PDF
```

Note the following explanation:

- PSF400 is the default directory used by Infoprint Server for temporary PDF files.
- 332539 is the job number of the WTR job.
- QPADEV0003, MIRA, and 332287 are the job name, user, and number that created the spooled file.
- 01122004 is the date that the spooled file is converted to PDF.
- 000001.PDF or X000001.PDF is assigned by Infoprint Server. The name with the X is the encrypted copy, and the version without the X is used for the uninterrupted copy. If the original spooled file is segmented, this value is incremented for each segment, for example 000002.PDF, X000002.PDF, and so on.

After the PDF files are created by Infoprint Server, the exit program is called again. The name of the un-encrypted version of the PDF stream file is passed to the PDF mapping program using the PDFFILE field of the input data structure.

When the mapping program sees that there is a value for the PDF file name, it performs the following actions:

- Prevents the file from being e-mail by setting the disposition value for e-mail, DISPOSTN, to zero.
- Performs character string manipulation to build the name of the encrypted version of the file. An X is inserted into the PDF file name.
- Passes the name of the encrypted version of the spooled file to a CL program. If you want to change the name of the spooled file, you can also pass other information about the file to the CL program. In the sample program, we pass the six-character customer number that was extracted from the first six characters of the routing tag.

9.2 Programming details

This program is loosely based on the PDF mapping program sample presented in Chapter 6, “Using the PDF mapping program” on page 63. We expect that you understand the programming concepts behind the use of a PDF mapping program. Only the differences between that program and the new program used to copy the encrypted files are discussed in this chapter.

Note: The program described in this chapter is based on the program that is described in Chapter 6, “Using the PDF mapping program” on page 63. The program from that chapter is referred to here as the *intelligent routing program*.

9.2.1 RPGLE program: Definition specifications

The definition specifications are used to define fields or to group fields together in data structures.

Data structure for secondary program name

The actual copying of the encrypted file takes place in a second program. A definition specification is used to assign the fully qualified name of the program, MIRA/COPYENCL, to a variable, COPYENCL. The library list for the PDF mapping program is limited to the library list of the writer and usually does not include user libraries. The definition allows the program to be called from a user library.

Input data structure

The input data structure for the mapping program is the same as the structure used in the intelligent routing PDM mapping program. See 6.4.1, “PDF mapping program input information” on page 66, for additional details.

Output data structure

The output data structure for the mapping program is greatly reduced in size from the structure used in the intelligent routing sample. It only includes a subset of fields that are needed to create the encrypted version of the spooled file.

For planning purposes, a spreadsheet is used to calculate the offsets to the various fields. This spreadsheet uses a subset of the fields that were used for intelligent routing. For more information about using a spreadsheet to build the output data structure, see 6.6, “Using a spreadsheet to calculate length and offset values” on page 72.

The fields used in the data structure include:

- All the fields from the base structure.
- Of the fields in the extension area, only the fields that deal with encryption are defined. Note that an absolute to and from position is used for ENCRPTOFF and ENCRPTLEN. This is because several fields that you normally find in the extension area are not used.
- The additional fields that used are only those that define the encryption.

For more information about the base structure, extension area, and other fields in the output data structure, see 6.5, “Structure of the output data” on page 68.

Stand-alone fields

Three new stand-alone fields are needed for this program:

- **NAMELEN**: Contains the length of the string for the name and path of the unencrypted PDF file.
- **ENNAME**: This is the name and path of the encrypted version of the PDF file.
- **CUSTNO**: The customer number is extracted from the routing tag.

RPGLE program: Calculation specifications

The calculation specifications are used to set initial and default values, define the encryption parameters, and to prepare for copying the encrypted file to a user directory.

Initial setup

The following steps are performed for each call of the PDF mapping program:

1. The parameters that are passed from Infoprint Server are defined.
2. The program checks that the output buffer space is long enough. See 6.8, “How Infoprint Server and the PDF mapping program communicate” on page 92, for more information about the checking the buffer space.
3. Reserved fields and default values are set. Disposition fields for all actions, other than e-mail, are set to ‘0’.

PDF file name not known

A check is done to see if PDFFILE is all blanks. If the condition is true, the PDF files are not created. If this is the case, the program sets the disposition field for e-mail, DISPOSTN, to ‘1’, and sets the various fields associated with encryption. A value must be set for ADDRESS to

prevent an error from occurring. This address is ultimately ignored when the e-mail disposition is set back to ‘0’ in a later step.

PDF file name is known

This is the ELSE branch of the condition. The value for the PDFFILE field contains the name and path of the temporary PDF file that is not encrypted. The name and path of the encrypted version of the file is built by using string manipulation to insert an “X” before the file sequence number portion of the name.

A substring operation is done on the routing tag to extract the customer number.

A CL program is defined by the COPYENCL variable. The name of the encrypted PDF file and customer number are passed to it as variables.

9.3 CL program to copy the encrypted file

The CL program, COPYENCL, receives the two parameters from the RPGLE program and performs the copy function. For illustration purposes, we include two versions of the copy:

- In the first copy, the PDF file is copied to a new location, in this case the /encrypted directory, but the name portion of the file is unchanged. This is done using the TODIR parameter in the CPY command.
- In the second copy, a new name and path is built using the customer number. It is placed in the &TARGET field. The TOOBJ parameter in the CPY command tells OS/400 to use that entire string as the destination.

You may want to use this opportunity to change the iSeries object authority of the PDF stream file using the Change Authority (CHGAUT) command. This is different than the encryption since it is an iSeries system function. This is not done in the sample program.

9.4 Results

We tested two files:

- QSYSPRT is a simple screen print. The routing tag is added to the spooled file using the user-defined data:

  ```
  CHGSPLFA QSYSPRT USRDFNDTA('(mailtag(123456)')
  ```

- OUTPUTMRG contains embedded tags that were generated using the Create AFP Data (CRTAFPDTA) command. The first three characters of the customer number are blank. Only the pages corresponding to the first segment of the spooled file are processed.

Both files are processed by the PDF mapping program, resulting in four files being generated:

- Two with custom names
- Two using the original system assigned names

Figure 9-1 shows the resulting files as they appear using a mapped network drive in Windows Explorer.

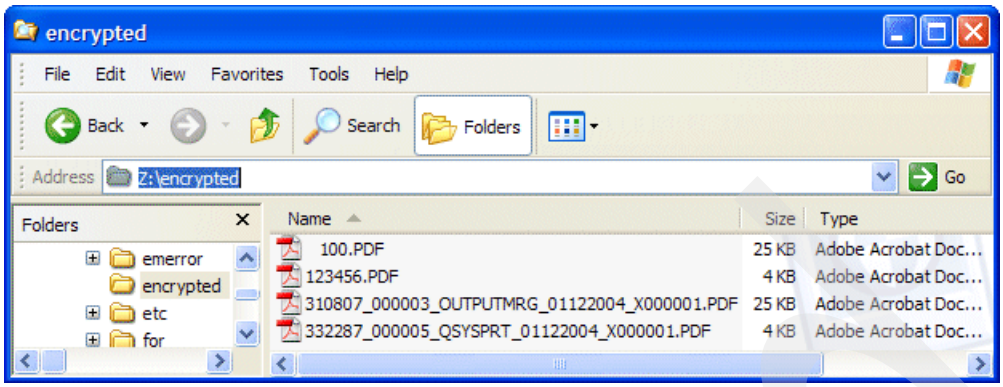


Figure 9-1 Windows Explorer showing encrypted PDF files

Notice that the spaces at the beginning of the customer number remain at the beginning of the file name. Also, notice the “X” in the file names of the samples that used the system-supplied names.

A

Infoprint Server jobs and job logs

There may be times when you don't achieve the expected results from an Infoprint Server application. In most cases, you can find, in the iSeries job logs, information about why the job failed to do what was expected. The challenge is knowing where to find the job logs for the Infoprint Server jobs.

Infoprint Server jobs

To understand how to find your job logs, you must understand the types of jobs that are used by Infoprint Server Portable Document Format (PDF) generation.

When you start the writer for your Infoprint Server PDF conversion job, you actually start three iSeries jobs. All jobs three run in QSPL. They are designated as:

- WTR
- BCH
- PDJ

It is useful to know how each of these jobs fits in to the overall process. This helps you know where you need to look for errors that pertain to different problems.

Job names

The writer is usually given the same name as the device description, unless you specify a different name for the WTR parameter in the STRPRTWTR command. The writer name is used for the first part of the job name. The job user name is QSPLJOB. The job numbers for the three jobs are assigned in the order WTR, BCH, and PDJ. These job numbers may not be sequential if other iSeries jobs are initiated at the same time.

The WTR (writer) job

This job is used by Print Services Facility (PSF) to convert the spooled file from its original data stream, such as *SCS, *LINE, *IPDS, or *AFPDSLINE to Advanced Function Presentation data stream (AFPDS), and then to *IPDS. As part of this operation, it processes external resources that are needed to generate the page. Consequently, the most common types of messages in the job log associated with the WTR job are ones that pertain to missing resources.

The writer job also processes requests for AFP Respool. It has the lowest job number.

The BCH (batch) job

This job is used by Infoprint Server to perform the actual Intelligent Printer Data Stream (IPDS) to PDF conversion. We have yet to have a problem that resulted in messages being generated by this job.

The BCH job has the second job number.

The PDJ (print driver) job

Under normal printing operations, the PDJ job manages the delivery of the IPDS to the output device. In the case of Infoprint Server's PDF generation, this job manages the delivery of the PDF output to the specified destination. This includes routing information that is defined in the PSF configuration object, a user PDF mapping program, or by using the PDF mapping object.

The PDJ job has the highest job number.

Figure A-1 shows the panel that you see when you run the WRKACTJOB SBS(QSPL) command. This command shows all active jobs running in subsystem QSPL. Press F11 twice to see the view that includes the job number.

EMAIL52P is a writer that is set up for PDF generation. Three jobs are associated with this device. Note the sequence of the job numbers for the three segments.

MP17PRN is an IPDS printer configured with AFP(*YES). It has a WTR job and a PDJ job, but no BCH job, associated with it.

PRT4324LPR is a non-IPDS device. It only has the one WTR job associated with it.

```
                          Work with Active Jobs                    BLDAS52
                                                                10/29/03  11:54:24
CPU %:   10.1     Elapsed time:   00:00:15     Active jobs:   254

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Number  Type  CPU %  Threads
     QSPL           QSYS        287129  SBS      .0        1
       EMAIL52P     QSPLJOB     287980  BCH     3.3        1
       EMAIL52P     QSPLJOB     287979  WTR     1.8        1
       EMAIL52P     QSPLJOB     287981  PDJ     3.9        9
       MP17PRN      QSPLJOB     287695  WTR      .0        1
       MP17PRN      QSPLJOB     287696  PDJ      .0        1
       PRT4324LPR   QSPLJOB     287131  WTR      .0        1
 Parameters or command
===>
F3=Exit   F5=Refresh   F7=Find   F10=Restart statistics   F11=Display status
F12=Cancel   F17=Top   F18=Bottom   F23=More options      F24=More keys
```

Figure A-1 Different types of writers

Finding messages associated with the writer

The method to find the messages in the job log depends on whether the writer jobs remained active or are stopped, either as a result of the error or by the operator.

Messages for an active writer

If the error is minor, with respect to the PDF writer, the spooled file is held, but processing of the next job continues. Examples of these include:

- Missing resources. For example, PSF may be unable to find an overlay, for example, a message is placed in the job log for the WTR job. In this case, the decision to print or hold the spooled file depends on the FIDELITY parameter in the spooled file.
- Routing errors, such as incorrect mail tag syntax. If the e-mail address is specified with the incorrect syntax, for example, if a pound symbol (#) is used instead of the at symbol (@), such as joe#ibm.com, the spooled file is held. You can find messages for this type of error in the job log for the PDJ job. An exception to this is that messages that pertain to the AFP Respool function are found in the job log for the WTR job.

To find the messages for errors while the writer is still active, perform these actions:

1. Enter the command:

   ```
   WRKWTR pdfmail1
   ```

If you enter the full name of the writer, you see the Work with Printer Writer display (Figure A-2). If you enter a generic name, such as pdf*, you see a selection list. Use option 5 to select the writer you want to work with.

```
                          Work with Printer Writer

Writer . . . . . :   PDFMAIL1       User . . . . . :   QSPLJOB
Number . . . . . :   287996

Started by user  . . . . . :   MIRA
Status:
  Writing  . . . . . :   N
  Waiting on message . . . . . :   N
  Held . . . . . :   N
  End pending  . . . . . :   N
  Hold pending . . . . . :   N
  Between files  . . . . . :   Y
  Between copies . . . . . :   N
  Waiting for data . . . . . :   N
  Waiting for device . . . . . :   N
  On job queue . . . . . :   N

                                                                    More...
Press Enter to continue.

F3=Exit     F5=Refresh   F6=Messages   F10=Release   F11=Hold   F12=Cancel
F14=Queue   F24=More keys
```

Figure A-2 Work with Printer Writer display

F6 takes you to messages in the message queue that are used for the writer. Be aware of these points:

- Not all of the job's messages are sent to the message queue. In particular, the errors in the PDJ job do not appear when you select this option.
- If the message queue happens to be the message queue for QSYSOPR, which is the default when a device is created, there may be many other types of system messages in the queue. This may make it more difficult to find the message you need.

A better way is to work with the writer job. The option to select this is displayed when you press F24 for more keys. Figure A-3 shows the additional key prompts that are shown when you press F24.

```
Press Enter to continue.

F13=Change   F17=Writer job   F18=End writer   F24=More keys
```

Figure A-3 Pressing F24 on the Work with Printer Writer display

Press F17 to work with the writer jobs. You see a display similar to the example in Figure A-4. Note that you do not see an entry for the BCH job.

```
                          Select Writer Job

Type options, press Enter.
  1=Select

Opt     Job            User           Number     Type
        PDFMAIL1       QSPLJOB        287996     WTR
        PDFMAIL1       QSPLJOB        287998     PDJ
```

Figure A-4 Select Writer Job

2. If you suspect that the problem lies with the formatting of the page, in particular with external resources, or with AFP Respool, select the WTR job. If you suspect that the problem is related to the PDF generation or the routing of the document or segment, select the PDJ job. Press Enter.
3. The Work with Job display opens. Select option 10 to display the job log.
4. Depending on the type of error, you may need to press F10 to see detailed messages.
5. When you find a message that appears to describe your problem, move the cursor onto the message line and press F1 for additional message information. Figure A-5 shows an example of a message that is issued in the PDJ job due to an addressing error.

```
                        Additional Message Information

Message ID . . . . . :   PQT3608      Severity . . . . . :   10
Message type . . . . :   Diagnostic
Date sent  . . . . . :   10/29/03     Time sent  . . . . . :   23:34:48

Message . . . . :   Email address is not specified correctly.
Cause . . . . . :   An error was detected while performing syntax checking for
  email address joe#blow.com. The error code is 10. The error codes listed
  below indicate the error detected:
     3 - Domain name missing.
     9 - The user ID is missing.
    10 - Address does not contain the '@' character.
    11 - The address contains a character that is interpreted as an incorrect
  delimiter.
    12 - Address exceeds maximum allowable length. The maximum length for a
  user ID is 64 characters. The maximum length for the entire address is 256
  characters.
                                                                        More...
Press Enter to continue.

F3=Exit   F6=Print   F9=Display message details   F12=Cancel
F21=Select assistance level
```

Figure A-5 Addressing error

Messages for a writer that has ended

If the writer has ended, either normally by the operator entering an ENDWTR command, or abnormally because of an error, the job logs are generated as spooled files. Errors that cause the writer to end abnormally include:

- A data queue is specified in the PSF configuration object, but it does not exist.
- A severe error is found in the PDF mapping program. For example, the PDF mapping program refers to a database file, but is unable to open the file.

You can no longer use the WRKWTR command as the starting point to find messages in the job log. There are a few ways to find the job logs associated with the writer that has ended.

Looking in the output queue for all job logs

The default output queue for job logs is QSYS/QEZJOBLOG. Check with your system administrator to see if this has changed. To find messages in the QEZJOBLOG output queue, follow these steps:

1. Enter the following command to see a list of all spooled files in that output queue:

   ```
   WRKOUTQ QEZJOBLOG
   ```

   Press F18 to jump to the bottom of the queue. Look for spooled files named QPJOBLOG that are owned by QSPLJOB. The user data field contains the name of the writer. Pressing F11 displays the job number, which you can use to distinguish between the WTR, BCH, and PDJ jobs.

2. Enter option 5 next to the job log that you want to view. If the job log has many pages, use F18 to jump to the end and then page up to find the error.

3. Figure A-6 shows an example of an addressing error that is logged for a job that has ended. Note that the format is slightly different than the display for similar information for an active job, as shown in Figure A-5.

```
                                    Display Spooled File
File  . . . . . :   QPJOBLOG                                                  Page/Line   2/50
Control . . . . .   +2                                                        Columns     1 - 130
Find  . . . . . .   joe
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0....+....1....+....2....+....3
PQT3608    Diagnostic             10   10/29/03  23:34:48.636296  QSI1062502   QRPLOBJ     17A5     QCMD        QSYS        01A6
                                Message . . . . :   Email address is not specified correctly.
                                Cause . . . . . :   An error was detected while performing syntax checking for
                                  email address joe#blow.com. The error code is 10. The error codes listed
                                  below indicate the error detected:  3 - Domain name missing.  9 - The user
                                  ID is missing. 10 - Address does not contain the '@' character. 11 - The
                                  address contains a character that is interpreted as an incorrect delimiter.
                                  12 - Address exceeds maximum allowable length. The maximum length for a user
                                  ID is 64 characters. The maximum length for the entire address is 256
                                  characters. 13 - Address is not using the correct Internet address format.
                                  16 - The user ID might be missing. The error occurred while processing the
                                  print request for file QSYSPRT number 000003 in job 263870/MIRA/QPADEV000C
                                  created on system BLDAS52 at 09/19/03 11:16:15. Recovery  . . . :   Correct
                                  the email address and submit the print request again. If an explanation is
                                  not given for the error code and you are unable to correct the address,
                                                                                                          More...
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

Figure A-6 Addressing an error in the job log

Looking for spooled files owned by QSPLJOB

If you are working on a very busy iSeries server, you may find that there are many entries in the output queue for job logs, many of which may not have any relationship to printing or PDF generation. A way to narrow the list is to focus on spooled files that are owned by QSPLJOB. You can narrow the search even further with the knowledge that the user data field is the same name as the writer.

Follow these steps to look at spooled files owned by QSPLJOB:

1. Enter the WRKSPLF command on a command line and press F4.
2. You see the Work with Spooled Files display. Enter the name of the user, QSPLJOB, and the name of the writer, for example PDFMAIL1, as shown in Figure A-7. Press Enter.

```
                      Work with Spooled Files (WRKSPLF)

Type choices, press Enter.

Select files for:
  User . . . . . . . . . . . . . > QSPLJOB       Name, *CURRENT, *ALL
  Print device . . . . . . . . . > *ALL          Name, *ALL, *OUTQ
  Form type  . . . . . . . . . . > *ALL          Form type, *ALL, *STD
  User data  . . . . . . . . . . > PDFMAIL1      User data, *ALL
  ASP  . . . . . . . . . . . . .   *ALL          1-32, *ALL
Output . . . . . . . . . . . . .   *            *, *PRINT
```

Figure A-7 Finding a spooled file owned by QSPLJOB

3. The layout of the WRKSPLF display is slightly different from the layout used by the WRKOUTQ command. Again you can use F18 to jump to the more recent entries at the bottom of the list. Pressing F11 twice displays the job number to help you distinguish between the three types of jobs.
4. After you find the job log spooled file that you think has the error information you need, use option 5 to view it. The file looks like the one shown in Figure A-6.

Looking for spooled files by job name

Another method to find job logs is to use the WRKJOB command as explained here:

1. Enter the command:

   WRKJOB *jobwtr*

   Here *jobwtr* is the name of the writer.
2. Since there is more than one job for your writer, you see the Select Job panel (Figure A-8). In this example, the PDFMAIL1 writer was started and stopped twice, resulting in two sets of the three types of jobs. The advantage of this method is that it shows the job type on the first panel, so you don't have to look at the job number and try to guess.

   Select the job for which you want to see more details.

```
                              Select Job                          BLDAS52
                                                        11/01/03  10:26:50
Type option, press Enter.
  1=Select

                                                                  Entered
Option  Job         User        Number  Type      -----Status-----  System
        PDFMAIL1    QSPLJOB     287998  PDJ       OUTQ              10/29/03
        PDFMAIL1    QSPLJOB     287997  BATCH     OUTQ              10/29/03
        PDFMAIL1    QSPLJOB     287996  WTR       OUTQ              10/29/03
        PDFMAIL1    QSPLJOB     287995  PDJ       OUTQ              10/29/03
        PDFMAIL1    QSPLJOB     287994  BATCH     OUTQ              10/29/03
        PDFMAIL1    QSPLJOB     287993  WTR       OUTQ              10/29/03
```

Figure A-8 Selecting a job from the WRKJOB command

3. The Work with Job display (Figure A-9) appears. Select option 4 to work with the spooled files associated with this job.

```
                              Work with Job
                                                          System:   BLDAS52
Job:   PDFMAIL1       User:   QSPLJOB        Number:   287995

Select one of the following:

     1. Display job status attributes
     2. Display job definition attributes
     3. Display job run attributes, if active
     4. Work with spooled files

    10. Display job log, if active or on job queue
    11. Display call stack, if active
    12. Work with locks, if active
    13. Display library list, if active
    14. Display open files, if active
    15. Display file overrides, if active
    16. Display commitment control status, if active
                                                                      More..
Selection or command
===>
```

Figure A-9 Work with Job display

4. Select option 5 to display the spooled file. The display is similar to those that you see when you use the other two methods (see Figure A-6 on page 122).

Message PQT4140: PDF Mapping program error codes

If the data being returned to the mapping program is incorrect for some reason, message PQT4140 is issued and the spooled file is held. A copy of the PDF file in error is sent to the PDF administrator if one is specified. Processing continues with the next spooled file.

There are several different types of errors that can cause message PQT4140 to be issued. The text of the message includes a reason code to further qualify the type of message. The message contains a list of reason codes with their explanation. On V5R2, not all possible reasons for the error are included in the accompanying message text. They are listed in the *Using Intelligent Routing* document.

On V5R3, the accompanying message text to PQT4140 includes all possible message codes.

B

PDF conversion completion data queue

This appendix explains how to use a Portable Document Format (PDF) conversion data queue. It also contains information about the format of the data queue.

Using a data queue with PDF conversion

To help clients write PDF mapping programs, Infoprint Server provides a data queue to track when a PDF conversion is complete. Entries are logged in this data queue when a spooled file is converted to PDF and stored in the integrated file system (IFS) or document library system.

A PDF mapping program can use the Receive Data Queue application programming interface (API) (QRCVDTAQ) to determine when a spooled file is converted to PDF. You may use this to trigger a secondary process. For example, after a file is converted to PDF, post it on a Web site. For more information about data queues and the QRCVDTAQ API, see the iSeries Information Center at:

http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm

Using the data queue

If the data queue is specified properly, each time a spooled file is converted to PDF and stored in the IFS or document library system, an entry is sent to the queue. To use the data queue, follow these steps:

1. Use Create Data Queue (CRTDTAQ) command to create the data queue with these values:
   - Maximum message length (MAXLEN) is 752 or higher.
   - Sequence (SEQ) is *FIFO or *LIFO.
2. Specify the name of the data queue on the Print Services Facility (PSF) configuration object. To do this, use the PDFDTAQ parameter on the Create PSF Configuration Object (CRTPSFCFG) or Change PSF Configuration Object (CHGPSFCFG) command. PDFGEN must have a value other than *NONE.

