RICOH

RICOH InfoPrint Manager™ for Linux

Procedures

Version 4.15.1

Administrative Procedures: Configuring InfoPrint Manage 1

Administrative Procedures: Configuring for Host Printing 2

Administrative Procedures: Customizing for Special Jobs 3

Operator and User Procedures 4

Reference Information 5

For information not in this manual, refer to the Help System in your product.

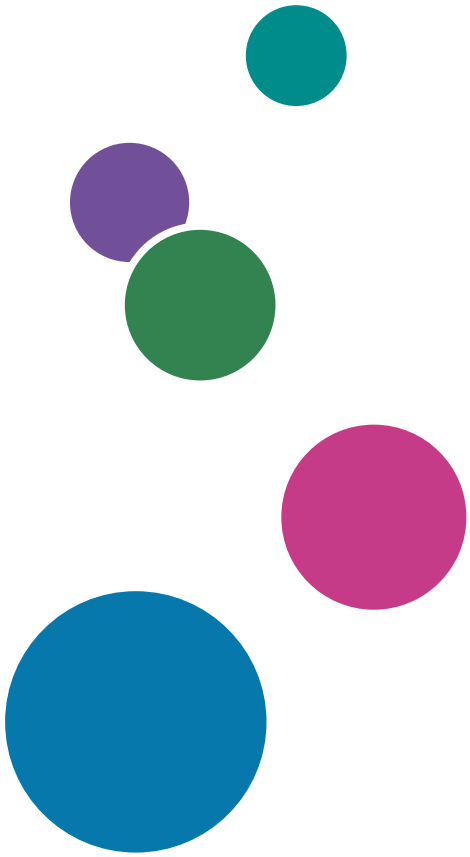


TABLE OF CONTENTS

Introduction 8
Important 8
Cautions regarding this guide 8
Guides for this application 8
How to read the documentation 9
Abbreviations 10
Trademarks 12

1 Administrative Procedures: Configuring InfoPrint Manager

Using Multiple Network Adapter Cards 15
Creating and managing servers 16
Preparing to create additional servers in a distributed environment 16
Creating additional servers or starting existing servers 17
Configuring servers with a mixed language environment 17
Shutting down a server 18
Installing and configuring printer devices and related hardware 18
Configuring InfoPrint Manager with the InfoPrint 5000 to print AFP 19
Configuring InfoPrint Manager to use the InfoPrint 5000 or RICOH Pro VC series models to print PostScript and PDF 24
Configuring InfoPrint Manager to use the InfoPrint 4000 and 4100 26
Specifying a halftone for printing on the InfoPrint Manager server 28
Working with SNMP printers 29
SNMP support 30
About SNMP communication 30
Setting up and using SNMP 34
Effects on performance with SNMP 37
Managed IPDS Dialog support 38
Enabling MID support 38
Considerations to activate MID support 39
Configuring media 39
Determining the media that are in the servers 39
Creating a media object 40
Viewing or changing attributes of a medium 40
InfoPrint Email support 41
Configuring InfoPrint Email 41
Submitting e-mail jobs 47
Color Email 51

Using CUPS DSS with SELinux set in enforcing mode 52
Using CUPS DSS with SELinux set in enforcing mode 52
User-exit programs (PSF only) 52
What type of user-exits does InfoPrint Manager support? 53
Creating and using your own user-exit programs 55
Compiling and installing the user-exit program 55
User-exit program structures 58
Common input and output fields 59
The header page and trailer page user-exit programs 63
The separator page user-exit program 63
The accounting, post-print accounting and audit user-exit programs 64
The input data user-exit program 67
The output data user-exit program 70
User-exit programs for the line-data transform 71
Attributes of the line-data input file 79
Creating and managing auxiliary sheets and user-exit programs 80
Preparing to work with auxiliary sheets 80
Creating and configuring auxiliary-sheet objects (PSF only) 89
Using Interrupt Message Pages 91
Using Interrupt Message Report 93
Allowing the auxiliary-sheet-selection job attribute with the BSD DSS 93
Gathering accounting and auditing data about print jobs 94
Working with InfoPrint Manager server accounting information 95
Working with PSF accounting, post-print accounting, and auditing user-exit programs 103
Creating and managing resource-context objects 108
The search order for AFP resources 109
How InfoPrint Manager provides resource access for PSF DSS printers 115
Creating a new resource-context object 116
Changing a directory path for a resource-context object 116
Assigning resource-context objects to an actual destination 117
Assigning a resource-context object to a default-document object 117
Using InfoPrint Manager notifications 118
Understanding notification profiles 118
Default notification profiles 121
Changing notification profiles 124
Getting help for notification messages 128