If a data queue's format is not valid and PSF tries to add entries to it, PSF sends a message to the message queue associated with the print writer and continues processing.

It is the user's responsibility to manage the data queues. This includes creating, clearing, and deleting the data queue.

See "Format of the PDF conversion completed data queue" on page 126 for the format of an entry sent to the data queue when a spooled file is converted to PDF.

Additional notes about the PDF data queue

The PDF conversion completion data queue does not contain any entries that deal with the AFP Respool function. Only actions that involve the converted PDF version of the document are logged.

If more than one action is chosen for a file or segment, multiple entries are generated in the PDF data queue. For example, if you choose to e-mail a PDF file and store a copy of it in the IFS, you see two entries in the data queue: one with a return value of 01 and one with a return value of 03.

Format of the PDF conversion completed data queue

Table B-1 shows the format of an entry sent to the data queue when a spooled file is converted to PDF.

Table B-1 PDF data queue entry format

Decimal offset	Hex offset	Type	Description
0	0	CHAR(10)	Function
10	A	CHAR(2)	Record type
12	C	CHAR(2)	Return value
14	E	CHAR(26)	Qualified job name **CHAR(10)** Job name **CHAR(10)** User name **CHAR(6)** Job number
40	28	CHAR(10)	Spooled file name
50	32	CHAR(2)	Reserved
52	34	BINARY(4)	Spooled file number
56	38	CHAR(250)	Routing tag
306	132	CHAR(340)	Path and name of PDF file
646	286	CHAR(2)	Reserved
648	288	BINARY(4)	Path name CCSID
652	28C	CHAR(10)	Mail sender
662	296	CHAR(10)	User data
672	2A0	CHAR(80)	Reserved

Here is a description of each field listed in Table B-1:

- **Function**: Identifies the record type within the function that created the data queue entry. The value for converting a spooled file to PDF is *PDFWTR.
- **Record type**: Identifies the function that created the data queue entry. Valid value is 01, for spooled file conversion record.
- **Return value**: Identifies whether the conversion from Intelligent Printer Data Stream (IPDS) to PDF was successful. The values are:
  - **01** File was successfully converted and stored in a specified folder.
  - **02** File was successfully converted and stored on a specified output queue.
  - **03** File was successfully converted and sent as e-mail.
  - **04** File was successfully converted but the mapping program was specified not to mail the PDF file.
  - **05** File was successfully converted but was not e-mailed because the routing tag or address was not valid.
  - **06** Conversion error.
  - **07** Spooling to output queue failed.
  - **08** Mail servers are unavailable.

  - **09** Placing a file in IFS failed.
  - **10** Mail sender does not have a valid entry in the system distribution directory.
  - **11** Error condition was set by the mapping program.
- **Qualified job name**: Identifies the qualified job name of the job that created the spooled file that was converted to PDF. The first 10 characters are the job name, the next 10 characters are the user name, and the last six characters are the job number.
- **Spooled file name**: Identifies the name of the spooled file that was converted to PDF.
- **Spooled file number**: Identifies the unique number of the spooled file that was converted.

C

Using screen prints to test printer file parameters

When developing a new print application or trying to solve a problem, it is sometimes helpful to test one option or parameter at a time, outside the scope of the larger application. The screen print function provides an easy way to generate quick and easy tests. You can easily use the technique presented in the following section to test any of the multitude of parameters that are available on a printer file.

Generating simple print tests

To generate simple print tests, follow these steps:

1. Enter the CALL QCMD command. Then you see a panel with a number of lines on the bottom. Every time you enter a command, it is echoed in the top section of the panel. That way, when you perform a screen print, it is self-documenting.
2. Create a temporary printer file and give it the characteristics you want to test. For example, to see how a given Advanced Function Presentation (AFP) font looks, you can create a printer file with this command:

   ```
   CRTPRTF QTEMP/TEST DEVTYPE(*AFPDS) FNTCHRSET(C0H200B0 T1V10500)
   ```

3. The screen print function normally uses the QSYSPRT printer file. Tell it to use your custom file by using the Override with Printer File (OVRPRTF) command:

   ```
   OVRPRTF QSYSPRT QTEMP/TEST
   ```

   Note: Instead of creating a printer file, you can override QSYSPRT directly with the parameters that you want to test. We found this problematic because it is easy to lose track of your overrides if you leave the invocation stack. Using the printer file method alleviates this problem since QTEMP/TEST is available as long as you are signed on. If you need the printer file to remain beyond the active session, place it in a permanent or test library.

4. Set up the screen print function to print to your iSeries-attached printer. Traditional 5250 displays had a key marked *Print*. This is not always available with the emulator products. Do *not* use the PC Print Screen key. It sends a copy of the screen image to the printer that you use for workstation printing, bypassing the iSeries. To specify that the iSeries Host Print is used with the screen print, choose one of the following options:
   - Modify your keyboard mapping to assign the Host Print function to one of your PC's keys. Press that key to initiate Host Print.
   - Right-click the emulation display, and a small box of choices appears. Select **Pad2** and click **HostPrn**.

   In either case, you should see the message `Print Operation Complete to the default printer device file.` at the bottom of the display. Press Reset to release the keyboard if it is locked. Reset is usually assigned to the Ctrl key.
5. Move the output to the desired output queue.
6. If necessary, make changes to QTEMP/TEST using CHGPRTF, and redo the test.

Testing longer files

You can use the previous method to test most of the printer file parameters. However, if you need a longer spooled file that fills the page out to 132 characters and 66 lines, use the joblog function. In this case, follow these steps:

1. Create a temporary printer file as described in step 2 in the previous section.
2. Override the printer file as described in step 3 in the previous section. Specify QPJOBLOG instead of QSYSPRT:

   ```
   OVRPRTF QPJOBLOG QTEMP/TEST
   ```

3. Use this command to print your current job's log:

   ```
   DSPJOBLOG * *PRINT
   ```

The last few entries in the job log show the commands that you used to set up the printer file. Once again, it is self-documenting.

Either method can help you to understand the printer file options without developing complete applications. After you determine which parameters produce the output you need, you can add them to your production work.

Testing intelligent routing

You can also use this method to test the logic in a Portable Document Format (PDF) mapping program or PDF mapping object. For example, rather than creating an application with routing tags imbedded, you can create a single spooled file and use the MAILTAG in the user-defined data to insert the routing tag.

The following temporary printer file is used to test an entry in a PDF mapping object. The entry in the mapping object is looking for spooled file MAPME, so that name was used instead of TEST, and the routing tag being tested was for customer 100. To reiterate the test multiple times, the SAVE parameter is set to *YES:

```
CRTPRTF QTEMP/MAPME USRDFNDTA('MAILTAG(100)') OUTQ(MAPOBJ) SAVE(*YES)

OVRPRTF QSYSPRT QTEMP/MAPME
```

D

Spooled file types for PDF conversion

The iSeries supports many different types of spooled files. The data stream within a given spooled file is determined by the DEVTYPE parameter on the printer file that was used to generate the output. This appendix describes the various spooled file types that are supported on the iSeries server and their impact on Portable Document Format (PDF) conversion.

SCS

Systems Network Architecture (SNA) Character Set (SCS) is the default spooled file type used on the iSeries server. It it mainly text, but it may contain some simple formatting information, such as change in CPI or LPI, if a data description specification (DDS) is used.

SCS spooled files can be converted PDF or Advanced Function Presentation (AFP) using Infoprint Server. The entire spooled file is processed as one file. There is no option to split an SCS spooled file.

IPDS

The Intelligent Printer Data Stream (IPDS) value for spooled files was originally intended for the original IPDS printers that were twinaxially attached to the iSeries and were supported without the use of Print Services Facility (PSF). This type of spooled file supports a subset of the full IPDS capability. Your application can specify different fonts printer resident fonts, some image types, and barcodes. Overlays, page segments, and all points addressibility are some functions that are not supported if the spooled file has DEVTYPE(*IPDS).

IPDS spooled files can be converted PDF or AFP using Infoprint Server. The entire spooled file is processed as one file. There is no option to split an IPDS spooled file.

AFPDS

Spooled files that are created using the AFP data stream have the greatest amount of flexibility with respect to formatting on the iSeries server. They can use printer resident fonts or host font resources, overlays, page segments, barcodes, and graphics.

Advanced Function Presentation data stream (AFPDS) spooled files can contain index tags, which Infoprint Server uses to split the spooled file into multiple segments, or to generate PDF bookmarks. Each segment of a spooled file can be processed in a different manner if a PDF mapping object or PDF mapping program is used. This is referred to as *intelligent routing*.

Index tags, or routing tags, can be inserted into AFPDS in several ways:

- **DDS**: STRPAGGRP and ENDPAGGRP keywords
- **AFP Toolbox**: Begin Group and End Group
- **CRTAFPDTA**: If the original spooled file is *LINE, you can use CRTAFPDTA to convert the spooled file to AFPDS and to generate index tags at the same time.

AFPDS spooled can be converted to PDF or respooled as AFP using Infoprint Server. The entire spooled file can be processed as one, or you can split it based on index tags into multiple segments.

LINE

Spooled files that are generated with DEVTYPE(*LINE) contain little formatting. Usually, the first character is reserved as a forms control character that controls skipping and spacing. Sometimes a second character is used to map to fonts. Other than that, the rest of the spooled file contains only text.

The formatting for *LINE data is accomplished by an external object known as a *page definition*. The page definition contains rules to convert the simple lines of text to a fully formatted, all points addressable page. A *form definition* is used to control how the output is presented on the device, such as which paper source to use and the inclusion of overlays.

Spooled files that are generated using *LINE data can be printed directly on a printer that supports SCS, IPDS, or AFPDS. If you want to print line data on an ASCII device that is configured to use Host Print Transform, you must use the CVTLINDTA parameter on the printer file. This converts the application output to AFPDS as the spooled file is generated.

Printer files that are generated as *LINE can be converted to PDF. The AFP Respool function does not work for these types of files. If your application is generating *LINE data for use with a page definition, consider using CVTLINDTA on the printer file to generate the spooled file as *AFPDS.

Spooled files that contain *LINE data do not have index tags, and consequently, cannot be split. If you have an application that is generating *LINE data, and want to split it into multiple segments for use with a mapping program or mapping object, use the CRTAFPDTA command. This process is described in *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250.

AFPDSLINE (mixed)

Spooled files that are generated using a device type of *AFPDSLINE primarily contain line data, with the addition of a limited set of AFPDS structured fields that are used to switch overlays, or other formatting features. This type of spooled file is not commonly used on the iSeries server, but is included for compatibility with other platforms.

Spooled files that contain AFPDSLINE can be converted to PDF. The AFP Respool function does not work for these types of spooled files. You must use CRTAFPDTA first.

USERASCII

Spooled files that are created with DEVTYPE(*USERASCII) can contain any type of data that the user wants. The iSeries does not check the validity of the data. This type of spooled file is commonly used for ASCII data streams, such as PCL, PostScript, or PDF, but it is not limited to those. PDF spooled files that are generated by Infoprint Server have a spooled file type of *USERASCII.

USERASCII spooled files are not converted to PDF or to AFPDS by Infoprint Server directly. It is possible to use another feature of Infoprint Server known as *Transform Manager* to convert PCL, PostScript, or PDF to AFP, which can than be further transformed to PDF or AFP.

For more information about Transform Manager, see the following documents:

- *Infoprint Server for iSeries: User's Guide*, G544-5775
- *Infoprint Server for iSeries: User's Guide*, G544-5775
- *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250

E

PDF mapping object versus PDF mapping program comparison

Table E-1 compares the functions available using a Portable Document Format (PDF) mapping object compared to using a PDF mapping program.

Table E-1 Differences between PDF mapping program and PDF mapping object

Function	PDF mapping object	PDF mapping program
Skill level	The user understands the business application.	User understands the business application and programming in RPGLE or other high level language.
Routing data	Must be entered manually into the PDF mapping object. Optionally use an application programming interface (API) to add entries to the PDF mapping object.	May exist in standard iSeries database files.
Logic	Very structured.	It is up to the discretion of the programmer to include any logic based on information in the input information data structure, including additional data available in other database files.
Addresses (To, CC, BCC)	May be entered directly in the PDF mapping object field or referenced in a stream file.	Must be included directly in the output information data structure.
Message text	Use the system default text or the exact stream entered in the PDF mapping table.	May build a highly informative message by concatenating information from a variety of sources.
Spooled file attachment name	Default naming convention or exact name as entered in the PDF mapping table. Limited to 80 characters.	The default naming convention or any string the programmer constructs using available information.
Subject	Use the system default text or the exact stream entered in the PDF mapping table.	May build a highly informative message by concatenating information from a variety of sources.

Function	PDF mapping object	PDF mapping program
Encryption	Encryption only applies to the e-mailed version of the PDF file.	A programmer can copy the temporary file to have an encrypted version available for storage as a stream file in the integrated file system (IFS).
Password length	Passwords are limited to a 10-character field.	Passwords can be up to 32 characters in length.
Master password	The Option to use an automatically generated master password is available.	Automatic master password generation is not available.
Data updates	Writers that use the given PDF mapping object must be ended.	Data in standard iSeries database file can be updated regardless of whether the PDF writers are active.
Mail sender	The mail sender can be modified as part of defining a PDF map action for Mail = *YES. Otherwise the value from the spooled file's MAILSENDER tag or the Print Services Facility (PSF) configuration object PDFSENDER parameter is used.	Uses the value from the spooled file MAILSENDER tag or the PSF configuration object PDFSENDER parameter.
Stream file	The path plus the name are limited to 255 characters.	No practical limit to the path plus the name.
Invalid routing tag	The recommendation in the manual is to build an extra entry in table to handle unmatched routing tags.	Programmer can take any action, including using the Error code to send a message automatically to the PDF administrator.
Body and additional attachment files	A full path name must be entered in the panels. A maximum of four of each type can be specified.	Enter the path name once as the DIRPTH, Directory for files, field. Individual files names can then be entered without the full path by assigning a value of '1' to BDYUSEx or to ATTUSEx. Any number of body or attachments can be specified.

F

Version 5 enhancements to Infoprint Server

This appendix describes the enhancements and changes made in Infoprint Server with Version 5 Release 2 (V5R2) and Version 5 Release 3 (V5R3).

Infoprint Server V5R2

Infoprint Server V5R2 provides the following enhancements over Version 5 Release 1 (V5R1).

SMTP mail server support

Previously, the only way to send e-mail from Infoprint Server was through the SNDDST command. With Infoprint Server V5R2, you can specify that Infoprint Server uses an Simple Mail Transfer Protocol (SMTP) server to send your e-mail.

Control over e-mail content

Previously with Infoprint Server, you could only use a Portable Document Format (PDF) mapping program to map mail tags when you specified a keyword, such as a customer number, for the mail tag. Now you can use the PDF mapping program to interpret mail tags, specify the subject text, and add a customized message to the beginning of each e-mail. If you use an SMTP mail server to send the e-mail, you can also specify a file to use in the body of the e-mail, add carbon copy (CC) and blind copy (BCC).

ASCII printing transform enhancements

The PDF, Printer Control Language (PCL), and PostScript to Advanced Function Presentation (AFP) transforms that are shipped with Infoprint Server V5R2 were enhanced for better performance. The PCL to AFP transform now supports PCL 6.

PDF bookmarks

When transforming a file to PDF, Infoprint Server can now place index tags at group boundaries and return one PDF file. This lets you easily navigate the file when viewing it on your workstation.

Reduced PDF file sized

Infoprint Server V5R2 lets you generate a PDF file without embedding the fonts. This way, you can produce smaller PDF files.

Improved Create AFP Data (CRTAFPDTA) command functions

You can now enjoy easier file management with the CRTAFPDTA command. It lets you tell Infoprint Server to delete the output stream files after it merges them together. Two new parameters make it easier to identify the input file.

Interactive access to PDF and e-mail functions

Infoprint Server for iSeries, together with iSeries Access V5R2, lets you use the fully graphical iSeries command interface to run PDF and e-mail functions interactively. You can send output by e-mail in one step. Plus you can write single or multiple output files to the integrated file system (IFS).

Web access support

Infoprint Server for iSeries, when installed with iSeries Access, enables direct viewing of spooled files via PDF and PDF printing from your browser.

2003 enhancements to V5R2

The following enhancements became available to V5R2 in mid 2003 via program temporary fixes (PTFs) for Infoprint Server for iSeries and Print Services Facility (PSF) for OS/400. For a current list of PTFs, refer to iSeries Software Knowledge Base document 23381552 at:

http://www-912.ibm.com/s_dir/slkbase.nsf/slkbase/

For further assistance in obtaining PTFs, contact your local IBM Support Center.

Intelligent routing

Currently, the Infoprint Server PDF subsystem enables you to transform a print file or segment of a print file to PDF. Then you specify one of these actions:

- Spool as PDF to an output queue
- Store as a PDF file in the IFS
- Send as an attachment to an e-mail

All print files or print file segments in a specified processing run (driven by output queue) had to have the same action. All were spooled to an output queue or all were sent as e-mail.

With intelligent routing, you can perform any combination of the these actions with each output PDF file. If the input spooled file contains group tags, you can segment the print file at group boundaries and distribute each segment in any combination of the following ways:

- Spool it to an output queue as PDF
- Store it as a stream file in the IFS
- Send it as e-mail
- Respool it as AFP (for print, fax, or other operation)

In addition, you can instruct the intelligent routing function to take multiple actions on a given file or segment. For example, you may define one segment where the customer or user wants to e-mail or print a copy. You can treat each segment differently.

Print files or print file segments can be respooled separately. This enables both downstream print and fax operations.

Definition of intelligent routing functions can be done either with the existing user exit interface or with the new set of commands (PDF mapping interface) for setup and profiling.

PDF mapping interface

The new PDF mapping interface significantly enhances the ease and control of deploying document distribution transformations. Currently, all but the simple PDF distribution functions require a user exit program to provide key information during PDF processing. The PDF subsystem looks to the exit program for this input.

With the new PDF mapping interface, a new set of commands enables an end user to create a set of profiles that provide the information required. This data can be created without application programming.

The PDF mapping interfaces captures two key elements. First, a table of entries defines or identifies the target print file or print file segment. You can use such data fields as output queue, filename, user, user data, form type, and group code to select this match.

Second, a set of action entries is set up for each target match. Here, you define the data supporting e-mail, respool as AFP, respool as PDF, and write into IFS. For example, if one of the operations on the selected print file or print file segment is e-mail, then you can specify the e-mail data requirements of address, subject, message text, CC addressed, BCC addresses, Reply to addresses, file attachments, and PDF characteristics, such as encryption.

For each selected print file or print file segment, you can define multiple operations. For example, an invoice segment within a print file can be profiled to be e-mail to several IDs, sent by CC to several others, written to IFS for Web access, and queued as AFP to a remote printer queue for printing.

AFP Respool

With AFP Respool, you can respool a print file or print file segment (as AFP) after PDF processing. You have the option to physically print a spooled file and respool it as AFP or simply respool it as AFP without printing.

In conjunction with new intelligent routing functions, AFP Respool enables a separate operation to be done on a print file or print file segment. This is especially useful with segments. You can route a segment (for example, specific customer invoice, portion of a sales report) to a queue in standard AFP format for printing or other operations such as fax.

PDF distribution recovery

This function lets you specify an administrator's e-mail address to use when you transform spooled files to PDF. If there is a problem with the creation or distribution of the PDF, PSF sends an e-mail with the error information to this address. When possible, the PDF file is also attached to the e-mail. This enables error recovery operations to take place.

PDF encryption

Currently, PDF files created with Infoprint Server are unencrypted and not password-protected. With this enhancement, you can specify security options, such as encryption and password protection, for the PDF file or files created for a print file or print file segment.

Files designated for encryption have a master and use password associated with them. The document or report data is physically encrypted within the PDF, and a password is required to decrypt and view the data.

In addition, several related security functions are enabled. These include blocking the ability to change, print, copy, or annotate the PDF file.

Full color image support

You can now design and distribute documents with full-color image content. Full color architecture is already enabled in OS/400 V5R1, but to date, the ability to tap into this function has been limited. AFP is the integrated page architecture on iSeries. AFP image support was already extended to include a new image type (IOCA FS45 specification). This is a specification providing for high-resolution, full-process color images that are highly compressed. The ASCII comparison is JPEG.

Several tools are available to capture full color FS45 images in your documents. With the latest enhancements, Infoprint Designer for iSeries is one of these tools.

Now, you can produce documents for distribution containing full color content. Infoprint Server is enhanced to support FS45 color processing in the PDF subsystem. This means that iSeries documents containing color can be transformed to PDF with the color intact, and available for viewing, printing or both.

2D barcode support

PDF support in Infoprint Server is enhanced to support new two-dimensional barcode symbologies. Those symbologies are UPS Maxicode, PDF417, and Datamatrix. Documents encoded with these barcodes are transformed from AFP to PDF with the barcodes.

Among the tools that were previously available to encode these new barcodes in iSeries output are Page Printer Formatting Aid (PPFA/400, 5798-AF3), OS/400 Data Description Specifications (DDS), and Infoprint Designer. Support via PPFA/400 was implemented in V5R1. Support with DDS in V5R2 and support for Infoprint Designer are covered in this announcement.

PDF naming and routing

Currently, Infoprint Server creates the 103036_000013_QPDCDVV_07172002_000001.PDF generated file name for PDF files. This file name is actually a combination of the system file name, job name, date, and spooled file number. In many cases, it is not very usable in PDF deployments. In addition, control was limited over the actual IFS directory where the PDF file was written.

With this set of enhancements, you now have discrete control over the PDF file name and IFS location for each print file or print file segment transformed to PDF for IFS routing.

PDF orientation

When you use Infoprint Server to convert a landscape spooled file to PDF, the pages are automatically rotated appropriately when you view it. Previously, landscape documents opened as rotated 90 degrees in the viewer, reflecting the correct orientation for printing. Now, you can ensure the document opens in the logical orientation for viewing.

PDF file size improvements

Several improvements are included in the PDF file content. This results in significant savings in file size. Smaller PDF file sizes mean less processing since they are produced with less network impact as they are distributed.

Planet barcode

The Planet barcode, a variant of Zip Code, is now supported with Infoprint Server PDF processing.

Changes for V5R3

Most of the information in this appendix applies to both V5R2 with the intelligent routing and PDF mapping object PTFs applied, and to V5R3, as noted. This section summarizes differences that you should know:

- Many changes are made to the PSF configuration object:
  - In V5R2, you specify PDFADMIN, AFPSAVE, AFPOUTQ, and PDFMAPOBJ as parameters within the PDFDFNOPT. There is a limit of 30 to the size of the string for any option in PDFDFNOPT. This limits the actual value of PDFADMIN to a maximum length of 20 characters.

    In V5R3, PDFADMIN, AFPSAVE, AFPOUTQ, and PDFMAP are actual parameters in the PSF configuration object. The value for PDFADMIN can be up to 132 characters long.
  - In V5R2, if you use a PDF mapping object, you must specify QSYS/QPQMAPEXIT for the PDFMAPPGM. In V5R3, you specify PDFMAPPGM(*IBMPGM).
  - In V5R2, you can only select *MAIL, *SPLF, or *STMF for PDFGEN. Based on this selection, you are prompted for additional parameters that pertain only to the action selected. For example, if you select *MAIL, you are prompted for the mail server type, default sender, and PDF mapping program name, but you cannot specify a default path for any PDF steam files that are generated using the PDF mapping program or PDF mapping object. Nor can you specify a default output queue for PDF spooled files.

    This is described further in 2.1.1, “Creating a PSF configuration object” on page 12.

    In V5R3, this restriction is eliminated.
- In V5R2, the PDF mapping object has an object type *USRIDX with an attribute of PDFMAP. In V5R3, the PDF mapping object has an object type *PDFMAP.

Migration considerations

PSF configuration objects and PDF mapping objects created on an iSeries running V5R2 continue to work when the operating system is upgraded to V5R3. The objects work unmodified.

When you access a PSF configuration object that was created on V5R2 using PSFDFNOPT for some of the parameters, they are automatically converted to the equivalent keyword. For example PSFDFNOPT(‘PDFADMIN(user@ibm.com)’) is converted to the parameter PDFADMIN(user@ibm.com).

PDF mapping objects created on V5R2 as *USRIDX objects continue to work. Even if you access them to make changes they remain as that object type. New PDF mapping objects created on V5R3 are created as the new object type, *PDFMAP. If you have both a *USRIDX and *PDFMAP of the same name on your system, PSF will use *PDFMAP.

G

Infoprint Server performance

This appendix contains selected results from a V5R3 Infoprint Server segmentation, intelligent routing, and image performance evaluation. The performance evaluation was performed by the IBM Printing Systems Division in Boulder, Colorado.

Environment

V5R3 Infoprint Server performance was measured using the secondary partition on an iSeries system, Model 820 Type 9406. The system has a Commercial Processing Workload (CPW) Rating of 3700 and four processors (CPU) of Type 2438. The secondary partition has 2.75 processors (CPU) and was completely dedicated to printing with no other processes active except for measurement.

Hardware

The iSeries system partition configuration consists of:

- System Model 820 Type 9406
- Secondary Partition
- 2.75 Processors (CPU) Type 2438
- 2 GB Memory
- 105.2 GB Disk Storage
- Disk Model 070 and 074 Type 6718

Software

The V5R3 software was at the preliminary GA level. It is believed to represent GA level performance. Software parameters that are relevant to performance were set to the following settings unless stated otherwise:

- 100 MB Spool (QSPL) Storage
- Print While Convert (PWC)=YES
- Font Inline=NO

Methodology

The parameters in Table G-1 determine V5R3 Infoprint Server segmentation, intelligent routing, and Image performance.

Table G-1 Performance parameters

Term	Definition
AFPDS to IPDS Transform Time	This is the elapsed time of Advanced Function Presentation data stream (AFPDS) to Intelligent Printer Data Stream (IPDS) transforming between the first page of the case and the last page of the case. It is represented in seconds.
IPDS to PDF Transform Time	This is the elapsed time of IPDS to PDF transforming between the first page of the case and the last page of the case. It is represented in seconds.
AFPDS to IPDS Transform Throughput Rate	This is the rate of AFPDS to IPDS transforming in pages per minute (PPM) between the first page of the case and the last page of the case. It is calculated from the AFPDS to IPDS transform time and number of pages.
IPDS to PDF Transform Throughput Rate	This is the rate of IPDS to PDF transforming in PPM between the first page of the case and the last page of the case. It is calculated from the IPDS to PDF transform time and number of pages.
PDJ Time	This is the elapsed time of PDJ process. It is represented in seconds. For detailed explanation about PDJ, see “The PDJ (print driver) job” on page 118.
Total Time	This is the elapsed time between case submission and approximately printing of the last page of the case. This is the total job time. It is represented in minutes:seconds.

Term	Definition
Output Data Rate	This is the rate of producing PDF outputs. It is produced by dividing the PDF output file size by the total job time. It is represented in bytes per second.
CPU Times	This is the use of the CPU time during the writer (WTR) or the AFPDS to IPDS transforming, the printing (PDJ), and the batch (BCH), or the IPDS to PDF transforming phases. For a detailed explanation about WTR, BCH, and PDJ, see “Infoprint Server jobs” on page 118. For segmentation and intelligent routing, PDJ includes sending data to printer and distributing PDF segments. For Image, PDJ just sends data to the printer. The total CPU is the sum of the CPU time for WTR, PDJ, and BCH. They are represented in seconds.
%CPU Utilizations	This is the percent CPU utilization during the writer (WTR), batch (BCH), and printing (PDJ) phases. For a detailed explanation about WTR, BCH, and PDJ, see “Infoprint Server jobs” on page 118. The percent CPU utilization for WTR is produced by dividing the CPU time for WTR by the AFPDS to IPDS transform time. The percent CPU utilization for BCH is produced by dividing the CPU time for BCH by the IPDS to PDF transform time. For segmentation and intelligent routing, the percent CPU utilization for PDJ is produced by dividing the CPU time for PDJ by the sum of sending data to printer time and distribution time. For Image, the percent CPU utilization for PDJ is produced by dividing the CPU time for PDJ by the sending data to printer time. The percent CPU utilization for Total is produced by dividing the total CPU time for WTR, BCH, and PDJ by the total job time.
CPU Times Per Page	This is the use of the CPU times (milliseconds) per page during the writer (WTR), batch (BCH), and printing (PDJ) phases. For a detailed explanation about WTR, BCH, and PDJ, see “Infoprint Server jobs” on page 118. The CPU time per page is produced by dividing the CPU time by number of pages. It is represented in milliseconds per page.

The procedure for Infoprint Server V5R3 measurements begins with deactivation of all processes other than the printing and PSF Trace program. The files to be measured are already placed on the spool. Each measurement was made using this procedure:

1. Print an easy case to make sure the writer has started properly.
2. Start the PSF Trace program to record start and stop times for AFPDS to IPDS transform and IPDS to PDF transform.
3. Start the device and the writer.
4. Start the Work With System Status and Work With Active Jobs commands to gather information about CPU use while printing.
5. Release the spooled file to be measured and start a timer at the same time.
6. Record the approximate time at which the last page was printed.
7. End the Work With System Status and Work With Active Jobs commands to stop gathering information about CPU use.
8. Stop the PSF Trace program.
9. End the writer and the device.
10. Retrieve information from the data recorded by PSF Trace and two commands.

Performance cases

The following sections describe the performance cases used for Infoprint Server V5R3 segmentation, intelligent routing, and Image performance evaluation.

Segmentation cases

For segmentation measurements, eight performance cases were used. All of them are native iSeries Super Sun Seeds Invoice applications. They were based on a Super Sun Seeds invoice file, which has 10 pages and breaks down into six segments.

Four of these eight performance cases were converted to PDF. All segments were written to an integrated file system (IFS) as described in Table G-2.

Table G-2 Segmented cases for storage in the IFS

Case name	Pages printed	Total PDF size (bytes)
IFS1	1	49,962
IFS100	100	3,080,700
IFS1000	1,000	30,807,000
IFS2000	2,000	61,614,000

Table G-3 lists the detailed information for each segment.

Table G-3 Details for segments stored in the IFS

Segment	Segment name	Pages	Convert to PDF and write to IFS	PDF size (bytes)
1	BestWay	1	Yes	49,962
2	CleanTruck	2	Yes	54,553
3	Stork	2	Yes	51,323
4	Fed Ex	2	Yes	51,268
5	UPS	2	Yes	52,023
6	Airborne	1	YES	48,941

The other four performance cases were converted to PDF and all segments were e-mailed. These are shown in Table G-4.

Table G-4 Segmented cases for e-mail

Case name	Pages printed	Total PDF size (bytes)
e-mail1	1	49,962
e-mail100	100	3,080,700
e-mail1000	1,000	30,807,000
e-mail2000	2,000	61,614,000

Table G-5 lists the detailed information for each segment.

Table G-5 Details for e-mailed segments

Segment	Segment name	Pages	Convert to PDF and e-mail	PDF size (bytes)
1	BestWay	1	Yes	49,962
2	CleanTruck	2	Yes	54,553
3	Stork	2	Yes	51,323
4	Fed Ex	2	Yes	51,268
5	UPS	2	Yes	52,023
6	Airborne	1	Yes	48,941

Intelligent routing cases

For intelligent routing measurements, four print jobs performance cases were used. All of them were native iSeries Super Sun Seeds Invoice applications. They were based on a Super Sun Seeds invoice file, which has 10 pages and breaks down into six segments. These are described in Table G-6.

The only two differences between these performance cases and the segmentation cases are that each segment can have a different destination and then one destination.

Table G-6 Intelligent routing cases

Case name	Pages printed	Total PDF size (bytes)
SSSMAP1	1	99,924
SSSMAP100	100	3,580,320
SSSMAP1000	1,000	35,803,200
SSSMAP2000	2,000	71,606,400

Table G-7 lists the detailed information for each segment.

Table G-7 Details for intelligent routing segments

Segment	Segment name	Pages	Convert to PDF and e-mail	Convert to PDF and write to IFS	PDF size (bytes)
1	BestWay	1	Yes	Yes	49,962
2	CleanTruck	2	Yes	No	54,553
3	Stork	2	No	Yes	51,323
4	Fed Ex	2	Yes	No	51,268
5	UPS	2	Yes	No	52,023
6	Airborne	1	YES	NO	48,941

Image Object Content Architecture Function Set 45 (IOCA FS45) image cases

Eight print jobs were used. Four of them were based on the same 4-square inch (2 inch by 2 inch) full color JPEG compressed image in IOCA FS45 format, but had different number of pages. The case name for these jobs started with “SQIN04”. The other four jobs were based on the same 16-square inch (4 inch by 4 inch) full color JPEG compressed image in IOCA

FS45 format, but had different number of pages. The case name for these jobs starts with “SQIN16”. Table G-8 summarizes these jobs.

Table G-8 Image cases

Case name	Pages printed	Total PDF size (bytes)
SQIN041	1	64,641
SQIN0410	10	636,943
SQIN04100	100	6,359,979
SQIN041000	1,000	63,597,893
SQIN161	1	186,466
SQIN1610	10	1,855,185
SQIN16100	1,000	18,542,380

Results

This section contains tables of performance information and analysis. For descriptions about the table headings, refer to “Methodology” on page 146. For descriptions about the cases, see “Performance cases” on page 147.

All the measurements were made with Print While Convert (PWC)=YES, Spool (QSPL) Storage=100M, Font Inline=NO, unless otherwise stated. PWC=YES causes printing or IPDS to PDF transform to start while AFPDS to IPDS transform is happening.

Note: For some cases, the performance results for the same cases are separated into two tables (1 of 2) and (2 of 2). The first of these focuses on results for elapsed times and rates. The second table focuses on CPU usage results.

Segmentation: Write to IFS

All segments were written to the IFS. For descriptions about the table headings, see “Methodology” on page 146. For descriptions about the cases, refer to “Performance cases” on page 147. Table G-9 and Table G-10 are for the same cases.

Table G-9 Segmentation performance: Write to IFS results (Part 1 of 2)

iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.							
Case name	**AFPDS to IPDS time (sec)**	**IPDS to PDF time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (mm:ss)**	**Output data rate (bytes/sec)**
IFS1	0.070	0.299	857	201	1	00:01	49962
IFS100	0.290	13.574	20690	442	35	00:35	88020
IFS1000	1.055	134.218	56872	447	445	07:25	69229
IFS2000	5.490	267.770	21858	448	984	16:25	62552

Table G-10 Segmentation performance: Write to IFS results (Part 2 of 2)

iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
Case name	**CPU time (sec)**				**%CPU utilization**				**CPU time per page (ms/page)**			
	WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
IFS1	<0.1	0.1	0.1	<0.3	*	10.0	33.4	*	*	100.0	100.0	*
IFS100	0.1	3.4	9.9	13.4	34.5	10.0	72.9	38.3	1.0	34.0	99.0	134.0
IFS1000	0.6	85.2	98.5	184.3	56.9	19.0	73.4	41.4	0.6	85.2	98.5	184.3
IFS2000	1.3	325.5	196.7	523.5	23.7	33.0	73.5	53.1	0.7	162.8	98.3	261.8

Segmentation: E-mail

All the segments were e-mailed. For descriptions about the table headings, refer to “Methodology” on page 146. For descriptions about the cases, refer to “Performance cases” on page 147. Table G-11 and Table G-12 are for the same cases.

Table G-11 Segmentation performance: E-mail results (Part 1 of 2)

iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.							
Case name	**AFPDS to IPDS time (sec)**	**IPDS to PDF time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (mm:ss)**	**Output data rate (bytes/sec)**
e-mail1	0.048	0.305	1250	197	1	00:01	49962
e-mail100	0.322	13.418	18634	447	40	00:40	77018
e-mail1000	1.142	137.392	52539	437	480	08:00	64181
e-mail2000	3.817	272.731	31438	440	1189	19:49	51820

Table G-12 Segmentation performance: E-mail results (Part 2 of 2)

iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
Case name	**CPU time (sec)**				**%CPU utilization**				**CPU time per page (ms/page)**			
	WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
e-mail1	<0.1	<0.1	0.1	<0.3	*	*	32.8	*	*	*	100.0	*
e-mail100	0.1	3.5	9.9	13.5	31.1	9.0	73.8	33.8	1.0	35.0	99.0	135.0
e-mail1000	0.7	83.4	97.7	181.8	61.3	17.0	71.1	37.9	0.7	83.4	97.7	181.8
e-mail2000	1.2	340.4	196.8	538.4	31.4	29.0	72.2	45.3	0.6	170.2	98.4	269.2

Intelligent routing

Intelligent routing refers to the set of capabilities of Infoprint Server to profile recipients for print files or print file segments and define multi-channel (print, Web, e-mail, fax) delivery options down to the segment level. For example, a given segment can be profiled for different

recipients with multiple delivery options for the same recipient. The multi-channel options in intelligent routing include:

- E-mail
- Write to IFS for client or Web access
- Re-queue as AFP for printing or for “downstream” options such as fax or archive
- Re-queue as PDF for printing

For descriptions about the table headings, refer to “Methodology” on page 146. For descriptions about the cases, refer to “Performance cases” on page 147. Table G-13 and Table G-14 are for the same cases.

Table G-13 Intelligent routing performance: Results (Part 1 of 2)

Intelligent routing performance. iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.							
Case name	**AFPDS to IPDS time (sec)**	**IPDS to PDF time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (mm:ss)**	**Output data rate (bytes/sec)**
SSSMAP1	0.118	0.305	509	197	1	00:01	99924
SSSMAP100	1.372	13.365	4373	449	45	00:45	79563
SSSMAP1000	11.784	136.214	5092	441	495	08:15	72330
SSSMAP2000	34.965	273.014	3432	440	1117	18:39	60224

Table G-14 Intelligent routing performance: Results (Part 2 of 2)

Intelligent routing performance. iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
Case name	**CPU time (sec)**				**%CPU utilization**				**CPU time per page (ms/page)**			
	WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
SSSMAP1	<0.1	0.1	0.2	<0.4	*	10.0	65.6	*	*	100.0	200.0	*
SSSMAP100	0.5	4.0	9.8	14.3	36.4	9.0	73.3	31.8	5.0	40.0	98.0	143.0
SSSMAP1000	4.8	70.3	97.6	172.7	40.7	14.0	71.7	34.9	4.8	70.3	97.6	172.7
SSSMAP2000	9.6	238.0	196.0	443.6	27.5	21.0	71.8	39.6	4.8	119.0	98.0	221.8
***** Transform times and CPU times are too small to support accurate %CPU utilizations and CPU times per page.												

For segmentation: Write to IFS, e-mail, and intelligent routing

As shown by the results in Table G-9 on page 150 through Table G-14, AFPDS to IPDS transform times are *not* proportional to the number of pages in the case. IPDS to PDF transform times *are* proportional to the number of pages in the case.

PDJ times are almost the same as the total job times. PDJ process starts a few milliseconds after the job starts and ends when the job ends. The output data rates are not similar because the total job times are not proportional to the number of pages in the case.

The CPU times for BCH are proportional to the number of pages in the case. %CPU utilizations for PDJ are low.

IOCA FS45 image

Note the following points for the IOCA FS45 format:

- iSeries' overall image architecture is IOCA.
- Color definition is called FS45.
- FS45 is a highly-compressed and high-resolution color.
- Infoprint Server supports this new full color image IOCA FS45.
- It is packaged as an iSeries page segment object.

Unlike the segmentation and intelligent routing cases, these image cases are not segmented. For descriptions about the table headings, see “Methodology” on page 146. For descriptions about the cases, see “Performance cases” on page 147. Table G-15 and Table G-16 are for the same cases.

Table G-15 IOCA FS45 image performance results (Part 1 of 2)

IOCA FS45 image performance, 4 and 16 square inches. iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.							
Case name	**AFPDS to IPDS time (sec)**	**IPDS to PDF Time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (mm:ss)**	**Output data rate (bytes/sec)**
SQIN041	0.044	1.070	1364	56	2	00:02	32321
SQIN0410	0.520	9.891	1154	61	10	00:10	63694
SQIN04100	1.843	96.619	3256	62	97	01:37	65567
SQIN041000	39.542	966.789	1517	62	975	16:15	65229
SQIN161	0.051	3.815	1176	16	4	00:04	46617
SQIN1610	0.675	37.048	889	16	38	00:38	48821
SQIN16100	10.220	370.129	587	16	372	06:12	49845

Table G-16 IOCA FS45 image performance results (Part 2 of 2)

IOCA FS45 image performance, 4 and 16 square inches. iSeries Model 820, Type 9406, 2.75 Type 2438 Processors. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
	CPU time (sec)				**%CPU utilization**				**CPU time per page (ms/page)**			
Case name	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
SQIN041	<0.1	0.1	0.9	<1.1	*	5.0	84.1	*	*	100.0	900.0	*
SQIN0410	0.1	0.1	9.5	9.7	19.2	1.0	96.0	97.0	10.0	10.0	950.0	970.0
SQIN04100	0.4	0.6	94.1	95.1	21.7	0.6	97.4	98.0	4.0	6.0	941.0	951.0
SQIN041000	4.2	3.8	941.0	949.0	10.6	0.4	97.3	97.3	4.2	3.8	941.0	949.0
SQIN161	<0.1	0.1	3.6	<3.8	*	2.5	94.4	*	*	100.0	3600.0	*
SQIN1610	0.1	0.1	35.8	36.0	14.8	0.3	96.6	94.7	10.0	10.0	3580.0	3600.0
SQIN16100	1.1	1.4	357.6	360.1	10.8	0.4	96.6	96.8	11.0	14.0	3576.0	3601.0

***** For SQIN041 and SQIN161 case, transform times and CPU times are too small to support accurate %CPU utilizations and CPU times per page.

For 4 and 16 square-inch images

As shown in Table G-15 and Table G-16, measured AFPDS to IPDS transform times vary from measurement to measurement when the times are too short to be accurate. The times in this column are from one set of measurements. Larger and smaller times were observed in other similar measurements.

AFPDS to IPDS transform times are not proportional to the number of pages in the case. However, IPDS to PDF transform times are proportional to the number of pages in the case.

PDJ times are almost the same as the total job times. PDJ process starts a few milliseconds after the job starts and ends when the job ends.

The CPU times for BCH are proportional to the number of pages in the case. %CPU utilizations for PDJ are very low. %CPU utilizations for BCH are very high and close to 100%.

For 4 square-inch images

The output data rates for 4 square-inch images are similar because the total job times are almost proportional to the number of pages in the case. The exception is for the SQIN041 case, which is too small and difficult to measure and evaluate accurately.

For 16 square-inch images

The output data rates for 16 square-inch images are similar because the total job times are almost proportional to the number of pages in the case.

4 square inch versus 16 square inch

The following observations were made when comparing performance cases with the same number of pages:

- For IPDS to PDF transform time, 16 square-inch cases take about four times as long as the 4 square inch.
- For PDJ and Total job time, 16 square-inch cases take about four times as long as the 4 square inch.
- For CPU times for BCH, 16 square-inch cases take about four times as long as the 4 square inch.

Comparison of AFPDS to IPDS transforming time and printing rate

Table G-17 compares AFPDS to IPDS transforming times and printing rates for segmentation write to IFS, segmentation e-mail, and intelligent routing. For detailed descriptions about the differences between segmentation and intelligent routing cases, see “Performance cases” on page 147.

Table G-17 AFPDS to IPDS transforming time and printing rates

PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.						
Number of pages	**Segmentation, write to IFS**		**Segmentation, e-mail**		**Intelligent routing**	
	Time	**Rate**	**Time**	**Rate**	**Time**	**Rate**
1	0.070	857	0.048	1250	0.118	509
100	0.290	20690	0.322	18634	1.372	4373
1,000	1.055	56872	1.142	52539	11.784	5092
2,000	5.490	21858	3.817	31438	34.965	3432

As shown by the results in Table G-17, measured AFPDS to IPDS transform times vary from measurement to measurement when the times are too short to be accurate. The times in this column are from one set of measurements. Larger and smaller times were observed in other similar measurements. AFPDS to IPDS transform times are not proportional to the number of pages in the case.

The AFPDS to IPDS transform times are close for segmentation write to IFS and segmentation e-mail, because they are processed in similar ways. The AFPDS to IPDS transform times are much longer for intelligent routing compared to segmentation write to IFS and segmentation e-mail.

Comparison of IPDS to PDF transforming time and printing rate

IPDS to PDF transforming times and printing rates for segmentation write to IFS, segmentation e-mail, and intelligent routing are compared in Table G-18. For detailed descriptions about the differences between segmentation and intelligent routing cases, see “Performance cases” on page 147.

Table G-18 IPDS to PDF transforming time and printing rate

PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.						
Number of pages	**Segmentation, write to IFS**		**Segmentation, e-mail**		**Intelligent routing**	
	Time	**Rate**	**Time**	**Rate**	**Time**	**Rate**
1	0.299	201	0.305	197	0.305	197
100	13.574	442	13.418	447	13.365	449
1,000	134.218	447	137.392	437	136.214	441
2,000	267.770	448	272.731	440	273.014	440

As shown by the results in Table G-18, IPDS to PDF transform times are proportional to the number of pages in the case. The results are similar for segmentation and intelligent routing.

Comparison of PDJ time and total job time

For segmentation write to IFS, all segments are written to IFS. For segmentation e-mail, all segments are e-mailed. For intelligent routing, each segment is either written to IFS, e-mailed, or both.

Table G-19 Distribution time and total job time

PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.						
Number of pages	**Segmentation, write to IFS**		**Segmentation, e-mail**		**Intelligent routing**	
	PDJ time (sec)	**Total time (min:sec)**	**PDJ time (sec)**	**Total time (min:sec)**	**PDJ time (sec)**	**Total time (min:sec)**
1	1	00:01	1	00:01	1	00:01
100	35	00:35	40	00:40	45	00:45
1,000	445	07:25	480	08:00	495	08:15
2,000	984	16:25	1,189	19:49	1,117	18:39

As shown in Table G-19, segmentation e-mail always has longer PDJ times and total job times compared to segmentation write to IFS because segmentation e-mail takes longer to process PDF segments. It takes longer to send e-mails with attachment than to write to IFS.

Intelligent routing has longer PDJ times and total job times compared to segmentation for smaller jobs. However, for the 2000 page case, intelligent routing has shorter PDJ times and total job times than segmentation e-mail because there are significant lower numbers of segments that need to be e-mailed in intelligent routing than segmentation e-mail.

Comparison of CPU time

Table G-20 shows the CPU times for WTR, PDJ, BCH, and Total.

Table G-20 CPU time

PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
Number of pages	**Segmentation, write to IFS**				**Segmentation, e-mail**				**Intelligent routing**			
	WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
1	<0.1	0.1	0.1	<0.3	<0.1	<0.1	0.1	<0.3	<0.1	0.1	0.2	<0.4
100	0.1	3.4	9.9	13.4	0.1	3.5	9.9	13.5	0.5	4.0	9.8	14.3
1,000	0.6	85.2	98.5	184.3	0.7	83.4	97.7	181.8	4.8	70.3	97.6	172.7
2,000	1.3	325.5	196.7	523.5	1.2	340.4	196.8	538.4	9.6	238.0	196.0	443.6

The CPU times of WTR are the same for segmentation write to IFS and segmentation e-mail. The AFPDS to IPDS transform times for those two are also close to each other. Since intelligent routing has the longest AFPDS to IPDS transform times, the CPU times of WTR for intelligent routing are the highest.