Other notification methods....128
Managing security in InfoPrint Manager for Linux....130
Managing Access Control Lists security for InfoPrint Manager for Linux....130
Managing LDAP security for InfoPrint Manager for Linux....134
Managing Federated Authentication for InfoPrint Manager for Linux....137
Managing Transport Layer Security encryption for InfoPrint Manager for Linux....140
Managing job encryption for InfoPrint Manager for Linux....152
Managing encryption in transit and at rest in InfoPrint Manager for Linux....155
Enabling Secure Sockets Layer (SSL) in the Web Server for HTTPS protocol....156
Configuring Tomcat to redirect all HTTP requests to HTTPS....158
Customizing error logs in the InfoPrint Manager server....159
Customizing an InfoPrint Manager server error log....159
Customizing a Notification daemon error log....160
Setting InfoPrint Manager environment variables....161
Preparing to work with InfoPrint Manager environment variables....161
Querying environment variables....163
Editing the /etc/profile.d/ipm_environment.sh file....164
Setting an environment variable from the command line....165
Verifying the setting of environment variables....165
Monitoring memory and space on your InfoPrint Manager server....165
Monitoring memory use on your InfoPrint Manager server....166
Configuring Tomcat memory allocation....167
Monitoring file system use (automatically)....167
Monitoring file system use (manually)....169
Using InfoPrint Manager with SELinux set in enforcing mode....170
Running InfoPrint Manager with SELinux set in enforcing mode....170
Reverting the SELinux changes for InfoPrint Manager....171
Changing the server host name and pdserver name....171
Using the InfoPrint Manager System Migration Utility (ISMU)....172
Prerequisites....173
Migrating your InfoPrint Manager settings....173
Configuring InfoPrint systems for interoperability....177
Understanding interoperating environments....177
Configuring a primary Linux server with a secondary AIX server....180
Configuring a primary Linux server with a secondary Linux server....180
Configuring a primary Linux server with a secondary Windows server....181

Using the LPD Gateway with InfoPrint Manager for Linux....185
Submitting print jobs through the LPD from an AIX system....186
Submitting print jobs through the LPD from Linux, and other UNIX systems....187
Submitting print jobs through the LPD from a Windows system....187
Using the Internet Printing Protocol (IPP) with InfoPrint Manager for Linux....187
Printing through the IPP Gateway....188
Using the Virtual Printer Daemon (VPD) Gateway with InfoPrint Manager for Linux....188
Submitting print jobs....189
Using AFP Upload....189
Software prerequisites....189
General information about the host server program....190
Transmitting AFP file attributes to MVS systems....193
Using InfoPrint Manager commands with AFP Upload....196
When the connection to the MVS system host fails....196
Troubleshooting problems with AFP Upload on TCP/IP networks....196
Specifying JCL Statements to run the host server program in a startable address space....199
Specifying JCL Statements to run the host server program as a job....199
Example JCL Statements to run the host server program in a Startable Address Space....199
Example JCL Statements to run the host server program as a job....200
Using the SystemD services....200
2 Administrative Procedures: Configuring for Host Printing
Printing data sets from an MVS system through MVS Download Receiver....203
Preparing to work with Download for z/OS....203
Modifying the InfoPrint mvsprpsm.sh shell script....206
Passing the parameter list from the mvsprsd daemon to the shell script....209
Translating print options to InfoPrint attributes and values....211
Using the mvsprsd command to start the daemon....214
Using Management Interface to create an MVS Download Receiver for a default configuration....215
Using Download for z/OS support for multiple data set jobs....216
Resubmitting or deleting MVS Download Receiver orphaned files....220
3 Administrative Procedures: Customizing for Special Jobs
Working with transforms....223
Customizing the PCL, PostScript, and PDF transforms....224
Stapling and punch options available from the ps2afp transform when using the CPSI transform engine....237