While the IPDS to PDF transform times are the same for segmentation write to IFS, segmentation e-mail, and intelligent routing, their CPU times of BCH are also the same.

For longer cases, intelligent routing has the lowest CPU times of PDJ. Segmentation write to IFS and e-mail have similar CPU times of PDJ.

Comparison of CPU utilization

%CPU utilizations were calculated for segmentation and intelligent routing based on measured CPU use, transforming times, PDJ times, and total job times shown in Table G-21.

Table G-21 CPU utilization

CPU Utilization. PWC=YES, Spool (QSPL) Storage=100M, Font Inline=NO.												
Number of pages	**Segmentation, write to IFS**				**Segmentation, e-mail**				**Intelligent routing**			
	WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
1	*	10.0	33.4	*	*	*	32.8	*	*	10.0	65.6	*
100	34.5	10.0	72.9	38.3	31.1	9.0	73.8	33.8	36.4	9.0	73.3	31.8
1,000	56.9	19.0	73.4	41.4	61.3	17.0	71.1	37.9	40.7	14.0	71.7	34.9
2,000	23.7	33.0	73.5	53.1	31.4	29.0	72.2	45.3	27.5	21.0	71.8	39.6
* Small jobs are too difficult to measure and evaluate accurately.												

%CPU utilizations for WTR are produced by dividing the CPU times for WTR by the AFPDS to IPDS transform times. %CPU utilizations for BCH are produced by dividing the CPU time for BCH by the IPDS to PDF transform time. %CPU utilizations for PDJ are produced by dividing the CPU time for PDJ by the sum of sending data to printer times and processing segments times. %CPU utilizations for Total are produced by dividing the total CPU times for WTR, BCH, and PDJ by the total job time.

%CPU utilizations for PDJ are very low. %CPU utilizations for BCH are very high because IPDS to PDF transform is a CPU extensive process. Segmentation and intelligent routing have similar %CPU utilizations for BCH because they have similar CPU times of BCH and similar IPDS to PDF transform times.

Spool (QSPL) Storage = 100 M compared to Spool (QSPL) Storage = 10.5 M

Table G-22 and Table G-23 compare intelligent routing results for different spool storage sizes.

Table G-22 Storage comparisons results (Part 1 of 2)

Spool Storage = 100 M versus Spool Storage = 10.5 M. PWC=YES, Font Inline=NO.								
Case name	**Spool size**	**AFPDS to IPDS time (sec)**	**IPDS to PDF time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (hh:mm:ss)**	**Output data rate (bytes/sec)**
SSSMAP1000	100	11.784	136.214	5092	441	495	00:08:15	72330
SSSMAP1000	10.5	15.991	134.641	3752	446	3748	01:02:30	9548
SSSMAP2000	100	34.965	273.014	3432	440	1117	00:18:39	60224
SSSMAP2000	10.5	29.096	272.447	4124	441	11502	03:11:44	6224

Table G-23 Storage comparison results (Part 2 of 2)

Spool storage = 100 M versus spool storage = 10.5 M. PWC=YES, Font Inline=NO.									
Case name	**Spool size**	**CPU time (sec)**				**%CPU**			
		WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
SSSMAP1000	100	4.8	70.3	97.6	172.7	40.7	14.0	71.7	34.9
SSSMAP1000	10.5	4.9	126.5	98.2	229.6	30.6	3.0	72.9	46.4
SSSMAP2000	100	9.6	238.0	196.0	443.6	27.5	21.0	71.8	39.6
SSSMAP2000	10.5	9.7	399.6	196.4	605.7	33.3	3.0	72.1	54.1

The impact of spool storage size on segmentation is insignificant compared to intelligent routing. Table G-22 and Table G-23 show the impact that the spool storage size has on the results for intelligent routing cases.

AFPDS to IPDS transform times and IPDS to PDF transform times are similar when different spool storage sizes are used.

For the SSSMAP1000 case, the PDJ time and total time for spool size = 10.5 M are eight times longer compared to 100M. For the SSSMAP2000 case, the PDJ time and total time for spool size = 10.5 M are 10 times longer compared to 100M.

The CPU times for WTR and BCH are similar when different spool storage sizes are used. However, CPU times for PDJ have increased by almost two times.

Font Inline = NO compared to Font Inline = YES

Font Inline = Yes is the option to embed fonts in a PDF. It ensures fidelity, but it produces PDF files with larger sizes and has more impact on performance. Font Inline = No produces PDF files with smaller sizes and give better performance.

The detailed information for each segment is different for Font Inline = No and Font Inline = Yes, as shown in Table G-24. Since the PDF size for each segment is different, the total PDF sizes for the cases are different.

Table G-24 Effect of font inclusion case descriptions: Unsegmented

Case name	Pages printed	Font inline=NO Total PDF size (bytes)	Font inline=YES Total PDF size (bytes)
IFS1	1	49,962	163,191
IFS100	100	3,080,700	9,873,610
IFS1000	1,000	30,807,000	98,736,100
IFS2000	2,000	61,614,000	197,472,200

Table G-25 lists the detailed information for each segment.

Table G-25 Effect of font inclusion case descriptions: Segmented

Segment	Segment name	Pages	Convert to PDF and write to IFS	Font Inline=NO PDF size (bytes)	Font Inline=YES PDF size (bytes)
1	BestWay	1	Yes	49,962	163,191
2	CleanTruck	2	Yes	54,553	167,756
3	Stork	2	Yes	51,323	164,532
4	Fed Ex	2	Yes	51,268	164,479
5	UPS	2	Yes	52,023	165,234
6	Airborne	1	Yes	48,941	162,169

Table G-26 and Table G-27 compare segmentation write to IFS results for Font Inline=No and Font Inline=Yes.

Table G-26 Effect of font inclusion on performance (Part 1 of 2)

Font Inline=NO versus Font Inline=YES. PWC=YES, Spool Storage = 100M.								
Case name	**Font inline**	**AFPDS to IPDS time (sec)**	**IPDS to PDF time (sec)**	**AFPDS to IPDS (rate in PPM)**	**IPDS to PDF (rate in PPM)**	**PDJ time (sec)**	**Total time (mm:ss)**	**Output data rate (bytes/sec)**
IFS1000	No	1.055	134.218	56872	447	445	07:25	69229
IFS1000	Yes	1.109	141.715	54103	423	460	07:25	214644
IFS2000	No	5.490	267.770	21858	448	984	16:25	62552
IFS2000	Yes	7.754	284.269	15476	422	1032	17:14	191349

Table G-27 Effect of font inclusion on performance (Part 2 of 2)

Font Inline=NO versus Font Inline=YES. PWC=YES, Spool Storage = 100M.									
Case name	**Font inline**	**CPU Time (sec)**				**%CPU Utilization**			
		WTR	**PDJ**	**BCH**	**Total**	**WTR**	**PDJ**	**BCH**	**Total**
IFS1000	No	0.6	85.2	98.5	184.3	56.9	19.0	73.4	41.4
IFS1000	Yes	0.7	85.6	102.2	188.5	63.1	19.0	72.1	424.0
IFS2000	No	1.3	325.5	196.7	523.5	23.7	33.0	73.5	53.1
IFS2000	Yes	1.3	328.0	203.6	532.9	16.8	32.0	71.6	54.1

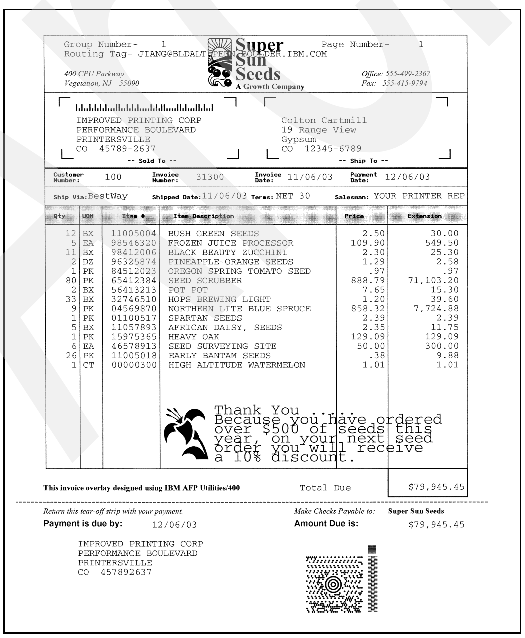
The total PDF output sizes for Font Inline = Yes are 3.2 times as large as Font Inline = No. As shown by the results in Table G-26 and Table G-27, when Font Inline = Yes, AFPDS to IPDS transform times, IPDS to PDF transform times, PDJ times, Total times, CPU times for PDJ and BCH are slightly longer than when Font Inline = No.

Recommendations

We recommend Spool (QSPL) Storage = 100 MB for the users. 10.5 MB of spool (QSPL) storage may be too small especially for intelligent routing. Smaller image sizes give better performance. We recommend that you minimize image sizes where possible. Only use color image when required.

Be aware of the performance differences in segmentation IFS, segmentation e-mail, and intelligent routing. Use Font Inline = Yes only when required. With configurations and cases similar to those shown in this appendix, the iSeries system gives good performance overall.

Sample output



Group Number- 1
Routing Tag- JIANG@BLDALTPENN.BOULDER.IBM.COM
Page Number- 1

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 --

Colton Cartmill
19 Range View
Gypsum
CO 12345-6789
-- Ship To --

Customer Number: 100 Invoice Number: 31300 Invoice Date: 11/06/03 Payment Date: 12/06/03

Ship Via: BestWay Shipped Date: 11/06/03 Terms: NET 30 Salesman: YOUR PRINTER REP

Qty	UOM	Item #	Item Description	Price	Extension
12	BX	11005004	BUSH GREEN SEEDS	2.50	30.00
5	EA	98546320	FROZEN JUICE PROCESSOR	109.90	549.50
11	BX	98412006	BLACK BEAUTY ZUCCHINI	2.30	25.30
2	DZ	96325874	PINEAPPLE-ORANGE SEEDS	1.29	2.58
1	PK	84512023	OREGON SPRING TOMATO SEED	.97	.97
80	PK	65412384	SEED SCRUBBER	888.79	71,103.20
2	BX	56413213	POT POT	7.65	15.30
33	BX	32746510	HOPS BREWING LIGHT	1.20	39.60
9	PK	04569870	NORTHERN LITE BLUE SPRUCE	858.32	7,724.88
1	PK	01100517	SPARTAN SEEDS	2.39	2.39
5	BX	11057893	AFRICAN DAISY, SEEDS	2.35	11.75
1	PK	15975365	HEAVY OAK	129.09	129.09
6	EA	46578913	SEED SURVEYING SITE	50.00	300.00
26	PK	11005018	EARLY BANTAM SEEDS	.38	9.88
1	CT	00000300	HIGH ALTITUDE WATERMELON	1.01	1.01

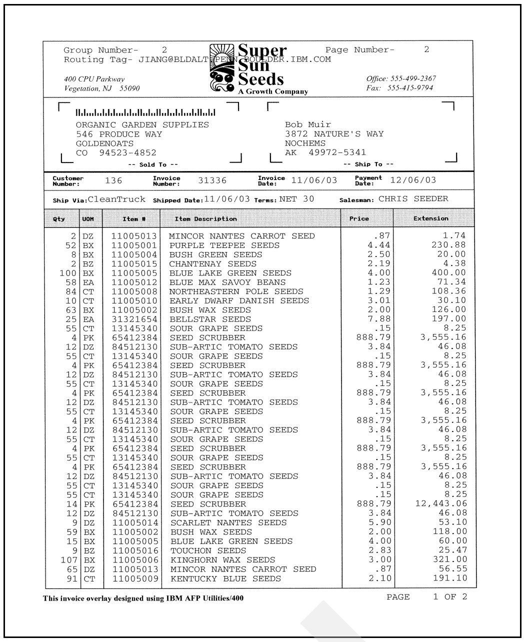
Thank You ....
Because you have ordered
over $500 of seeds this
year, on your next seed
order you will receive
a 10% discount.

This invoice overlay designed using IBM AFP Utilities/400
Total Due $79,945.45

Return this tear-off strip with your payment.
Payment is due by: 12/06/03
Make Checks Payable to: Super Sun Seeds
Amount Due is: $79,945.45

IMPROVED PRINTING CORP
PERFORMANCE BOULEVARD
PRINTERSVILLE
CO 457892637

Figure G-1 Segment 1, BestWay, 1 page



Group Number- 2 Page Number- 2
Routing Tag- JIANG@BLDALT.PENN.BOULDER.IBM.COM

400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

ORGANIC GARDEN SUPPLIES
546 PRODUCE WAY
GOLDENOATS
CO 94523-4852
-- Sold To --

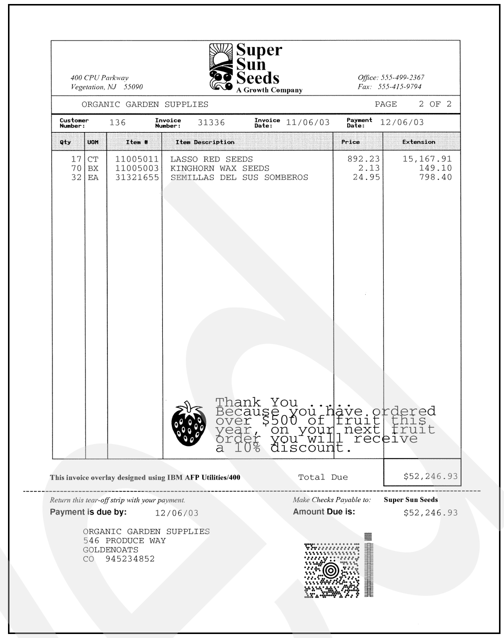
Bob Muir
3872 NATURE'S WAY
NOCHEMS
AK 49972-5341
-- Ship To --

Customer Number: 136 Invoice Number: 31336 Invoice Date: 11/06/03 Payment Date: 12/06/03

Ship Via: CleanTruck Shipped Date: 11/06/03 Terms: NET 30 Salesman: CHRIS SEEDER

Qty	UOM	Item #	Item Description	Price	Extension
2	DZ	11005013	MINCOR NANTES CARROT SEED	.87	1.74
52	BX	11005001	PURPLE TEEPEE SEEDS	4.44	230.88
8	BX	11005004	BUSH GREEN SEEDS	2.50	20.00
2	BZ	11005015	CHANTENAY SEEDS	2.19	4.38
100	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	400.00
58	EA	11005012	BLUE MAX SAVOY BEANS	1.23	71.34
84	CT	11005008	NORTHEASTERN POLE SEEDS	1.29	108.36
10	CT	11005010	EARLY DWARF DANISH SEEDS	3.01	30.10
63	BX	11005002	BUSH WAX SEEDS	2.00	126.00
25	EA	31321654	BELLSTAR SEEDS	7.88	197.00
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
14	PK	65412384	SEED SCRUBBER	888.79	12,443.06
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
9	DZ	11005014	SCARLET NANTES SEEDS	5.90	53.10
59	BX	11005002	BUSH WAX SEEDS	2.00	118.00
15	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	60.00
9	BZ	11005016	TOUCHON SEEDS	2.83	25.47
107	BX	11005006	KINGHORN WAX SEEDS	3.00	321.00
65	DZ	11005013	MINCOR NANTES CARROT SEED	.87	56.55
91	CT	11005009	KENTUCKY BLUE SEEDS	2.10	191.10

This invoice overlay designed using IBM AFP Utilities/400 PAGE 1 OF 2



400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

ORGANIC GARDEN SUPPLIES PAGE 2 OF 2

Customer Number: 136 Invoice Number: 31336 Invoice Date: 11/06/03 Payment Date: 12/06/03

Qty	UOM	Item #	Item Description	Price	Extension
17	CT	11005011	LASSO RED SEEDS	892.23	15,167.91
70	BX	11005003	KINGHORN WAX SEEDS	2.13	149.10
32	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	798.40

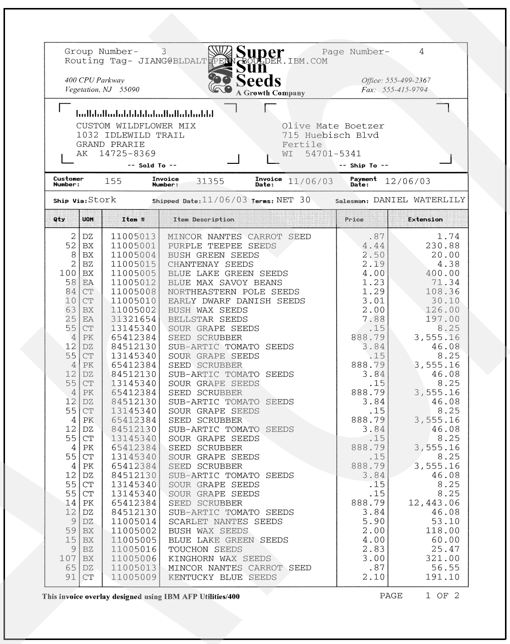
Thank You ...
Because you have ordered over $500 of fruit this year, on your next fruit order you will receive a 10% discount.

This invoice overlay designed using IBM AFP Utilities/400 Total Due $52,246.93

Return this tear-off strip with your payment. Make Checks Payable to: Super Sun Seeds
Payment is due by: 12/06/03 Amount Due is: $52,246.93

ORGANIC GARDEN SUPPLIES
546 PRODUCE WAY
GOLDENOATS
CO 945234852

Figure G-2 Segment 2, CleanTruck, 2 pages



Group Number- 3 Page Number- 4
Routing Tag- JIANG@BLDALT.PENN.BOULDER.IBM.COM

400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

CUSTOM WILDFLOWER MIX
1032 IDLEWILD TRAIL
GRAND PRARIE
AK 14725-8369
-- Sold To --

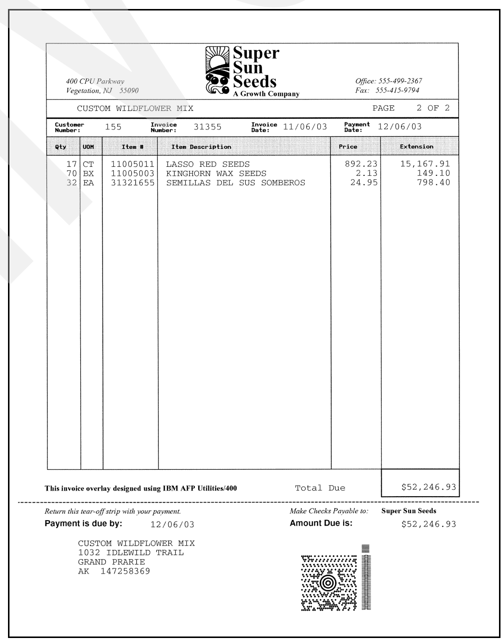
Olive Mate Boetzer
715 Huebisch Blvd
Fertile
WI 54701-5341
-- Ship To --

Customer Number: 155 Invoice Number: 31355 Invoice Date: 11/06/03 Payment Date: 12/06/03

Ship Via: Stork Shipped Date: 11/06/03 Terms: NET 30 Salesman: DANIEL WATERLILY

Qty	UOM	Item #	Item Description	Price	Extension
2	DZ	11005013	MINCOR NANTES CARROT SEED	.87	1.74
52	BX	11005001	PURPLE TEEPEE SEEDS	4.44	230.88
8	BX	11005004	BUSH GREEN SEEDS	2.50	20.00
2	BZ	11005015	CHANTENAY SEEDS	2.19	4.38
100	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	400.00
58	EA	11005012	BLUE MAX SAVOY BEANS	1.23	71.34
84	CT	11005008	NORTHEASTERN POLE SEEDS	1.29	108.36
10	CT	11005010	EARLY DWARF DANISH SEEDS	3.01	30.10
63	BX	11005002	BUSH WAX SEEDS	2.00	126.00
25	EA	31321654	BELLSTAR SEEDS	7.88	197.00
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
14	PK	65412384	SEED SCRUBBER	888.79	12,443.06
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
9	DZ	11005014	SCARLET NANTES SEEDS	5.90	53.10
59	BX	11005002	BUSH WAX SEEDS	2.00	118.00
15	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	60.00
9	BZ	11005016	TOUCHON SEEDS	2.83	25.47
107	BX	11005006	KINGHORN WAX SEEDS	3.00	321.00
65	DZ	11005013	MINCOR NANTES CARROT SEED	.87	56.55
91	CT	11005009	KENTUCKY BLUE SEEDS	2.10	191.10

This invoice overlay designed using IBM AFP Utilities/400 PAGE 1 OF 2



400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

CUSTOM WILDFLOWER MIX PAGE 2 OF 2

Customer Number: 155 Invoice Number: 31355 Invoice Date: 11/06/03 Payment Date: 12/06/03

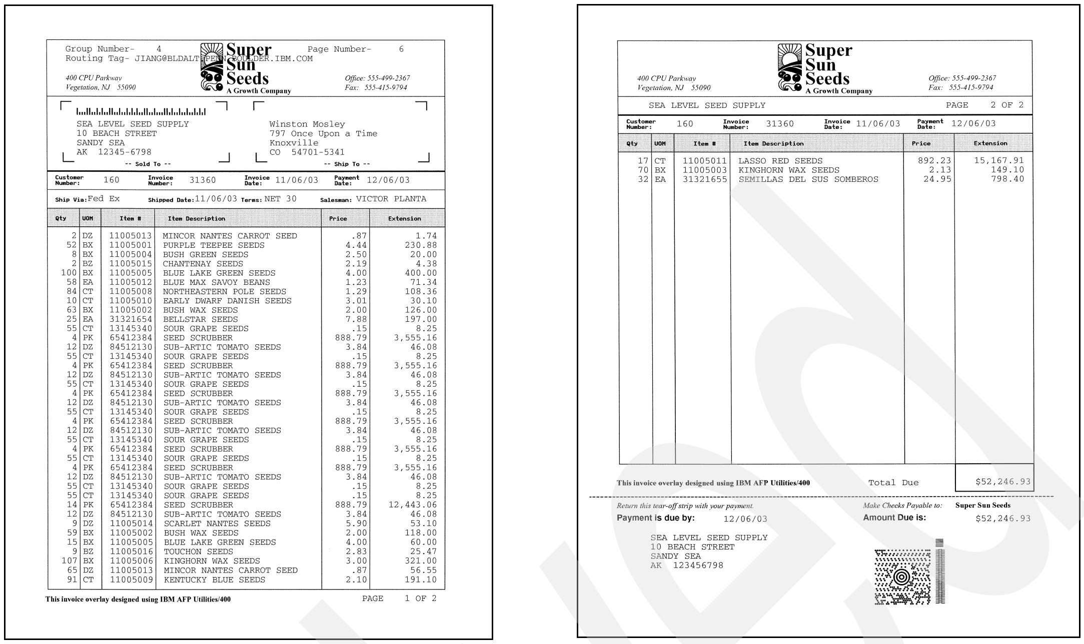
Qty	UOM	Item #	Item Description	Price	Extension
17	CT	11005011	LASSO RED SEEDS	892.23	15,167.91
70	BX	11005003	KINGHORN WAX SEEDS	2.13	149.10
32	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	798.40

This invoice overlay designed using IBM AFP Utilities/400 Total Due $52,246.93

Return this tear-off strip with your payment. Make Checks Payable to: Super Sun Seeds
Payment is due by: 12/06/03 Amount Due is: $52,246.93

CUSTOM WILDFLOWER MIX
1032 IDLEWILD TRAIL
GRAND PRARIE
AK 147258369

Figure G-3 Segment 3, Stork, 2 pages



Group Number- 4 | Routing Tag- JIANG@BLDALTPENN.BOULDER.IBM.COM | Page Number- 6

Super Sun Seeds — A Growth Company
400 CPU Parkway, Vegetation, NJ 55090 — Office: 555-499-2367, Fax: 555-415-9794

-- Sold To --
SEA LEVEL SEED SUPPLY
10 BEACH STREET
SANDY SEA
AK 12345-6798

-- Ship To --
Winston Mosley
797 Once Upon a Time
Knoxville
CO 54701-5341

Customer Number: 160 | Invoice Number: 31360 | Invoice Date: 11/06/03 | Payment Date: 12/06/03
Ship Via: Fed Ex | Shipped Date: 11/06/03 | Terms: NET 30 | Salesman: VICTOR PLANTA

Qty	UOM	Item #	Item Description	Price	Extension
2	DZ	11005013	MINCOR NANTES CARROT SEED	.87	1.74
52	BX	11005001	PURPLE TEEPEE SEEDS	4.44	230.88
8	BX	11005004	BUSH GREEN SEEDS	2.50	20.00
2	BZ	11005015	CHANTENAY SEEDS	2.19	4.38
100	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	400.00
58	EA	11005012	BLUE MAX SAVOY BEANS	1.23	71.34
84	CT	11005008	NORTHEASTERN POLE SEEDS	1.29	108.36
10	CT	11005010	EARLY DWARF DANISH SEEDS	3.01	30.10
63	BX	11005002	BUSH WAX SEEDS	2.00	126.00
25	EA	31321654	BELLSTAR SEEDS	7.88	197.00
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
4	PK	65412384	SEED SCRUBBER	888.79	3,555.16
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
55	CT	13145340	SOUR GRAPE SEEDS	.15	8.25
14	PK	65412384	SEED SCRUBBER	888.79	12,443.06
12	DZ	84512130	SUB-ARTIC TOMATO SEEDS	3.84	46.08
9	DZ	11005014	SCARLET NANTES SEEDS	5.90	53.10
59	BX	11005002	BUSH WAX SEEDS	2.00	118.00
15	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	60.00
9	BZ	11005016	TOUCHON SEEDS	2.83	25.47
107	BX	11005006	KINGHORN WAX SEEDS	3.00	321.00
65	DZ	11005013	MINCOR NANTES CARROT SEED	.87	56.55
91	CT	11005009	KENTUCKY BLUE SEEDS	2.10	191.10

This invoice overlay designed using IBM AFP Utilities/400 — PAGE 1 OF 2

Super Sun Seeds — A Growth Company
400 CPU Parkway, Vegetation, NJ 55090 — Office: 555-499-2367, Fax: 555-415-9794

SEA LEVEL SEED SUPPLY — PAGE 2 OF 2

Customer Number: 160 | Invoice Number: 31360 | Invoice Date: 11/06/03 | Payment Date: 12/06/03

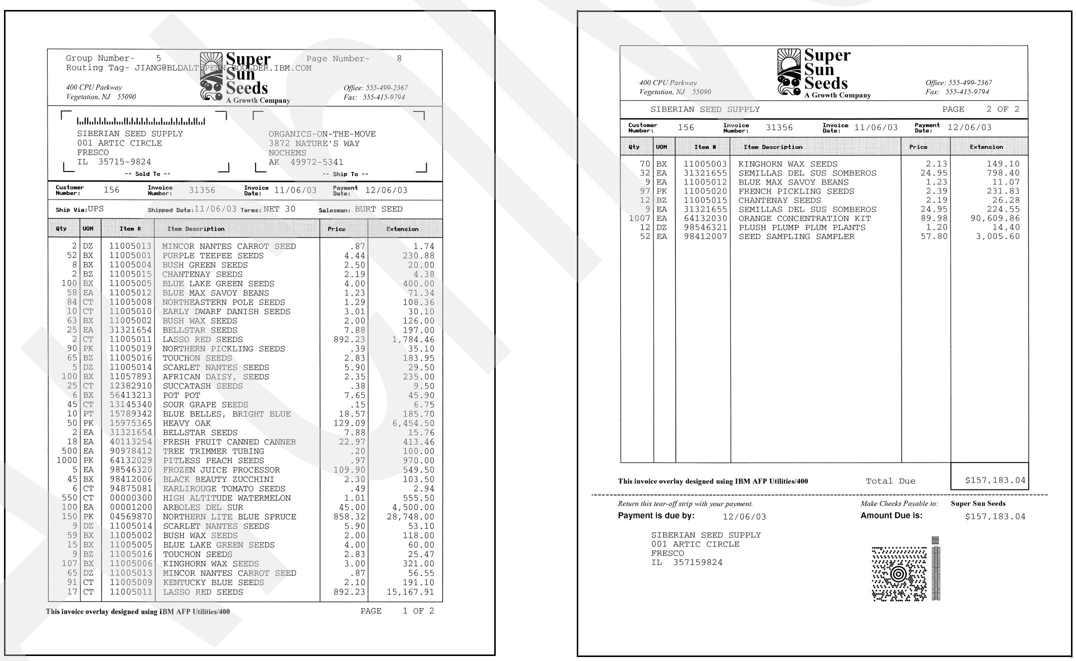
Qty	UOM	Item #	Item Description	Price	Extension
17	CT	11005011	LASSO RED SEEDS	892.23	15,167.91
70	BX	11005003	KINGHORN WAX SEEDS	2.13	149.10
32	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	798.40

This invoice overlay designed using IBM AFP Utilities/400 — Total Due $52,246.93

Return this tear-off strip with your payment. — Make Checks Payable to: Super Sun Seeds
Payment is due by: 12/06/03 — Amount Due is: $52,246.93

SEA LEVEL SEED SUPPLY
10 BEACH STREET
SANDY SEA
AK 123456798

Figure G-4 Segment 4, Fed Ex, 2 pages



Group Number- 5 | Routing Tag- JIANG@BLDALTPENN.BOULDER.IBM.COM | Page Number- 8

Super Sun Seeds — A Growth Company
400 CPU Parkway, Vegetation, NJ 55090 — Office: 555-499-2367, Fax: 555-415-9794

-- Sold To --
SIBERIAN SEED SUPPLY
001 ARTIC CIRCLE
FRESCO
IL 35715-9824

-- Ship To --
ORGANICS-ON-THE-MOVE
3872 NATURE'S WAY
NOCHEMS
AK 49972-5341

Customer Number: 156 | Invoice Number: 31356 | Invoice Date: 11/06/03 | Payment Date: 12/06/03
Ship Via: UPS | Shipped Date: 11/06/03 | Terms: NET 30 | Salesman: BURT SEED

Qty	UOM	Item #	Item Description	Price	Extension
2	DZ	11005013	MINCOR NANTES CARROT SEED	.87	1.74
52	BX	11005001	PURPLE TEEPEE SEEDS	4.44	230.88
8	BX	11005004	BUSH GREEN SEEDS	2.50	20.00
2	BZ	11005015	CHANTENAY SEEDS	2.19	4.38
100	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	400.00
58	EA	11005012	BLUE MAX SAVOY BEANS	1.23	71.34
84	CT	11005008	NORTHEASTERN POLE SEEDS	1.29	108.36
10	CT	11005010	EARLY DWARF DANISH SEEDS	3.01	30.10
63	BX	11005002	BUSH WAX SEEDS	2.00	126.00
25	EA	31321654	BELLSTAR SEEDS	7.88	197.00
2	CT	11005011	LASSO RED SEEDS	892.23	1,784.46
90	PK	11005019	NORTHERN PICKLING SEEDS	.39	35.10
65	BZ	11005016	TOUCHON SEEDS	2.83	183.95
5	DZ	11005014	SCARLET NANTES SEEDS	5.90	29.50
100	BX	11057893	AFRICAN DAISY, SEEDS	2.35	235.00
25	CT	12382910	SUCCATASH SEEDS	.38	9.50
6	BX	56413213	POT POT	7.65	45.90
45	CT	13145340	SOUR GRAPE SEEDS	.15	6.75
10	PT	15789342	BLUE BELLES, BRIGHT BLUE	18.57	185.70
50	PK	15975365	HEAVY OAK	129.09	6,454.50
2	EA	31321654	BELLSTAR SEEDS	7.88	15.76
18	EA	40113254	FRESH FRUIT CANNED CANNER	22.97	413.46
500	EA	90978412	TREE TRIMMER TUBING	.20	100.00
1000	PK	64132029	PITLESS PEACH SEEDS	.97	970.00
5	EA	98546320	FROZEN JUICE PROCESSOR	109.90	549.50
45	BX	98412006	BLACK BEAUTY ZUCCHINI	2.30	103.50
6	CT	94875081	EARLIROUGE TOMATO SEEDS	.49	2.94
550	CT	00000300	HIGH ALTITUDE WATERMELON	1.01	555.50
100	EA	00001200	ARBOLES DEL SUR	45.00	4,500.00
150	PK	04569870	NORTHERN LITE BLUE SPRUCE	858.32	28,748.00
9	DZ	11005014	SCARLET NANTES SEEDS	5.90	53.10
59	BX	11005002	BUSH WAX SEEDS	2.00	118.00
15	BX	11005005	BLUE LAKE GREEN SEEDS	4.00	60.00
9	BZ	11005016	TOUCHON SEEDS	2.83	25.47
107	BX	11005006	KINGHORN WAX SEEDS	3.00	321.00
65	DZ	11005013	MINCOR NANTES CARROT SEED	.87	56.55
91	CT	11005009	KENTUCKY BLUE SEEDS	2.10	191.10
17	CT	11005011	LASSO RED SEEDS	892.23	15,167.91

This invoice overlay designed using IBM AFP Utilities/400 — PAGE 1 OF 2

Super Sun Seeds — A Growth Company
400 CPU Parkway, Vegetation, NJ 55090 — Office: 555-499-2367, Fax: 555-415-9794

SIBERIAN SEED SUPPLY — PAGE 2 OF 2

Customer Number: 156 | Invoice Number: 31356 | Invoice Date: 11/06/03 | Payment Date: 12/06/03

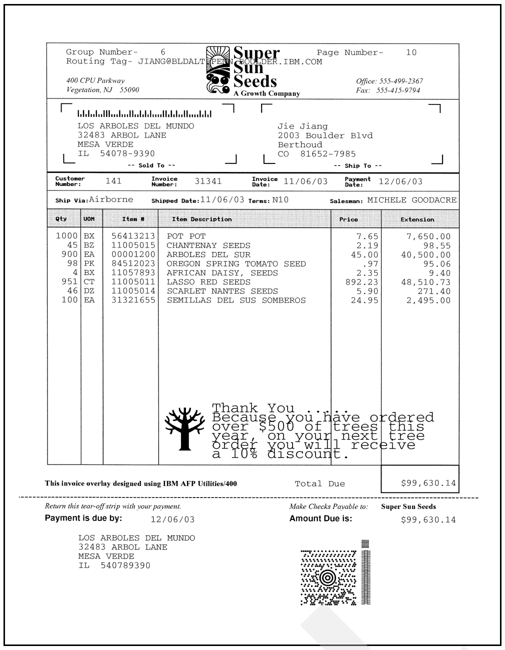
Qty	UOM	Item #	Item Description	Price	Extension
70	BX	11005003	KINGHORN WAX SEEDS	2.13	149.10
32	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	798.40
9	EA	11005012	BLUE MAX SAVOY BEANS	1.23	11.07
97	PK	11005020	FRENCH PICKLING SEEDS	2.39	231.83
12	BZ	11005015	CHANTENAY SEEDS	2.19	26.28
9	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	224.55
1007	EA	64132030	ORANGE CONCENTRATION KIT	89.98	90,609.86
12	DZ	98546321	PLUSH PLUMP PLUM PLANTS	1.20	14.40
52	EA	98412007	SEED SAMPLING SAMPLER	57.80	3,005.60

This invoice overlay designed using IBM AFP Utilities/400 — Total Due $157,183.04

Return this tear-off strip with your payment. — Make Checks Payable to: Super Sun Seeds
Payment is due by: 12/06/03 — Amount Due is: $157,183.04

SIBERIAN SEED SUPPLY
001 ARTIC CIRCLE
FRESCO
IL 357159824

Figure G-5 Segment 5, UPS, 2 pages



Group Number- 6 Page Number- 10
Routing Tag- JIANG@BLDALT.PENN.BOULDER.IBM.COM

Super Sun Seeds
A Growth Company

400 CPU Parkway
Vegetation, NJ 55090

Office: 555-499-2367
Fax: 555-415-9794

LOS ARBOLES DEL MUNDO
32483 ARBOL LANE
MESA VERDE
IL 54078-9390
-- Sold To --

Jie Jiang
2003 Boulder Blvd
Berthoud
CO 81652-7985
-- Ship To --

Customer Number: 141 **Invoice Number:** 31341 **Invoice Date:** 11/06/03 **Payment Date:** 12/06/03

Ship Via: Airborne **Shipped Date:** 11/06/03 **Terms:** N10 **Salesman:** MICHELE GOODACRE

Qty	UOM	Item #	Item Description	Price	Extension
1000	BX	56413213	POT POT	7.65	7,650.00
45	BZ	11005015	CHANTENAY SEEDS	2.19	98.55
900	EA	00001200	ARBOLES DEL SUR	45.00	40,500.00
98	PK	84512023	OREGON SPRING TOMATO SEED	.97	95.06
4	BX	11057893	AFRICAN DAISY, SEEDS	2.35	9.40
951	CT	11005011	LASSO RED SEEDS	892.23	48,510.73
46	DZ	11005014	SCARLET NANTES SEEDS	5.90	271.40
100	EA	31321655	SEMILLAS DEL SUS SOMBEROS	24.95	2,495.00

Thank You .....
Because you have ordered over $500 of trees this year, on your next tree order you will receive a 10% discount.

This invoice overlay designed using IBM AFP Utilities/400

Total Due $99,630.14

Return this tear-off strip with your payment.
Payment is due by: 12/06/03

Make Checks Payable to: **Super Sun Seeds**
Amount Due is: $99,630.14

LOS ARBOLES DEL MUNDO
32483 ARBOL LANE
MESA VERDE
IL 540789390

Figure G-6 Segment 6, Airborne, 1 page

Figure G-7 IOCA FS45 4 square-inch image

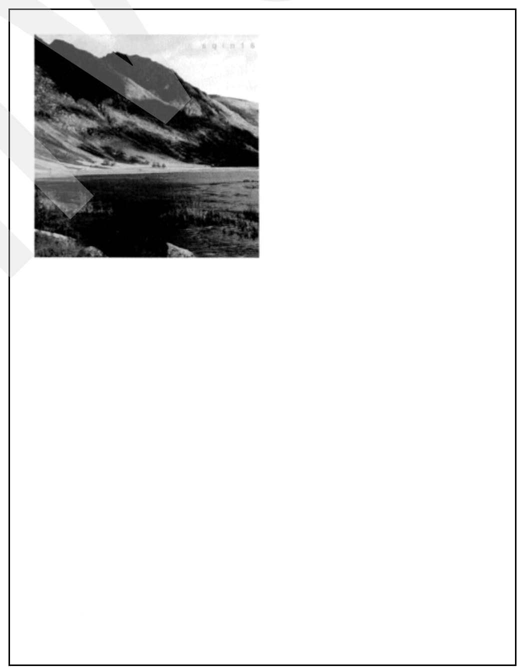


Figure G-8 IOCA FS45 16 square-inch image

H

Sample programs

This appendix contains the sample programs that are described in the chapters of this Redpaper. Softcopy versions of the sample source code and spreadsheets (in both Lotus 123 and Microsoft Excel format) are available on the IBM Redbooks Web site at:

ftp://www.redbooks.ibm.com/redbooks/REDP3752

See Appendix I, “Additional material” on page 193, for information about how to use the Web material.

Important: The following code samples are provided *as is* and are not supported in any way by IBM. Assistance with debugging or modifying the program to fit your needs requires a consulting agreement.

Sample programming objects used for intelligent routing

This section contains the programs described in Chapter 6, “Using the PDF mapping program” on page 63.

Sample PDF mapping program in RPGLE

The sample program shown in Example H-1 is used to route spooled file segments according to information obtained from a lookup file. For a complete description of this program, see Chapter 6, “Using the PDF mapping program” on page 63.