Collate PostScript options available from the ps2afp transform when using the CPSI transform engine 244
Customizing the TIFF, JPEG, and GIF transforms 244
Customizing the img2afp transform 246
Working with the transform for line data 248
Using the PDF Parallel RIP 253
Working with the XML transforms 260
Color management resource transform support 269
Understanding the transform objects and the custom steps (transforms) subsystem 270
Configuring custom steps (transforms) 271
Examples of creating custom steps (transform) objects 276
Defining a custom step (transform) sequence 285
Using print rules 286
Configuring print rules 286
Running a script in print rules 290
Defining a print rule sequence 292
Printing PCL or PostScript to PSF destinations when using the CPSI transform engine: Specifying which paper bin to use 292
Determining bin mappings 292
Editing configuration files 298
Creating a transform with bin mappings 298
Submitting PSF printer input to a PCL or PPDS printer 300
Setting up PCL tray mappings 300
Color and Grayscale Printing 302
AFP color and grayscale solutions 302
Color Printing Concepts 304
Grayscale Printing Concepts 308
Color Management 309
AFP Color Management 311
RICOH AFP color and grayscale products 321
AFP color solution scenarios 325
Related publications 331
Working with fonts when using the CPSI processing engine 332
Fonts for printing transformed PostScript and PDF data 332
Using OpenType Fonts 335
Fonts for printing ditroff data 340
Fonts for printing DBCS ASCII and EUC data 350

Fonts for printing line data 351
Printing with the DBCS Simulation Fonts with the font metric adjustment triplet....353
Installing the Japanese PostScript font feature 353
Using global resource identifiers 353
How can you use GRIDs?....353
What GRID files ship with InfoPrint?....354
Where does InfoPrint search for GRID files? 356
Understanding the general syntax rules and allowable values for GRID files....356
Modifying GRID files 358
Preventing InfoPrint from using GRID files....361
Configuring and using cut-sheet emulation....361
Using cut sheet emulation 362
Generating and submitting color mapping table source and output files 363
Parts of a color mapping table....364
Creating a color mapping table 365
Submitting jobs using a color mapping table 368
Setting up high speed printers for spacing....369
Enabling SNMP on an InfoPrint 4000 printer that uses the operator console....370
Enabling SNMP on a printer that uses the Enhanced Operator Console....370
Using InfoPrint Manager with the InfoPrint 5000 to print AFP....371
Setting up 371
Using and changing color and resource attributes....372
Sending an AFP job to the printer using the Linux command line....376
Sending an AFP job with color images to the printer using InfoPrint Submit Express 377
Sending a job using Download for z/OS....379
Using the InfoPrint 5000 and RICOH Pro VC series models with hot folders from InfoPrint Manager to print PostScript and PDF 379
Transferring InfoPrint Manager objects across servers....380
Transferring an actual destination to a different server (Web GUI) 381
Transferring a queue to a different server (Web GUI)....381

4 Operator and User Procedures

Finishing options 383
Setting up finishing options for IPDS-only printers 383
Setting up finishing options for PS/PCL printers....386
Other procedures 389
Adding an Internet Printing Protocol (IPP) gateway printer to your desktop....389

Submitting a job to a DocuTech Model 135 printer....390
Supporting status Information....391
Stopping, restarting, or pausing a job that is currently printing....391
Resubmitting a retained job for reprint....395
Pausing and restarting an actual destination....396
Interrupting and restarting a job that is currently printing on an InfoPrint 4000/4100....396
Moving backward or forward in a job that is currently printing....398
Stopping a printer to carry out scheduled maintenance....400
Fixing a printer problem and starting to print again....400
Using the correct halftones for the installed toner version....405

5 Reference Information

IPDS print operator commands....407
IPDS error recovery....409
IPDS error recovery: Data stream errors....409
IPDS error recovery: Insufficient memory in the printer....412
IPDS error recovery: Intervention required conditions....412
IPDS error recovery: Unrecoverable problems....413

Accessibility

Glossary

Introduction

Important

To the maximum extent permitted by applicable laws, in no event will the manufacturer be liable for any damages whatsoever arising out of failures of this product, losses of documents or data, or the use or non-use of this product and operation manuals provided with it.