Example: H-1 Mapping program for intelligent routing

```
 1  FLOOKUP    IF   E           K DISK    EXTFILE('QGPL/LOOKUP')
 2  D*
 3  D INPUTDS         DS
 4  D  JOBNAM                 1     26
 5  D  SPLFID                27     36
 6  D  SPLNO                 37     40B 0
 7  D  RTGTAG                41    290
 8  D  PDFFILE              291    630
 9  D  SVRTYPE              631    631
10  D  RES1                 632    632
11  D  PATHCCSID            633    636B 0
12  D  SENDER               637    646
13  D  USRDTA               647    656
14  D  SYSNAME              657    664
15  D  TIMESTMP             665    672
16  D  OUTQ                 673    682
17  D  OUTQLIB              683    692
18  D  RES5                 693    712
19  D  FORMTYPE             713    722
20  D*********************************************
21  D OUTDS           DS
22  D  DISPOSTN                1
23  D  CALLAGIN                1
24  D  RES2                    2
25  D  MSGLEN                  9B 0
26  D  ADDRLEN                 9B 0
27  D  MSGTEXT               255
28  D  RES3                    1
29  D  EXTOFF                  9B 0
30  D  CCSID                   9B 0
31  D  DSPPDFSTMF              1
32  D  DSPPDFSPLF              1
33  D  DSPPDFERR               1
34  D  DSPAFPSPLF              1
35  D  RES4                    7
36  D  ADDRESS               255
37  D*
38  D  EXTLEN                  9B 0
39  D  SUBOFF                  9B 0
40  D  SUBLEN                  9B 0
41  D  RPLYOFF                 9B 0
42  D  RPLYLEN                 9B 0
43  D  CCOFF                   9B 0
44  D  CCLEN                   9B 0
```

```
 45   D  BCCOFF                        9B 0
 46   D  BCCLEN                        9B 0
 47   D  BDYPTHOFF                     9B 0
 48   D  DIRPTHOFF                     9B 0
 49   D  DIRPTHLEN                     9B 0
 50   D  ATTPTHOFF                     9B 0
 51   D  STMFNAMOFF                    9B 0
 52   D  STMFNAMLEN                    9B 0
 53   D  EATTNAMOFF                    9B 0
 54   D  EATTNAMLEN                    9B 0
 55   D  PUBAUTOFF                     9B 0
 56   D  PUBAUTLEN                     9B 0
 57   D  PDFSPLOFF                     9B 0
 58   D  PDFSPLLEN                     9B 0
 59   D  AFPSPLOFF                     9B 0
 60   D  AFPSPLLEN                     9B 0
 61   D  ENCRPTOFF                     9B 0
 62   D  ENCRPTLEN                     9B 0
 63   D*
 64   D  EXTSUBJ                      80
 65   D  EXTRPLY                      80
 66   D  EXTCC                        80
 67   D  EXTBCC                       80
 68   D*
 69   D  BDYLEN                        9B 0
 70   D  BDYNUM                        9B 0
 71   D  BDYX1                         9B 0
 72   D  BDYY1                         9B 0
 73   D  BDYOFF1                       9B 0
 74   D  BDYPLEN1                      9B 0
 75   D  BDYUSE1                       1
 76   D  BDYRES1                       3
 77   D  BDYPTH1                      80
 78   D  BDYX2                         9B 0
 79   D  BDYY2                         9B 0
 80   D  BDYOFF2                       9B 0
 81   D  BDYPLEN2                      9B 0
 82   D  BDYUSE2                       1
 83   D  BDYRES2                       3
 84   D  BDYPTH2                      80
 85   D  BDYX3                         9B 0
 86   D  BDYY3                         9B 0
 87   D  BDYOFF3                       9B 0
 88   D  BDYPLEN3                      9B 0
 89   D  BDYUSE3                       1
 90   D  BDYRES3                       3
 91   D  BDYPTH3                      80
 92   D*
 93   D  ATTLEN                        9B 0
 94   D  ATTNUMB                       9B 0
 95   D  ATTX1                         9B 0
 96   D  ATTY1                         9B 0
 97   D  ATTOFF1                       9B 0
 98   D  ATTPLEN1                      9B 0
 99   D  ATTUSE1                       1
100   D  ATTRES1                       3
101   D  ATTPTH1                      80
102   D  ATTX2                         9B 0
103   D  ATTY2                         9B 0
104   D  ATTOFF2                       9B 0
```

```
105   D  ATTPLEN2                    9B 0
106   D  ATTUSE2                     1
107   D  ATTRES2                     3
108   D  ATTPTH2                    80
109   D  ATTX3                       9B 0
110   D  ATTY3                       9B 0
111   D  ATTOFF3                     9B 0
112   D  ATTPLEN3                    9B 0
113   D  ATTUSE3                     1
114   D  ATTRES3                     3
115   D  ATTPTH3                    80
116   D*
117   D  DIRPTH                     80
118   D*
119   D  PDFSTMFNAM                 80
120   D  PDFATTNAM                  80
121   D  PDFPUBAUT                  10
122   D*
123   D  PDFOUTQ                    10
124   D  PDFOUTQLIB                 10
125   D  PDFSPLNAM                  10
126   D  PDFUSRDTA                  10
127   D  PDFUSRDFN                 255
128   D  PDFFORM                    10
129   D*
130   D  AFPOUTQ                    10
131   D  AFPOUTQLIB                 10
132   D  AFPSPLNAM                  10
133   D  AFPUSRDTA                  10
134   D  AFPUSRDFN                 255
135   D  AFPFORM                    10
136   D*
137   D  PDFMASTPW                  32
138   D  PDFUSRPW                   32
139   D  PDFPRT                      1
140   D  PDFDOCCHG                   1
141   D  PDFCOPY                     1
142   D  PDFENCLVL                   1
143   D  PDFCNTACC                   1
144   D  PDFCHGCMT                   1
145   D  PDFDOCASB                   1
146   D*
147   D INPUTLEN        S            9B 0
148   D OUTPUTLEN       S            9B 0
149   D OUTINFO         S            9B 0
150   D*********************************************
151   C     *ENTRY       PLIST
152   C                  PARM                    INPUTDS
153   C                  PARM                    INPUTLEN
154   C                  PARM                    OUTDS
155   C                  PARM                    OUTPUTLEN
156   C                  PARM                    OUTINFO
157   C
158   C* Check to see if the output buffer is large enough.  OUTPUTLEN
159   C* contains the initial length of the output buffer.  If it is not
160   C* large enough, return with OUTINFO set to be the length required.
161   C* This program will be called right back with OUTPUTLEN = OUTINFO.
162   C*
163   C                  EVAL      OUTINFO   = 2509
164   C     OUTINFO      IFGT      OUTPUTLEN
```

```
165   C                   RETURN
166   C                   ENDIF
167   C*
168   C* Set data structure to all '00's, to make sure there is no stray data in
169   C* any unused fields.
170   C*
171   C                   MOVE      *ALLx'00'     OUTDS
172   C*
173   C* Set more processing (CALLAGIN), reserved fields, and pointer to
174   C* extension area.
175   C                   EVAL      CALLAGIN = '0'
176   C                   EVAL      RES2     = X'0000'
177   C                   EVAL      RES3     = X'00'
178   C                   EVAL      RES4     = X'00000000000000'
179   C                   EVAL      EXTOFF   = 542
180   C                   EVAL      EXTLEN   = 100
181   C*
182   C* Lookup customer preferences record
183   C                   SUBST     RTGTAG        CUSTNO          6
184   C     CUSTNO        CHAIN     LOOKUP
185   C
186   C* If customer record not found, tell PSF to send a message to PDFADMIN
187   C* by turning on the DSPPDFERR code.
188   C                   IF        NOT %FOUND
189   C                   EVAL      DSPPDFSTMF = '0'
190   C                   EVAL      DSPPDFSPLF = '0'
191   C                   EVAL      DSPPDFERR  = '1'
192   C                   EVAL      DSPAFPSPLF = '0'
193   C                   EVAL      EXTOFF   = 0
194   C                   RETURN
195   C                   ENDIF
196   C*
197   C* Error condition does not exist if found customer record.
198   C                   EVAL      DSPPDFERR  = '0'
199   C*
200   C* Customer found - evaluate different distributions.
201   C*
202   C* Check customer preference for e-mail
203   C     DISPEMAIL     IFEQ      'Y'
204   C*   Mail the file
205   C*     Base fields for e-mail
206   C                   EVAL      DISPOSTN = '1'
207   C                   EVAL      MSGLEN    = 255
208   C                   EVAL      ADDRLEN   = 255
209   C                   EVAL      MSGTEXT = 'Hello '
210   C                             + %TRIMR(PERSON) +', this is your invoice '
211   C                             + 'for ' + %TRIMR(COMP)+'.'
212   C                   EVAL      CCSID    = 0
213   C                   EVAL      ADDRESS = EADDR
214   C*     Extension Area for e-mail
215   C*       Subject
216   C                   EVAL      SUBOFF   = 642
217   C                   EVAL      SUBLEN   = 80
218   C                   EVAL      EXTSUBJ  = 'Using Intelligent Routing'
219   C*       Reply-to address for e-mail
220   C                   EVAL      RPLYOFF  = 722
221   C                   EVAL      RPLYLEN  = 80
222   C                   EVAL      EXTRPLY  = '''fred@supersunseeds.com'''
223   C*       CC address for e-mail
224   C                   EVAL      CCOFF    = 802
```

```
225   C                   EVAL      CCLEN    = 80
226   C                   EVAL      EXTCC    = '''rep1@supersunseeds.com'''
227   C                                         +'''mgr1@supersunseeds.com'''
228   C*       BCC address for e-mail
229   C                   EVAL      BCCOFF   = 882
230   C                   EVAL      BCCLEN   = 80
231   C                   EVAL      EXTBCC   = '''accounting@supersunseeds.com'''
232   C*       Rename PDF attachment
233   C                   EVAL      EATTNAMOFF = 1738
234   C                   EVAL      PDFATTNAM = 'Invoice #'
235   C                             + %subst(rtgtag:7:5) + '.pdf'
236   C                   EVAL      EATTNAMLEN = 80
237   C*
238   C*     Encryption of PDF file for e-mail
239   C                   EVAL      ENCRPTOFF  = 2438
240   C                   EVAL      ENCRPTLEN  = 71
241   C*
242   C                   EVAL      PDFMASTPW = 'master'
243   C                   EVAL      PDFUSRPW  = ''
244   C                   EVAL      PDFPRT    = '1'
245   C                   EVAL      PDFDOCCHG = '0'
246   C                   EVAL      PDFCOPY   = '0'
247   C                   EVAL      PDFENCLVL = '2'
248   C                   EVAL      PDFCNTACC = '1'
249   C                   EVAL      PDFCHGCMT = '1'
250   C                   EVAL      PDFDOCASB = '0'
251   C*
252   C*     Directory for Body and Attachment files
253   C                   EVAL      DIRPTHOFF = 1578
254   C                   EVAL      DIRPTHLEN = 80
255   C                   EVAL      DIRPTH = '/most'
256   C*
257   C*     Set up body files
258   C                   EVAL      BDYPTHOFF = 962
259   C                   EVAL      BDYLEN  = 308
260   C                   EVAL      BDYNUM  = 3
261   C*
262   C*       Set up 1st body file
263   C                   EVAL      BDYX1   = 100
264   C                   EVAL      BDYY1   = 20
265   C                   EVAL      BDYOFF1 = 20
266   C                   EVAL      BDYUSE1 = '1'
267   C                   EVAL      BDYRES1 = X'000000'
268   C                   EVAL      BDYPTH1 = 'sample.htm'
269   C                   EVAL      BDYPLEN1= %len(%trim(bdypth1))
270   C*       Set up 2nd body file
271   C                   EVAL      BDYX2   = 100
272   C                   EVAL      BDYY2   = 20
273   C                   EVAL      BDYOFF2 = 20
274   C                   EVAL      BDYUSE2 = '0'
275   C                   EVAL      BDYRES2 = X'000000'
276   C                   EVAL      BDYPTH2 = '/More/iris.jpg'
277   C                   EVAL      BDYPLEN2= %len(%trim(bdypth2))
278   C*       Set up 3rd body file
279   C                   EVAL      BDYX3   = 100
280   C                   EVAL      BDYY3   = 20
281   C                   EVAL      BDYOFF3 = 20
282   C                   EVAL      BDYUSE3 = '1'
283   C                   EVAL      BDYRES3 = X'000000'
284   C                   EVAL      BDYPTH3 = 'ascii.txt'
```

```
285   C                   EVAL      BDYPLEN3= %len(%trim(bdypth3))
286   C*
287   C*     Set up attachment files
288   C                   EVAL      ATTPTHOFF = 1270
289   C                   EVAL      ATTLEN  = 108
290   C                   EVAL      ATTNUMB = 1
291   C*
292   C*       Set up 1st attachment file
293   C                   EVAL      ATTX1   = 100
294   C                   EVAL      ATTY1   = 20
295   C                   EVAL      ATTOFF1 = 20
296   C                   EVAL      ATTUSE1 = '1'
297   C                   EVAL      ATTRES1 = X'000000'
298   C                   EVAL      ATTPTH1 = 'prices.123'
299   C                   EVAL      ATTPLEN1= %len(%trim(attpth1))
300   C*       Set up 2nd attachment file
301   C                   EVAL      ATTX2   = 100
302   C                   EVAL      ATTY2   = 20
303   C                   EVAL      ATTOFF2 = 20
304   C                   EVAL      ATTUSE2 = '0'
305   C                   EVAL      ATTRES2 = X'000000'
306   C                   EVAL      ATTPTH2 = ''
307   C                   EVAL      ATTPLEN2= %len(%trim(attpth2))
308   C*       Set up a 3rd attachment file
309   C                   EVAL      ATTX3   = 100
310   C                   EVAL      ATTY3   = 20
311   C                   EVAL      ATTOFF3 = 20
312   C                   EVAL      ATTUSE3 = '1'
313   C                   EVAL      ATTRES3 = X'000000'
314   C                   EVAL      ATTPTH3 = ''
315   C                   EVAL      ATTPLEN3= %len(%trim(attpth3))
316   C*
317   C                   ELSE
318   C*
319   C*   Do not e-mail
320   C                   EVAL      DISPOSTN = '0'
321   C                   EVAL      MSGLEN    = 0
322   C                   EVAL      ADDRLEN   = 0
323   C                   EVAL      MSGTEXT  = ''
324   C                   EVAL      ADDRESS = ''
325   C*
326   C                   ENDIF
327   C*
328   C* Check customer preference for storage in IFS
329   C     DISPIFS       IFEQ      'Y'
330   C*
331   C*   Store the PDF file as a Stream File
332   C                   EVAL      DSPPDFSTMF = '1'
333   C*     Rename & set Authority for STMF
334   C                   EVAL      STMFNAMOFF = 1658
335   C                   EVAL      PDFSTMFNAM = '/invoices/2004-01/'
336   C                             + %SUBST(rtgtag:7:5) + '.pdf'
337   C                   EVAL      STMFNAMLEN = 80
338   C                   EVAL      PUBAUTOFF  = 1818
339   C                   EVAL      PDFPUBAUT = '*R'
340   C                   EVAL      PUBAUTLEN  = %len(%trim(pdfpubaut))
341   C*
342   C                   ELSE
343   C*
344   C*   Do not store as stream file
```

```
C                   EVAL      DSPPDFSTMF = '0'
C                   EVAL      STMFNAMOFF = 0
C                   EVAL      STMFNAMLEN = 0
C*
C                   ENDIF
C*
C* Check customer preference for creation of PDF Spooled File
C     DISPPDF       IFEQ      'Y'
C*
C*   Create a PDF spooled file
C                   EVAL      DSPPDFSPLF = '1'
C*
C*     Attributes for PDF Spooled file
C                   EVAL      PDFSPLOFF  = 1828
C                   EVAL      PDFSPLLEN  = 305
C                   EVAL      PDFOUTQ    = 'PDFOUTQ'
C                   EVAL      PDFOUTQLIB = 'qgpl'
C                   EVAL      PDFSPLNAM  = 'PDFSPLF'
C                   EVAL      PDFUSRDTA  = 'User Data'
C                   EVAL      PDFUSRDFN  = 'PDF User Dfn Dta'
C                   EVAL      PDFFORM    = '*SPLF'
C*
C                   ELSE
C*
C*   Do not create PDF Spooled file
C                   EVAL      DSPPDFSPLF = '0'
C                   EVAL      PDFSPLOFF  = 0
C                   EVAL      PDFSPLLEN  = 0
C*
C                   ENDIF
C*
C* Check customer preference for creation of AFP Spooled File
C     DISPAFP       IFEQ      'Y'
C*
C*   Create an AFP spooled file
C                   EVAL      DSPAFPSPLF = '1'
C*
C*     Attributes for AFP Spooled file
C                   EVAL      AFPSPLOFF  = 2133
C                   EVAL      AFPSPLLEN  = 305
C                   EVAL      AFPOUTQ    = 'AFPOUTQ'
C                   EVAL      AFPOUTQLIB = 'qgpl'
C                   EVAL      AFPSPLNAM  = 'AFPSPLF'
C                   EVAL      AFPUSRDTA  = 'User Data'
C                   EVAL      AFPUSRDFN  = 'AFP User Dfn Dta'
C                   EVAL      AFPFORM    = '*SPLF  '
C*
C                   ELSE
C*
C*   Do not create AFP Spooled file
C                   EVAL      DSPAFPSPLF = '0'
C                   EVAL      AFPSPLOFF  = 0
C                   EVAL      AFPSPLLEN  = 0
C*
C                   ENDIF
C*
C*
C*
C                   RETURN
```

Data description specifications for LOOKUP file

Example H-2 shows the data description specifications (DDS) used to create the lookup file used in conjunction with the Portable Document Format (PDF) mapping program. The file is used to determine the routing for a PDF file: send as an e-mail attachment, store in the integrated file system (IFS), print as a PDF spooled file, or print as an AFPDS spooled file.

Example: H-2 DDS for external LOOKUP file

```
.....A..........T.Name++++++RLen++TDpB......Functions++++++++++++++++++++++++++
     A                                      UNIQUE
     A          R LOOKUPR
     A            CUSTNO         6A         COLHDG('CUSTOMER' 'NUMBER')
     A            COMP          25A         COLHDG('COMPANY' 'NAME')
     A            PERSON        25A         COLHDG('CONTACT')
     A            EADDR         80A         COLHDG('EMAIL' 'ADDRESS')
     A            DISPEMAIL      1A         COLHDG('SEND AS E-MAIL Y/N')
     A                                      VALUES('Y' 'N')
     A            DISPIFS        1A         COLHDG('STORE IN IFS   Y/N')
     A                                      VALUES('Y' 'N')
     A            DISPPDF        1A         COLHDG('PRINT AS PDF   Y/N')
     A                                      VALUES('Y' 'N')
     A            DISPAFP        1A         COLHDG('PRINT AS AFP   Y/N')
     A                                      VALUES('Y' 'N')
     A          K CUSTNO
```

Spreadsheet for mapping program calculations

The charts in Figure H-1, Figure H-2, and Figure H-3 illustrate the spreadsheet that was used to plan for the sample PDF mapping program. A softcopy version, in both Lotus 1-2-3 and Microsoft Excel, is available for download as explained in Appendix I, “Additional material” on page 193.

PDF Mapping program output data structure.

Description	Offset Within Section	Final Offset	type	Allowance or Length	Start Post'n	End Post'n	Sample Value	Field Name
Disposition of PDF e-mail	0	0	CHAR	1	1	1	'1'	DISPOSTN
More processing	1	1	CHAR	1	2	2	'0'	CALLAGIN
Reserved - set to X'00's	2	2	CHAR	2	3	4	X'0000'	RES2
Length of message text	4	4	Binary	4	5	8	255	MSGLEN
Length of mail address	8	8	Binary	4	9	12	255	ADDRLEN
Message text	12	12	CHAR	255	13	267	'This is your invoice.'	MSGTEXT
Reserved - set to X'00's	267	267	CHAR	1	268	268	X'00'	RES3
Offset to extension area	268	268	Binary	4	269	272	542	EXTOFF
CCSID of mssage text & subject	272	272	Binary	4	273	276	0	CCSID
Disposition of PDF Stream file	276	276	Char	1	277	277	'1'	DSPPDFSTMF
Disposition of PDF Spooled file	277	277	Char	1	278	278	'0'	DSPPDFSPLF
Disposition of PDF Error	278	278	Char	1	279	279	'0'	DSPPDFERR
Dispostion of AFPDF Spooled file	279	279	Char	1	280	280	'0'	DSPAFPSPLF
Reserved - set to X'00's	280	280	CHAR	7	281	287	X'00000000000000'	RES4
E-mail address * (can be 16M)	287	287	CHAR	255	288	542	'"user1@domain.com"'	ADDRESS
Length of Extension area format	0	542	Binary	4	543	546	100	EXTLEN
Offset to subject	4	546	Binary	4	547	550	642	SUBOFF
Length of subject	8	550	Binary	4	551	554	80	SUBLEN
Offset to Reply to e-mail address	12	554	Binary	4	555	558	722	RPLYOFF
Length of Reply to e-mail address	16	558	Binary	4	559	562	80	RPLYLEN
Offset to CC e-mail address	20	562	Binary	4	563	566	802	CCOFF
Length of CC e-mail address	24	566	Binary	4	567	570	80	CCLEN
Offset to BCC e-mail address	28	570	Binary	4	571	574	882	BCCOFF
Length of BCC e-mail address	32	574	Binary	4	575	578	80	BCCLEN
Offset to list of path names for body o e-mail	36	578	Binary	4	579	582	962	BDYPTHOFF
Offset to path name for directory for files	40	582	Binary	4	583	586	1578	DIRPTHOFF
Length of path name for directory for files	44	586	Binary	4	587	590	80	DIRPTHLEN
Offset to list of path names of attachments	48	590	Binary	4	591	594	1270	ATTPTHOFF
Offset to PDF stream file path and file name	52	594	Binary	4	595	598	1658	STMFNAMOFF
Length of PDF stream file path and file name	56	598	Binary	4	599	602	80	STMFNAMLEN
Offset to file name for e-mailed PDF attachment	60	602	Binary	4	603	606	1738	EATTNAMOFF
Length of file name for e-mailed PDF attachment	64	606	Binary	4	607	610	80	EATTNAMLEN
Offset to PDF file public authority	68	610	Binary	4	611	614	1818	PUBAUTOFF
Length of PDF file public authority	72	614	Binary	4	615	618	2	PUBAUTLEN

Figure H-1 PDF mapping program spreadsheet (Part 1 of 3)

Offset to spooled file PDF distribution	76	618	Binary	4	619	622	1828	PDFSPLOFF
Length of spooled file PDF distribution	80	622	Binary	4	623	626	305	PDFSPLLEN
Offset to spooled file AFP distribution	84	626	Binary	4	627	630	2133	AFPSPLOFF
Length of spooled file AFP distribution	88	630	Binary	4	631	634	305	AFPSPLLEN
Offset to PDF encryption information	92	634	Binary	4	635	638	2438	ENCRPTOFF
Length of PDF encryption information	96	638	Binary	4	639	642	71	ENCRPTLEN
Subject text		642	Char	80	643	722	'Sample V5R2 Plus e-m	EXTSUBJ
Reply to e-mail address		722	Char	80	723	802	""user2@domain.com"	EXTRPLY
CC e-mail address(es)		802	Char	80	803	882	""user3@domain.com"	EXTCC
BCC e-mail address(es)		882	Char	80	883	962	""user4@domain.com"	EXTBCC
Total Length of stream file info for body		962	Binary	4	963	966	308	BDYLEN
Number of stream files specified for body		966	Binary	4	967	970	3	BDYNUM
Total length of individual stream file format	0	970	Binary	4	971	974	100	BDYX1
Length of individual stream file format	4	974	Binary	4	975	978	20	BDYY1
Offset to path name	8	978	Binary	4	979	982	20	BDYOFF1
Length of path name	12	982	Binary	4	983	986	10	BDYPLEN1
Use specified directory (0 or 1)	16	986	Char	1	987	987	'1'	BDYUSE1
Reserved - set to X'00's	17	987	Char	3	988	990	X'000000'	BDYRES1
Path name	20	990	Char	80	991	1070	'sample.htm'	BDYPTH1
Total length of individual stream file format	0	1070	Binary	4	1071	1074	100	BDYX2
Length of individual stream file format	4	1074	Binary	4	1075	1078	20	BDYY2
Offset to path name	8	1078	Binary	4	1079	1082	20	BDYOFF2
Length of path name	12	1082	Binary	4	1083	1086	14	BDYPLEN2
Use specified directory (0 or 1)	16	1086	Char	1	1087	1087	'0'	BDYUSE2
Reserved - set to X'00's	17	1087	Char	3	1088	1090	X'000000'	BDYRES2
Path name	20	1090	Char	80	1091	1170	'/more/iris.jpg'	BDYPTH2
Total length of individual stream file format	0	1170	Binary	4	1171	1174	100	BDYX3
Length of individual stream file format	4	1174	Binary	4	1175	1178	20	BDYY3
Offset to path name	8	1178	Binary	4	1179	1182	20	BDYOFF3
Length of path name	12	1182	Binary	4	1183	1186	9	BDYPLEN3
Use specified directory (0 or 1)	16	1186	Char	1	1187	1187	'0'	BDYUSE3
Reserved - set to X'00's	17	1187	Char	3	1188	1190	X'000000'	BDYRES3
Path name	20	1190	Char	80	1191	1270	'ascii.txt'	BDYPTH3
Total Length of stream file info for attachments		1270	Binary	4	1271	1274	108	ATTLEN
Number of stream files specified for attachments		1274	Binary	4	1275	1278	1	ATTNUM
Total length of individual stream file format	0	1278	Binary	4	1279	1282	100	ATTX1
Length of individual stream file format	4	1282	Binary	4	1283	1286	20	ATTY1
Offset to path name	8	1286	Binary	4	1287	1290	20	ATTOFF1
Length of path name	12	1290	Binary	4	1291	1294	10	ATTPLEN1
Use specified directory (0 or 1)	16	1294	Char	1	1295	1295	'1'	ATTUSE1
Reserved - set to X'00's	17	1295	Char	3	1296	1298	X'000000'	ATTRES1
Path name	20	1298	Char	80	1299	1378	'prices.123'	ATTPTH1

Figure H-2 PDF mapping program spreadsheet (Part 2 of 3)

Total length of individual stream file format	0	1378	Binary	4	1379	1382	100	ATTX2
Length of individual stream file format	4	1382	Binary	4	1383	1386	20	ATTY2
Offset to path name	8	1386	Binary	4	1387	1390	20	ATTOFF2
Length of path name	12	1390	Binary	4	1391	1394	0	ATTPLEN2
Use specified directory (0 or 1)	16	1394	Char	1	1395	1395	'0'	ATTUSE2
Reserved - set to X'00's	17	1395	Char	3	1396	1398	X'000000'	ATTRES2
Path name	20	1398	Char	80	1399	1478	''	ATTPTH2
Total length of individual stream file format	0	1478	Binary	4	1479	1482	100	ATTX3
Length of individual stream file format	4	1482	Binary	4	1483	1486	20	ATTY3
Offset to path name	8	1486	Binary	4	1487	1490	20	ATTOFF3
Length of path name	12	1490	Binary	4	1491	1494	0	ATTPLEN3
Use specified directory (0 or 1)	16	1494	Char	1	1495	1495	'0'	ATTUSE3
Reserved - set to X'00's	17	1495	Char	3	1496	1498	X'000000'	ATTRES3
Path name	20	1498	Char	80	1499	1578	''	ATTPTH3
Directory for files		1578	Char	80	1579	1658	'/most'	DIRPTH
PDF stream file path and name		1658	Char	80	1659	1738	/invoices/2004-01/nnnnn	PDFSTMFNAM
Name for PDF attachment for e-mail		1738	Char	80	1739	1818	Invoice #nnnnn.pdf'	PDFATTNAM
PDF file public authority		1818	Char	10	1819	1828	'*R'	PDFPUBAUT
Name of output queue for spooled PDF file	0	1828	Char	10	1829	1838	'PDFOUTQ'	PDFOUTQ
Output queue library	10	1838	Char	10	1839	1848	'qgpl'	PDFOUTQLIB
Spooled file name	20	1848	Char	10	1849	1858	PDFSPLF'	PDFSPLNAM
User data	30	1858	Char	10	1859	1868	'User Data'	PDFUSRDTA
User defined data	40	1868	Char	255	1869	2123	'Pdf User Dfn Dta'	PDFUSRDFN
Form type	295	2123	Char	10	2124	2133	'*SPLF'	PDFFORM
Name of output queue on which to spool AFP	0	2133	Char	10	2134	2143	'AFPOUTQ'	AFPOUTQ
Output queue library	10	2143	Char	10	2144	2153	'qgpl'	AFPOUTQLIB
Spooled file name	20	2153	Char	10	2154	2163	'AFPSPLF'	AFPSPLNAM
User Data	30	2163	Char	10	2164	2173	'User Data'	AFPUSRDTA
User defined data	40	2173	Char	255	2174	2428	'AFP User Dfn Dta'	AFPUSRDFN
Form type	295	2428	Char	10	2429	2438	'*SPLF'	AFPFORM
PDF master password	0	2438	Char	32	2439	2470	'master'	PDFMASTPW
PDF user password	32	2470	Char	32	2471	2502	''	PDFUSRPW
PDF print	64	2502	Char	1	2503	2503	'1'	PDFPRT
PDF document change	65	2503	Char	1	2504	2504	'0'	PDFDOCCHG
PDF copy	66	2504	Char	1	2505	2505	'0'	PDFCOPY
PDF encrypition level	67	2505	Char	1	2506	2506	'2'	PDFENCLVL
PDF content access enablement	68	2506	Char	1	2507	2507	'1'	PDFCNTACC
PDF change comments	69	2507	Char	1	2508	2508	'1'	PDFCHGCMT
PDF document assembly	70	2508	Char	1	2509	2509	'0'	PDFDOCASB

Figure H-3 PDF mapping program spreadsheet (Part 3 of 3)

Tip: If you are using relative notation in your description specifications, remember that any field that is listed here as Binary 4 needs to be coded as having a length of 9. See 6.3.2, “Binary fields” on page 65.

Output queue monitor for fax

The following two programs work together to take the output from Infoprint Server and send the desired document or segment as a fax. For a further description about this process, see Chapter 7, “Using intelligent routing with Facsimile Support/400” on page 99.

Mainline monitor program

The program shown in Example H-3 receives information from a data queue that is associated with an output queue whenever a spooled file is placed in the output queue in RDY state. It calls a second program to retrieve the user-defined data for that spooled file. Then it uses the information to submit a request to Facsimile Support/400.

Example: H-3 Mainline output monitor program for fax XFER21F

```
            PGM

            DCL        &DTAQNAME  *CHAR   10  VALUE(FAXMON)
            DCL        &DTAQLIB   *CHAR   10  VALUE(xfer2fax  )
            DCL        &ENTLEN    *DEC     5  VALUE(128)
            DCL        &ENTRY     *CHAR  128
            DCL        &WAIT      *DEC     5  VALUE(300)
            DCL        &ENDSTS    *CHAR    1

            DCL        &JOB       *CHAR   10
            DCL        &USER      *CHAR   10
            DCL        &JOBNBR    *CHAR    6
            DCL        &QJOB      *CHAR   26    /* QUALIFIED JOB */
            DCL        &FILE      *CHAR   10
            DCL        &OUTQ      *CHAR   10
            DCL        &SPLNBRB   *CHAR    4    /* BINARY    */
            DCL        &SPLNBRD   *DEC     9    /* DECIMAL   */
            DCL        &SPLNBRC   *CHAR    4    /* CHARACTER */
            DCL        &USRDFNDTA *CHAR  255    /* user-defined data   */
            DCL        &FAXNO     *CHAR   20    /* fax number          */
            DCL        &FAXUSR    *CHAR   20    /* Fax recipient       */
            DCL        &XOPT      *CHAR    7    /* Error code from QUSRSPLA */
            DCL        &XCD       *CHAR    1    /* E for Error  */

/* ------------------------------------------------------------------ */
/* 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     &JOB      VALUE(%SST(&ENTRY 13 10))
            CHGVAR     &USER     VALUE(%SST(&ENTRY 23 10))
            CHGVAR     &JOBNBR   VALUE(%SST(&ENTRY 33 6))
            CHGVAR     &FILE     VALUE(%SST(&ENTRY 39 10))
            CHGVAR     &OUTQ     VALUE(%SST(&ENTRY 53 10))
            CHGVAR     &SPLNBRB  VALUE(%SST(&ENTRY 49 4))
            CHGVAR     &SPLNBRD  VALUE(%BIN(&SPLNBRB))
            CHGVAR     &SPLNBRC  &SPLNBRD
            CHGVAR     &QJOB     VALUE(&JOB *CAT &USER *CAT &JOBNBR)

/* ------------------------------------------------------------------ */
/* Call XFER22F to retrieve the User Defined Data                     */
/* ------------------------------------------------------------------ */
            CALLPRC    PRC(XFER22F) PARM(&FILE &QJOB &SPLNBRD +
                         &USRDFNDTA &XCD &XOPT)

/* ------------------------------------------------------------------ */
/* Check error code                                                   */
/* ------------------------------------------------------------------ */
            IF         COND(&XCD *EQ 'E') THEN(GOTO CMDLBL(ERROR))

/* ------------------------------------------------------------------ */
/* Move to spooled file to unmonitored output queue.                  */
/* ------------------------------------------------------------------ */
            CHGSPLFA   FILE(&FILE) JOB(&JOBNBR/&USER/&JOB) +
                         SPLNBR(&SPLNBRC) OUTQ(XFER2FAX/FAXMON2)
/* ------------------------------------------------------------------ */
/* Submit Fax Command                                                 */
/* ------------------------------------------------------------------ */
            CHGVAR     &FAXNO    VALUE(%SST(&USRDFNDTA 1  20))
            CHGVAR     &FAXUSR   VALUE(%SST(&USRDFNDTA 21 20))

            SBMFAX     TYPE(*BCH) OPTION(*SPLEXIST) DSTSELMTH(*EXT) +
                         DEST(&FAXUSR) ID('identifier me') +
                         TEXT(DESCRIPTION) TO(&FAXNO &FAXUSR) +
                         FROM('Payday Pete' 'Super Sun Seeds') +
                         TITLE(TITLE) COMMENT(COMMENT) FILE(&FILE) +
                         JOB(&JOBNBR/&USER/&JOB) SPLNBR(&SPLNBRC)
            MONMSG     MSGID(FRQ0017 CPF0001) EXEC(GOTO CMDLBL(ERROR))

/* ------------------------------------------------------------------ */
/* GO READ NEXT ENTRY IN THE DTAQ                                     */
/* ------------------------------------------------------------------ */
            GOTO       CMDLBL(RCVDTAQ)

/* ------------------------------------------------------------------ */
/* Error handling                                                     */
/* ------------------------------------------------------------------ */
 ERROR:     SNDMSG     MSG('Error'                    ) +
                         TOMSGQ(MIRA) MSGTYPE(*INFO)
            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))

END:        ENDPGM
```

Program to retrieve spooled file attributes

The program in Example H-4 calls the QUSRSPLA application programming interface (API) to retrieve the spooled file attributes. It returns the user-defined data to the mainline program.

Example: H-4 Program to retrieve spooled file attributes XFER22F

```
     D RCVAR2          DS
     D  SPLFID               25     40
     D QUSUDD               1157   1411
     D*                                        Usr Defined Data
     D ERRCOD          DS
     D  ERRBYT                1      4B 0
     D  BYTAVL                5      8B 0
     D  EXCPT                 9     15
     D  RESVD                16     16
     D                 DS
     D  SPLNOB                1      4B 0
     D  RCVLE2               35     38B 0
     C*---------------------------------------------------*
     C     *ENTRY        PLIST
     C                   PARM                    FILE           10
     C                   PARM                    JOBINF         26
     C                   PARM                    SPLNOD          9 0
     C                   PARM                    USRDFNDTA      255
     C                   PARM                    XCD             1
     C                   PARM                    XOPT            7
     C*---------------------------------------------------*
     C*RETRIEVE COPIES AND PAGE WIDTH FROM SPOOL FILE ATTRIBUTES.
     C*---------------------------------------------------*
     C     COPIES        TAG
     C                   Z-ADD     1411          RCVLE2
     C                   MOVEL     'SPLA0100'    FMTNM1          10
     C                   MOVE      *BLANKS       IJOBID          16
     C                   MOVE      *BLANKS       SPLFID
     C                   Z-ADD     SPLNOD        SPLNOB
     C*
     C                   Z-ADD     16            ERRBYT
     C                   CALL      'QUSRSPLA'
     C                   PARM                    RCVAR2
     C                   PARM                    RCVLE2
     C                   PARM                    FMTNM1
     C                   PARM                    JOBINF
     C                   PARM                    IJOBID
     C                   PARM                    SPLFID
     C                   PARM                    FILE
     C                   PARM                    SPLNOB
     C                   PARM                    ERRCOD
     C*
     C     BYTAVL        IFNE      *ZERO
     C                   MOVE      'E'           XCD
```

```
C                   MOVEL     EXCPT         XOPT
C                   ELSE
C*
C                   EVAL      USRDFNDTA = QUSUDD
C                   END
C*
C                   RETURN
```

Program to submit STOP to data queue

The program shown in Example H-5 calls the QSNDDTAQ API to put a STOP entry into the FAX monitor data queue.

Example: H-5 CL program to submit STOP to the data queue

```
            PGM
            DCL        VAR(&QNAME) TYPE(*CHAR) LEN(10) +
                         VALUE(FAXMON)
            DCL        VAR(&QLIB) TYPE(*CHAR) LEN(10) +
                         VALUE(XFER2FAX  )
            DCL        VAR(&MSGLEN)  TYPE(*DEC)  LEN(5 0) VALUE(4)
            DCL        VAR(&MSG)     TYPE(*CHAR) LEN(4)   VALUE(STOP)

/* ------------------------------------------------------------------ */
            CALL PGM(QSNDDTAQ) PARM(&QNAME &QLIB &MSGLEN &MSG)

 END:       ENDPGM
```

Using APIs to add entries to a PDF mapping object

Two programs were used to add entries to a PDF mapping object using the system API QPQAPME. The main line RPGLE program does most of the work. A secondary program sends a message to the operator to indicate whether the process was successful.

RPG LE program to call API QPQAPME

Example H-6 is a listing of the sample program used to call the QPQAPME API to add an entry to a PDF mapping object.

Example: H-6 RPGLE program ADDMEAPI

```
F*-----------------------------------------------*
F*   File specification for log file-----------*
F*-----------------------------------------------*
FLOGAPI    O    E             DISK
D*-----------------------------------------------*
 * Copy member QPQAPME from QRPGLESRC in QSYSINC.  This member includes
 * data structures:
 *    - QPQAPMIE   (Qpq APMail Information Entry)
 *    - QPQEO100   (Structure for APMEO100 - spooled file attributes)
 *    - QPQPO100   (Structure for PMAPO100 Format - Mapping action)
 *  (because of pending PTF used personal copy of QPQAPME)
D*-----------------------------------------------*
```

```
D/COPY MIRA/QRPGLESRC,QPQAPME
D ERRCOD          DS
D  ERRBYT                 1      4B 0
D  BYTAVL                 5      8B 0
D  EXCPT                  9     15
D  RESVD                 16     16
D  ERRDTA                17    116
D                 DS
D  SPLNOB                 1      4B 0
D  RCVLE2                35     38B 0
C*----------------------------------------------------*
D PDFMAP          DS
D  PDFMAPOBJ                    10
D  PDFMAPLIB                    10
D*----------------------------------------------------*
DRLIB             S              10
DRLIBMSG          S              10
D*----------------------------------------------------*
D  ATTRIBLEN      S               9B 0
D  ATTRIBFMT      S               8
D  ACTIONLEN      S               9B 0
D  ACTIONFMT      S               8
D  ADDTYPE        S               9B 0
D*----------------------------------------------------*
D*Additional fields for e-mail
D*----------------------------------------------------*
D MAILDS          DS
D  MSGTXT                      255
D  SUBJECT                      80
D  REPLYTO                      80
D  TOADDR1                      80
D  TOADDR2                      80
D  BCCADDR1                     80
D  BDYFILE1                    255
D  BDYFILE2                    255
D  BDYFILE3                    255
D  ATTFILE1                    255
D  CCFILE                      255
D  BDYRES3                      80
D*----------------------------------------------------*
D*Fields for PDF stream file
D*----------------------------------------------------*
D STMFDS          DS
D  PDFOFF                       9B 0
D  PDFLEN                       9B 0
D  PUBAUT                      10
D  PDFRES                       2
D  PDFFILE                     255
D*----------------------------------------------------*
D*PDF spooled file
D*----------------------------------------------------*
D PDFSPLFDS       DS
D  PDFOUTQ                      10
D  PDFOUTQLIB                   10
D  PDFSPLF                      10
D  PDFUSRDTA                    10
D  PDFFORM                      10
D  PDFUSRDFN                   255
D  PDFSPLRES                     3
D*----------------------------------------------------*
```

```
D*AFPDS spooled file
D*------------------------------------------------------*
D AFPSPLFDS       DS
D  AFPOUTQ                       10
D  AFPOUTQLIB                    10
D  AFPSPLF                       10
D  AFPUSRDTA                     10
D  AFPFORM                       10
D  AFPUSRDFN                    255
D  AFPSPLRES                      3
D*------------------------------------------------------*
D*PDF mapping action data structure
D*------------------------------------------------------*
D ACTION          DS
D  AQPQPO100                    132
D  AQPQAPMIE                    256
D  AMAILDS                     2010
D  ASTMFDS                      275
D  APDFSPLFDS                   308
D  AAFPSPLFDS                   308
C*------------------------------------------------------*
C*Qualified PDF map object name
C*------------------------------------------------------*
C                   EVAL      PDFMAPOBJ = 'PDFMAP'
C                   EVAL      PDFMAPLIB = 'MIRA'
C*------------------------------------------------------*
C*PDF map entry spooled file attributes (DS QPQEO100)
C*------------------------------------------------------*
C                   EVAL      QPQSNBR    = 110
C                   EVAL      QPQOQNOO   = '*ALL'
C                   EVAL      QPQOQLIB   = '*ALL'
C                   EVAL      QPQSFILN   = 'MAPME'
C                   EVAL      QPQJN      = '*ALL'
C                   EVAL      QPQUN      = '*ALL'
C                   EVAL      QPQUD      = '*ALL'
C                   EVAL      QPQFT      = '*ALL'
C                   EVAL      QPQMT      = '100'
C                   EVAL      QPQERVED18 = *ALLx'00'
C*------------------------------------------------------*
C*Length of PDF map entry spooled file attributes
C*------------------------------------------------------*
C                   EVAL      ATTRIBLEN  = 326
C*------------------------------------------------------*
C*Format of PDf map entry spooled file attributes
C*------------------------------------------------------*
C                   EVAL      ATTRIBFMT  = 'APMEO100'
C*------------------------------------------------------*
C*PDF mapping action (DS QPQPO100)
C*------------------------------------------------------*
C* Mail info offset & length
C                   EVAL      QPQOMI      =   132
C                   EVAL      QPQLMI      =   256
C*                  EVAL      QPQOMI      =   0
C*                  EVAL      QPQLMI      =   0
C* PDF stream file info offset & length
C                   EVAL      QPQPDFSI    =   2398
C                   EVAL      QPQPDFSIOO  =   20
C*                  EVAL      QPQPDFSI    =   0
C*                  EVAL      QPQPDFSIOO  =   0
C* PDF spooled filE info offset & length
```

```
C                   EVAL      QPQPDFSI01  =   2673
C                   EVAL      QPQPDFSI02  =   308
C*                  EVAL      QPQPDFSI01  =   0
C*                  EVAL      QPQPDFSI02  =   0
C* AFP spooled filE info offset & length
C                   EVAL      QPQPDSSI    =   2981
C                   EVAL      QPQPDSSI00  =   308
C*                  EVAL      QPQPDSSI    =   0
C*                  EVAL      QPQPDSSI00  =   0
C* segmented?, description & reserved
C                   EVAL      QPQSE       =   '1'
C                   EVAL      QPQTD       =   'Entry added by API'
C                   EVAL      QPQERVED19  =   *ALLx'00'
C*-------------------------------------------------*
C*Mail Information Entry
C*-------------------------------------------------*
C                   EVAL      QPQOMT      =   388
C                   EVAL      QPQLMT      =   255
C                   EVAL      QPQOS       =   643
C                   EVAL      QPQLS       =   80
C                   EVAL      QPQDMTAS    =   0
C                   EVAL      QPQORTA     =   623
C                   EVAL      QPQLRTA     =   80
C                   EVAL      QPQOTOA     =   803
C                   EVAL      QPQNTOA     =   2
C                   EVAL      QPQLTOA     =   80
C                   EVAL      QPQOCCA     =   0
C                   EVAL      QPQNCCA     =   0
C                   EVAL      QPQLCCA     =   0
C                   EVAL      QPQOBCCA    =   963
C                   EVAL      QPQNBCCA    =   1
C                   EVAL      QPQLBCCA    =   80
C                   EVAL      QPQOBE      =   1043
C                   EVAL      QPQNBRBE    =   3
C                   EVAL      QPQLBE      =   255
C                   EVAL      QPQOAE      =   1808
C                   EVAL      QPQNBRAE    =   1
C                   EVAL      QPQLAE      =   255
C                   EVAL      QPQOTOS     =   0
C                   EVAL      QPQLTOS     =   0
C                   EVAL      QPQOCCS     =   2063
C                   EVAL      QPQLCCS     =   255
C                   EVAL      QPQOBCCS    =   0
C                   EVAL      QPQLBCCS    =   0
C                   EVAL      QPQPDFFE    =   2318
C                   EVAL      QPQPDFFE00  =   80
C                   EVAL      QPQMS       =   '*PSFCFG'
C                   EVAL      QPQDFFUP    =   '*NONE     '
C                   EVAL      QPQDFFOP    =   'owner'
C                   EVAL      QPQDFFEL    =   '2'
C                   EVAL      QPQPDFFP    =   '1'
C                   EVAL      QPQPDFFC    =   '0'
C                   EVAL      QPQPDFFC00  =   '0'
C                   EVAL      QPQDFFCC    =   '1'
C                   EVAL      QPQDFFCA    =   '1'
C                   EVAL      ASSEMBLY    =   '1'
C                   EVAL      MAILRES     =   *ALLx'00'
C                   EVAL      QPQUTOS     =   '0'
C                   EVAL      QPQUCCS     =   '1'
C                   EVAL      QPQUBCCS    =   '0'
```

```
C                   EVAL      QPQERVED17  =   *ALLx'00'
C*------------------------------------------------------*
C*Standalone Fields for e-mail
C*------------------------------------------------------*
C                   EVAL      MSGTXT      =  'hello world'
C                   EVAL      SUBJECT     =  'Used API'
C                   EVAL      REPLYTO     =  '''fred@sss.com'''
C                   EVAL      TOADDR1     =  '''doraj@improved.ca'''
C                   EVAL      TOADDR2     =  '''backup@improved.ca'''
C                   EVAL      BCCADDR1    =  '''accounting@sss.com'''
C                   EVAL      BDYFILE1    =  '/Most/sample.htm'
C                   EVAL      BDYFILE2    =  '/More/iris.jpg'
C                   EVAL      BDYFILE3    =  '/Most/ascii.txt'
C                   EVAL      ATTFILE1    =  '/Most/prices.123'
C                   EVAL      CCFILE      =  '/Most/cc.txt'
C                   EVAL      BDYRES3     =  '/invoice 31300.pdf'
C*------------------------------------------------------*
C*PDF stream file entry
C*------------------------------------------------------*
C                   EVAL      PDFOFF      =  2418
C                   EVAL      PDFLEN      =  255
C                   EVAL      PUBAUT      =  '*R'
C                   EVAL      PDFRES      =  X'0000'
C                   EVAL      PDFFILE     =  '/invoices/2004-01/31300.pdf'
C*------------------------------------------------------*
C*PDF spooled file
C*------------------------------------------------------*
C                   EVAL      PDFOUTQ     =  'PDFOUTQ'
C                   EVAL      PDFOUTQLIB  =  'QGPL'
C                   EVAL      PDFSPLF     =  '*SPLF'
C                   EVAL      PDFUSRDTA   =  '*SPLF'
C                   EVAL      PDFFORM     =  '*SPLF'
C                   EVAL      PDFUSRDFN   =  '*SPLF'
C                   EVAL      PDFSPLRES   =  X'000000'
C*------------------------------------------------------*
C*AFPDS spooled file
C*------------------------------------------------------*
C                   EVAL      AFPOUTQ     =  'AFPOUTQ'
C                   EVAL      AFPOUTQLIB  =  'QGPL'
C                   EVAL      AFPSPLF     =  '*SPLF'
C                   EVAL      AFPUSRDTA   =  '*SPLF'
C                   EVAL      AFPFORM     =  '*SPLF'
C                   EVAL      AFPUSRDFN   =  '*SPLF'
C                   EVAL      AFPSPLRES   =  X'000000'
C*------------------------------------------------------*
C*Length of PDF mapping attributes
C*------------------------------------------------------*
C                   EVAL      ACTIONLEN  = 3251
C*------------------------------------------------------*
C*Format of PDf map entry spooled file attributes
C*------------------------------------------------------*
C                   EVAL      ACTIONFMT  = 'PMAP0100'
C*------------------------------------------------------*
C*Add Type
C*------------------------------------------------------*
C                   EVAL      ADDTYPE    = 1
C*------------------------------------------------------*
C*Mapping Action
C*------------------------------------------------------*
C                   EVAL      AQPQP0100  = QPQP0100
```

```
C                   EVAL      AQPQAPMIE  = QPQAPMIE
C                   EVAL      AMAILDS    = MAILDS
C                   EVAL      ASTMFDS    = STMFDS
C                   EVAL      APDFSPLFDS = PDFSPLFDS
C                   EVAL      AAFPSPLFDS = AFPSPLFDS
C*--------------------------------------------------*
C                   EVAL      ERRBYT     = 116
C*--------------------------------------------------*
C* Call QPQAPME - Add PDF Map Entry
C*--------------------------------------------------*
C                   CALL      QPQAPME
C                   PARM                    RLIB
C                   PARM                    PDFMAP
C                   PARM                    QPQE0100
C                   PARM                    ATTRIBLEN
C                   PARM                    ATTRIBFMT
C                   PARM                    ACTION
C                   PARM                    ACTIONLEN
C                   PARM                    ACTIONFMT
C                   PARM                    ADDTYPE
C                   PARM                    ERRCOD
C*--------------------------------------------------*
C* Copy ACTION to LOGAPI file
C*--------------------------------------------------*
C                   EVAL      LOGAPIDS =    ACTION
C                   WRITE     LOGAPIR
C*--------------------------------------------------*
C*Check for errors
C*--------------------------------------------------*
C     BYTAVL        IFNE      *ZERO
C                   EVAL      RLIBMSG    = 'error'
C                   EVAL      LOGAPIDS =    ERRCOD
C                   WRITE     LOGAPIR
C                   ELSE
C                   EVAL      RLIBMSG    = RLIB
C                   ENDIF
C*
C                   CALL      'ADDAPIMSG'
C                   PARM                    RLIBMSG
C*
C                   RETURN
```

CL program to send completion

Example H-7 shows a listing of the sample program that is called by ADDMEAPI to send a completion message to the operator.

Example: H-7 CL program ADDAPIMSG

```
            PGM         PARM(&RLIBMSG)
            DCL         VAR(&RLIBMSG) TYPE(*CHAR) LEN(10)

            SNDMSG      MSG('LIBRARY IS ' *CAT &RLIBMSG)            +
                          TOUSR(QUSRSYS) MSGTYPE(*INFO)
            ENDPGM
```

Corrected fields for main information entry structure

At the time that this Redpaper was written, several of the fields at the end of the mail information entry section of the PDF mapping action structure were found as defined incorrectly in the copy of QPQAPME in QSYSINC. Example H-8 lists the corrected versions of the added and changed the fields. A program temporary fix (PTF) is planned to update the source members in QSYSINC. Check with the IBM Support Center for availability.

Example: H-8 Corrected fields in QRPGLESRC.QPQAPME

```
D*                                                  PDF File User Password
D QPQDFFOP              141    150
D*                                                  PDF File Owner Password
D QPQDFFEL              151    151
D*                                                  PDF File Encryption Level
D QPQPDFFP              152    152
D*                                                  PDF File Print
D QPQPDFFC              153    153
D*                                                  PDF File Copy
D QPQPDFFCOO            154    154
D*                                                  PDF File Change
D QPQDFFCC              155    155
D*                                                  PDF File Change Comments
D QPQDFFCA              156    156
D*                                                  PDF File Content Access
D ASSEMBLY              157    157
D*                                                  PDF File Document Assembly
D MAILRES               158    173
D*                                                  Reserved Field
D QPQUTOS               174    174
D*                                                  Use TO Stmf
D QPQUCCS               175    175
D*                                                  Use CC Stmf
D QPQUBCCS              176    176
D*                                                  Use BCC Stmf
D QPQERVED17            177    256
D*                                                  Reserved
D*****************************************************************
```

Planning spreadsheet for API mapping action structure

The spreadsheet shown in Figure H-4 and Figure H-5 was used calculate offset and length values used in the PDF mapping action structure used by the QPQAPME API program.

PDF Mapping program Action data structure.

Description	Offset Within Section	Final Offset	type	Allowance or Length	Start Post'n	End Post'n	Sample Value	Field Name
Offset to mail information entry	0	0	Binary	4	1	4	132	QPQOMI
Length of to mail information entry	4	4	Binary	4	5	8	256	QPQLMI
Offset to PDF stream file information	8	8	Binary	4	9	12	2398	QPQPDFSI
Length of to PDf stream file information	12	12	Binary	4	13	16	20	QPQPDFSI00
Offset to PDF spooled file information	16	16	Binary	4	17	20	2673	QPQPDFSI01
Length of PDF spooled file information	20	20	Binary	4	21	24	308	QPQPDFSI02
Offset to AFPDS spooled file information	24	24	Binary	4	25	28	2981	QPQPDSSI
Length of AFPDS spooled file information	28	28	Binary	4	29	32	308	QPQPDSSI00
Segmented entry	32	32	Char	1	33	33	'1'	QPQSE
Text description	33	33	Char	50	34	83	'Entry added by API'	QPQTD
Reserved	83	83	Char	49	84	132	*ALLx'00'	QPQERVED19
Offset to message text	0	132	Binary	4	133	136	388	QPQOMT
Length of message text	4	136	Binary	4	137	140	255	QPQLMT
Offset to subject	8	140	Binary	4	141	144	643	QPQOS
Length of subject	12	144	Binary	4	145	148	80	QPQLS
CCSID of message text and subject	16	148	Binary	4	149	152	0	QPQDMTAS
Offset to ReplyTo e-mail address	20	152	Binary	4	153	156	723	QPQORTA
Length of ReplyTo e-mail address	24	156	Binary	4	157	160	80	QPQLRTA
Offset to TO e-mail address entries	28	160	Binary	4	161	164	803	QPQOTOA
Number of TO e-mail address entries	32	164	Binary	4	165	168	2	QPQNTOA
Length of TO e-mail address entry	26	168	Binary	4	169	172	80	QPQLTOA
Offset to CC e-mail address entries	40	172	Binary	4	173	176	0	QPQOCCA
Number of CC e-mail address entries	44	176	Binary	4	177	180	0	QPQNCCA
Length of CC e-mail address entry	48	180	Binary	4	181	184	0	QPQLCCA
Offset to BCC e-mail address entries	52	184	Binary	4	185	188	963	QPQOBCCA
Number of BCC e-mail address entries	56	188	Binary	4	189	192	1	QPQNBCCA
Length of BCC e-mail address entry	60	192	Binary	4	193	196	80	QPQLBCCA
Offset to Body of e-mail entries	64	196	Binary	4	197	200	1043	QPQOBE
Number of Body of e-mail entries	68	200	Binary	4	201	204	3	QPQNBRBE
Length of Body of e-mail entry	72	204	Binary	4	205	208	255	QPQLBE
Offset to attachment e-mail entries	76	208	Binary	4	209	212	1808	QPQOAE
Number of attachment e-mail entries	80	212	Binary	4	213	216	1	QPQNBRAE
Length of attachment e-mail entry	84	216	Binary	4	217	220	255	QPQLAE
Offset to TO e-mail stream file	88	220	Binary	4	221	224	0	QPQOTOS
Length of TO e-mail stream file	92	224	Binary	4	225	228	0	QPQLTOS
Offset to CC e-mail stream file	96	228	Binary	4	229	232	2063	QPQOCCS
Length of CC e-mail stream file	100	232	Binary	4	233	236	255	QPQLCCS
Offset to BCC e-mail stream file	104	236	Binary	4	237	240	0	QPQOBCCS
Length of BCC e-mail stream file	108	240	Binary	4	241	244	0	QPQLBCCS
Offset to PDF file for e-mail	112	244	Binary	4	245	248	2318	QPQPDFFE
Length of PDF file for e-mail	116	248	Binary	4	249	252	80	QPQPDFFE00
Mail Sender	120	252	Char	10	253	262	'*PSFCFG'	QPQMS
User password	130	262	Char	10	263	272	'*NONE'	QPQDFFUP
Owner password	140	272	Char	10	273	282	'owner'	QPQDFFOP
Encryption level	150	282	Char	1	283	283	'2'	QPQDFFEL

Figure H-4 PDF mapping action spreadsheet (Part 1 of 2)

Print PDF file	151	283	Char	1	284	284	'1'	QPQPDFFP
Copy PDF file	152	284	Char	1	285	285	'0'	QPQPDFFC
Change PDF file	153	285	Char	1	286	286	'0'	QPQPDFFC00
Change comments in PDF file	154	286	Char	1	287	287	'1'	QPQDFFCC
Content Access to PDF file	155	287	Char	1	288	288	'1'	QPQDFFCA
Document assembly	156	288	Char	1	289	289	'1'	
Reserved	157	289	Char	16	290	305	*ALLx'00'	
Use stream file for TO e-mail address entries	173	305	Char	1	306	306	'0'	QPQUTOS
Use stream file for CC e-mail address entries	174	306	Char	1	307	307	'1'	QPQUCCS
Use stream file for BCC e-mail addres entries	175	307	Char	1	308	308	'0'	QPQUBCCS
Reserved - initialize to null	176	308	Char	80	309	388	*ALLx'00'	QPQERVED17
Message Text		388	Char	255	389	643	'hello world'	MSGTXT
Subject		643	Char	80	644	723	'Used API'	SUBJECT
Reply to		723	Char	80	724	803	'"fred@sss.com"'	REPLYTO
TO e-mail address 1		803	Char	80	804	883	'"doraj@improved.ca'	TOADDR1
TO e-mail address 2		883	Char	80	884	963	'"backup@improved.c	TOADDR2
BCC address 1		963	Char	80	964	1043	'"accounting@sss.co	BCCADDR1
Body file 1		1043	Char	255	1044	1298	'/Most/sample.htm'	BDYFILE1
Body file 2		1298	Char	255	1299	1553	'/More/iris.jpg'	BDYFILE2
Body file 3		1553	Char	255	1554	1808	'/Most/ascii.txt'	BDYFILE3
Attachment file 1		1808	Char	255	1809	2063	'/Most/prices.123'	ATTFILE1
Stream file for CC		2063	Char	255	2064	2318	'/Most/cc.txt'	CCFILE
PDF file for e-mail		2318	Char	80	2319	2398	'/invoice 31300.pdf'	BDYRES3
Offset to PDF file name	0	2398	Binary	4	2399	2402	2418	PDFOFF
Length of PDF file name	4	2402	Binary	4	2403	2406	255	PDFLEN
Public Authority	8	2406	Char	10	2407	2416	'*R'	PUBAUT
Reserved	11	2416	Char	2	2417	2418	X'0000'	PDFRES
PDF file name		2418	Char	255	2419	2673	'/invoices/2004-01/313(	PDFFILE
Output queue for output spooled file	0	2673	Char	10	2674	2683	PDFOUTQ	PDFOUTQ
Output queue library name for output spooled file	10	2683	Char	10	2684	2693	'QGPL'	PDFOUTQLIB
Spooled file name for output spooled file	20	2693	Char	10	2694	2703	'*SPLF'	PDFSPLF
User data for output spooled file	30	2703	Char	10	2704	2713	'*SPLF'	PDFUSRDTA
Form type for output spooled file	40	2713	Char	10	2714	2723	'*SPLF'	PDFFORM
User defined data for output spooled file	50	2723	Char	255	2724	2978	'*SPLF'	PDFUSRDFN
Reserved	305	2978	Char	3	2979	2981	X'000000'	PDFSPLRES
Output queue for output spooled file	0	2981	Char	10	2982	2991	'AFPOUTQ'	AFPOUTQ
Output queue library name for output spooled file	10	2991	Char	10	2992	3001	'QGPL'	AFPOUTQLIB
Spooled file name for output spooled file	20	3001	Char	10	3002	3011	'*SPLF'	AFPSPLF
User data for output spooled file	30	3011	Char	10	3012	3021	'*SPLF'	AFPUSRDTA
Form type for output spooled file	40	3021	Char	10	3022	3031	'*SPLF'	AFPFORM
User defined data for output spooled file	50	3031	Char	255	3032	3286	'*SPLF'	AFPUSRDFN
Reserved	305	3286	Char	3	3287	3289	X'000000'	AFPSPLRES

Figure H-5 PDF mapping action spreadsheet (Part 2 of 2)

Copy encrypted PDF files to IFS

This section contains the sample planning spreadsheet, an RPGLE program, and a CL program that are used to copy encrypted PDF files to the IFS.

Spreadsheet used to plan for COPYEN program

Figure H-6 and Figure H-7 show the spreadsheet that was used to plan for this version of a PDF mapping program. It illustrates the ability to copy an encrypted PDF file to the IFS.

PDF Mapping program encryption only.

Description	Offset Within Section	Final Offset	type	Allowance or Length	Start Post'n	End Post'n	Sample Value	Field Name
Disposition of PDF e-mail	0	0	CHAR	1	1	1	'1'	DISPOSTN
More processing	1	1	CHAR	1	2	2	'0'	CALLAGIN
Reserved - set to X'00's	2	2	CHAR	2	3	4	X'0000'	RES2
Length of message text	4	4	Binary	4	5	8	255	MSGLEN
Length of mail address	8	8	Binary	4	9	12	255	ADDRLEN
Message text	12	12	CHAR	255	13	267	'This is your invoice.'	MSGTEXT
Reserved - set to X'00's	267	267	CHAR	1	268	268	X'00'	RES3
Offset to extension area	268	268	Binary	4	269	272	542	EXTOFF
CCSID of mssage text & subject	272	272	Binary	4	273	276	0	CCSID
Disposition of PDF Stream file	276	276	Char	1	277	277	'1'	DSPPDFSTMF
Disposition of PDF Spooled file	277	277	Char	1	278	278	'0'	DSPPDFSPLF
Disposition of PDF Error	278	278	Char	1	279	279	'0'	DSPPDFERR
Dispostion of AFPDF Spooled file	279	279	Char	1	280	280	'0'	DSPAFPSPLF
Reserved - set to X'00's	280	280	CHAR	7	281	287	X'00000000000000'	RES4
E-mail address * (can be 16M)	287	287	CHAR	255	288	542	'"user1@domain.com"'	ADDRESS
Length of Extension area format	0	542	Binary	4	543	546	100	EXTLEN
Offset to subject	4	546	Binary	4	547	550	0	SUBOFF
Length of subject	8	550	Binary	4	551	554	0	SUBLEN
Offset to Reply to e-mail address	12	554	Binary	4	555	558	0	RPLYOFF
Length of Reply to e-mail address	16	558	Binary	4	559	562	0	RPLYLEN
Offset to CC e-mail address	20	562	Binary	4	563	566	0	CCOFF
Length of CC e-mail address	24	566	Binary	4	567	570	0	CCLEN
Offset to BCC e-mail address	28	570	Binary	4	571	574	0	BCCOFF
Length of BCC e-mail address	32	574	Binary	4	575	578	0	BCCLEN
Offset to list of path names for body o e-mail	36	578	Binary	4	579	582	0	BDYPTHOFF
Offset to path name for directory for files	40	582	Binary	4	583	586	0	DIRPTHOFF
Length of path name for directory for files	44	586	Binary	4	587	590	0	DIRPTHLEN
Offset to list of path names of attachments	48	590	Binary	4	591	594	0	ATTPTHOFF
Offset to PDF stream file path and file name	52	594	Binary	4	595	598	0	STMFNAMOFF
Length of PDF stream file path and file name	56	598	Binary	4	599	602	0	STMFNAMLEN
Offset to file name for e-mailed PDF attachment	60	602	Binary	4	603	606	0	EATTNAMOFF
Length of file name for e-mailed PDF attachment	64	606	Binary	4	607	610	0	EATTNAMLEN

Figure H-6 Spreadsheet used to plan for copying encrypted files to integrated file system (Part 1 of 2)

Offset to PDF file public authority	68	610	Binary	4	611	614	0	PUBAUTOFF
Length of PDF file public authority	72	614	Binary	4	615	618	0	PUBAUTLEN
Offset to spooled file PDF distribution	76	618	Binary	4	619	622	0	PDFSPLOFF
Length of spooled file PDF distribution	80	622	Binary	4	623	626	0	PDFSPLLEN
Offset to spooled file AFP distribution	84	626	Binary	4	627	630	0	AFPSPLOFF
Length of spooled file AFP distribution	88	630	Binary	4	631	634	0	AFPSPLLEN
Offset to PDF encryption information	92	634	Binary	4	635	638	642	ENCRPTOFF
Length of PDF encryption information	96	638	Binary	4	639	642	71	ENCRPTLEN
PDF master password	0	642	Char	32	643	674	'master'	PDFMASTPW
PDF user password	32	674	Char	32	675	706	''	PDFUSRPW
PDF print	64	706	Char	1	707	707	'1'	PDFPRT
PDF document change	65	707	Char	1	708	708	'0'	PDFDOCCHG
PDF copy	66	708	Char	1	709	709	'0'	PDFCOPY
PDF encrypition level	67	709	Char	1	710	710	'2'	PDFENCLVL
PDF content access enablement	68	710	Char	1	711	711	'1'	PDFCNTACC
PDF change comments	69	711	Char	1	712	712	'1'	PDFCHGCMT
PDF document assembly	70	712	Char	1	713	713	'0'	PDFDOCASB

Figure H-7 Spreadsheet used to plan for copying encrypted files to integrated file system (Part 2 of 2)

Note: The fields from SUBOFF through AFPSPLLEN were not used in the program. They are included in the spreadsheet to maintain the correct offsets for the fields that follow in the structure.

Copy encrypted PDF file: RPGLE program

Example H-9 shows the RPGLE source for COPYEN, the program that was used to copy encrypted PDF files to the IFS.

Example: H-9 COPYEN program source

```
D COPYENCL        C                   'QGPL/COPYENCL'
D*
D INPUTDS         DS
D  JOBNAM                 1     26
D  SPLFID                27     36
D  SPLNO                 37     40B 0
D  RTGTAG                41    290
D  PDFFILE              291    630
D  SVRTYPE              631    631
D  RES1                 632    632
D  PATHCCSID            633    636B 0
D  SENDER               637    646
D  USRDTA               647    656
D  SYSNAME              657    664
D  TIMESTMP             655    672
D  OUTQ                 673    682
D  OUTQLIB              683    692
D  RES5                 693    712
D  FORMTYPE             713    722
D*********************************************
D OUTDS           DS
D  DISPOSTN                      1
D  CALLAGIN                      1
```

```
D  RES2                           2
D  MSGLEN                         9B 0
D  ADDRLEN                        9B 0
D  MSGTEXT                      255
D  RES3                           1
D  EXTOFF                         9B 0
D  CCSID                          9B 0
D  DSPPDFSTMF                     1
D  DSPPDFSPLF                     1
D  DSPPDFERR                      1
D  DSPAFPSPLF                     1
D  RES4                           7
D  ADDRESS                      255
D*
D  EXTLEN                         9B 0
D*
D  ENCRPTOFF            635     638B 0
D  ENCRPTLEN            639     642B 0
D*
D  PDFMASTPW                     32
D  PDFUSRPW                      32
D  PDFPRT                         1
D  PDFDOCCHG                      1
D  PDFCOPY                        1
D  PDFENCLVL                      1
D  PDFCNTACC                      1
D  PDFCHGCMT                      1
D  PDFDOCASB                      1
D*
D INPUTLEN        S               9B 0
D OUTPUTLEN       S               9B 0
D OUTINFO         S               9B 0
D*
D NAMELEN         S               4  0
D ENNAME          S             341
D CUSTNO          S               6
D*****************************************
C     *ENTRY        PLIST
C                   PARM                    INPUTDS
C                   PARM                    INPUTLEN
C                   PARM                    OUTDS
C                   PARM                    OUTPUTLEN
C                   PARM                    OUTINFO
C
C* Check to see if the output buffer is large enough.  OUTPUTLEN
C* contains the initial length of the output buffer.  If it is not
C* large enough, return with OUTINFO set to be the length required.
C* This program will be called right back with OUTPUTLEN = OUTINFO.
C*
C                   EVAL      OUTINFO   = 713
C     OUTINFO       IFGT      OUTPUTLEN
C                   RETURN
C                   ENDIF
C*
C* Set data structure to all '00's, to make sure there is no stray data in
C* any unused fields.
C*
C                   MOVE      *ALLx'00'     OUTDS
C*
C* Set more processing (CALLAGIN), reserved fields, and pointer to
```

```
C* extension area.
C                   EVAL      CALLAGIN = '0'
C                   EVAL      RES2     = X'0000'
C                   EVAL      RES3     = X'00'
C                   EVAL      RES4     = X'00000000000000'
C                   EVAL      EXTOFF   = 542
C                   EVAL      EXTLEN   = 100
C*
C* Set disposition flags for IFS, AFP splf, PDF splf, & error
C                   EVAL      DSPPDFSTMF = '0'
C                   EVAL      DSPAFPSPLF = '0'
C                   EVAL      DSPPDFSPLF = '0'
C                   EVAL      DSPPDFERR  = '0'
C*
C     PDFFILE       IFEQ      *BLANKS
C*
C* If  the PDF spooled file is not known, set encryption and
C* pretend to mail the file
C*     Base fields for e-mail
C                   EVAL      DISPOSTN = '1'
C                   EVAL      MSGLEN    = 0
C                   EVAL      ADDRLEN   = 255
C                   EVAL      MSGTEXT = ''
C                   EVAL      CCSID    = 0
C                   EVAL      ADDRESS = '''user@ibm.com'''
C*     Encryption of PDF file for e-mail
C                   EVAL      ENCRPTOFF = 642
C                   EVAL      ENCRPTLEN = 71
C*
C                   EVAL      PDFMASTPW = 'master'
C                   EVAL      PDFUSRPW  = ''
C                   EVAL      PDFPRT    = '1'
C                   EVAL      PDFDOCCHG = '0'
C                   EVAL      PDFCOPY   = '0'
C                   EVAL      PDFENCLVL = '2'
C                   EVAL      PDFCNTACC = '1'
C                   EVAL      PDFCHGCMT = '1'
C                   EVAL      PDFDOCASB = '0'
C*
C* ELSE -- the PDF spooled file is known.  Do no e-mail.
C* Construct name of encrypted version of
C* PDF file, and call CL program to copy it to the integrated file system.
C                   ELSE
C*
C                   EVAL      DISPOSTN = '0'
C*
C                   EVAL      NAMELEN    =  %len(%trimr(PDFFILE))
C                   EVAL      ENNAME     =  %subst(PDFFILE:1:NAMELEN-10)
C                             + 'X' + %subst(PDFFILE:NAMELEN-9:10)
C*
C                   SUBST     RTGTAG        CUSTNO
C*
C                   CALL      COPYENCL
C                   PARM                    ENNAME
C                   PARM                    CUSTNO
C*
C                   ENDIF
C*
C                   RETURN
```

Copy encrypted PDF file: CL program

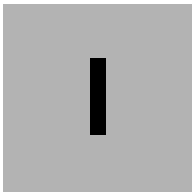
Example H-10 shows the CL program source for COPYENCL. This program is called by COPYEN and does that actual copy of the PDF stream files.

Example: H-10 CL program to copy PDF files

```
            PGM        PARM(&ENCRFILE &CUSTNO)
            DCL        VAR(&ENCRFILE) TYPE(*CHAR) LEN(80)
            DCL        VAR(&CUSTNO) TYPE(*CHAR) LEN(6)
            DCL        VAR(&TARGET)  TYPE(*CHAR) LEN(80 )

 /* Copy using original name   */
            CPY        OBJ(&ENCRFILE) TODIR('/encrypted')
            MONMSG     MSGID(CPFA09C CPFA0A0 CPFA0A9)

 /* Copy and change name       */
            CHGVAR     VAR(&TARGET) VALUE('/encrypted/' *TCAT +
                         &CUSTNO *TCAT  '.PDF')
            CPY        OBJ(&ENCRFILE) TOOBJ(&TARGET)
            MONMSG     MSGID(CPFA09C CPFA0A0 CPFA0A9)
            ENDPGM
```



Additional material

This Redpaper refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this Redpaper is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/REDP3752
```

Alternatively, you can go to the IBM Redbooks Web site at:

```
ibm.com/redbooks
```

Select the **Additional materials** and open the directory that corresponds with the Redpaper form number, RED3752.

Using the Web material

The additional Web material that accompanies this Redpaper includes the following files:

File name	*Description*
REDP3752 sample programs.zip	Softcopies of the sample iSeries source and spreadsheets as described in Appendix H, “Sample programs” on page 163.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: Approximately 150 MB
Operating System: OS/400 Version 5 Release 2 or later
Processor: Any processor supporting V5R2
Memory: Recommended 50 MB of memory for QSPL

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

The objects are grouped in subdirectories by application as described in the following chapters in this book:

- Chapter 6, “Using the PDF mapping program” on page 63
- Chapter 7, “Using intelligent routing with Facsimile Support/400” on page 99
- Chapter 8, “Using APIs to add entries to PDF mapping object” on page 105
- Chapter 9, “Saving encrypted PDF files” on page 111

The subdirectories contain a combination of spreadsheets and iSeries source.

Spreadsheets

Use either Lotus 1-2-3 Release 9.7 for Windows or Microsoft Excel 2000 to open the spreadsheet of your choice.

iSeries source

There are several ways to upload the source member to the iSeries. One method is to use File Transfer Protocol (FTP):

1. On your Windows workstation, click **Start-> run**.
2. At the prompt, type the following command and press Enter:

   ```
   cmd
   ```

3. A Windows system window opens. Start an FTP session with your iSeries by typing either of the following commands:

   ```
   FTP mysystem
   FTP nnn.nnn.nnn.nnn
   ```

4. Enter your user ID and password when prompted.
5. If you are not connected to the desired library, change the library by entering:

   ```
   cd mylib
   ```

6. Transfer the member from your workstation to the iSeries server:

   ```
   put qxxxsrc.member
   ```

7. Repeat step 6 if necessary.
8. To end the FTP session, enter:

   ```
   quit
   ```

9. Exit the Windows dialog by entering:

   ```
   exit
   ```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 196. Note that some of the documents referenced here may be available in softcopy only.

- *IBM AS/400 Printing III*, GG24-4028
- *IBM AS/400 Printing IV*, GG24-4389
- *IBM AS/400 Printing V*, SG24-2160
- *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250

Other publications

These publications are also relevant as further information sources:

- *Infoprint Server for iSeries: Introduction and Planning Guide*, G544-5774
- *Infoprint Server for iSeries: User’s Guide*, G544-5775
- *Infoprint Designer for iSeries: Getting Started*, G544-5773
- *IBM @server iSeries Guide to Output*, S544-5319
- *IBM @server iSeries Printer Device Programming*, SC41-5713
- *AS/400e Facsimile Support for OS/400 Installation Guide*, SC41-0654
- *AS/400e Facsimile Support for OS/400 User’s Guide*, SC41-0655
- *AS/400e Facsimile Support for OS/400 Programmer’s Guide and Reference*, SC41-0656
- The following documents found on the Web at:

  `http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipserveruser_i_ww`

  - *Using Intelligent Routing*
  - *Using Mapping Objects*

Online resources

These Web sites are also relevant as further information sources:

- iSeries printing and output

  `http://www.ibm.com/eserver/iseries/printing`

- iSeries Service and Knowledge Base

  `http://www.as400service.ibm.com`

- Online books

  http://as400bks.rochester.ibm.com/

- iSeries Information Center

  http://publib.boulder.ibm.com/html/as400/infocenter.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM @server iSeries Printing VII

Infoprint Server Implementation

Redpaper

High-return e-business output applications using Infoprint Server for iSeries

Implementation guidance for basic and advanced intelligent routing

Application techniques including color, fax, PDF encryption

This IBM Redpaper builds on the extensive coverage of the IBM Redbook *IBM @server iSeries Printing VI: Delivering the Output of e-business*, SG24-6250. It expands the implementation guidance for output delivery applications and includes new support available in V5R3. The target audience of this paper is iSeries Clients, IBM Business Partners, and IBM IT specialists who need to implement e-business communications solutions with Infoprint Server.

The principal focus of the paper is intelligent routing. Intelligent routing is a set of capabilities that enable the profiled delivery of business documents and reports to customers and users in the most productive format (print, Web, e-mail, fax). The intelligent routing functions are delivered with two different enabling interfaces. You'll find step-by-step coverage of both interfaces as well as sample programming to get you up and running quickly.

In addition, you'll find related tips and application techniques. These include use of color in iSeries documents, interfacing intelligent routing with fax functions, PDF encryption, and performance considerations.

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