Make sure that you always copy or have backups of important documents or data. Documents or data might be erased due to your operational errors or malfunctions of the software. Also, you are responsible for taking protective measures against computer viruses, worms, and other harmful software.

In no event will the manufacturer be responsible for any documents created by you using this product or any results from the data executed by you.

Cautions regarding this guide

- Some illustrations or explanations in this guide could differ from your product due to improvement or change in the product.
- The contents of this document are subject to change without notice.
- No part of this document may be duplicated, replicated, reproduced in any form, modified, or quoted without prior consent of the supplier.
- Throughout this publication, references to directory paths indicate the default paths only. If you install RICOH InfoPrint Manager™ or any of its components in a different location, including a different drive, you must adjust the paths accordingly.
  For example, if you install InfoPrint Manager on the `D:` drive of a computer running a Windows® operating system, replace `C:` with `D:` in the directory paths.

Guides for this application

This publication provides information about RICOH InfoPrint Manager™ for AIX, Linux, and Windows, Version 4.15 (Program Number 5648-F40).

This publication includes an overview of InfoPrint Manager and installation and configuration information about the product.

Instruction manuals

These instruction manuals are included:

For information about InfoPrint Manager, see these documents:

- *RICOH InfoPrint Manager for Windows: Planning Guide*, G550-1071
- *RICOH InfoPrint Manager for Windows: Getting Started*, G550-1072
- *RICOH InfoPrint Manager for Windows: Procedures*, G550-1073
- *RICOH InfoPrint Manager for Linux: Planning Guide*, G550-20262
- *RICOH InfoPrint Manager for Linux: Getting Started*, G550-20263
- *RICOH InfoPrint Manager for Linux: Procedures*, G550-20264
- *RICOH InfoPrint Manager for AIX and Linux: Configuring and Tuning Guide*, S550-1062

- *RICOH InfoPrint Manager for AIX: Planning Guide*, G550-1060
- *RICOH InfoPrint Manager for AIX: Getting Started*, G550-1061
- *RICOH InfoPrint Manager for AIX: Procedures*, G550-1066
- *RICOH InfoPrint Manager: High Availability Guidelines*, G550-20261
- *RICOH InfoPrint Manager: Reference*, S550-1052
- *RICOH InfoPrint Manager: PSF, Server, and Transform Messages*, G550-1053
- *RICOH InfoPrint Manager: Secure Print: Installing and Configuring*, G550-20129
- *RICOH InfoPrint Manager: SAP Planning and Configuring Guide*, S550-1051
- *RICOH InfoPrint Manager: Dictionary of Keywords*, S550-1188
- *AFP Conversion and Indexing Facility: User's Guide*, G550-1342
- *Page Printer Formatting Aid for Windows: User's Guide and Reference*, S550-0801
- *RICOH InfoPrint Manager AFP2PDF Transform Feature: Installing and Using*, G550-1057
- *RICOH InfoPrint Manager: Installing InfoPrint Manager Transform Feature*, G550-20160

Help

Property help is available on many screens to provide information for specific tasks and settings.

In addition, the **Help** menu provides access to the HTML version of the instruction manual directly from the user interface.

Note

- A PDF reader, such as Adobe® Acrobat® Reader®, must be installed to view the PDF documentation.

For more information about RICOH printing products, see:

RICOH Commercial and Industrial Printing website at https://www.ricoh-usa.com/en/products/commercial-industrial-printing.

RICOH Software Information Center at https://help.ricohsoftware.com/swinfocenter.

How to read the documentation

Displaying the instruction manual

Use this procedure to view the instruction manuals.

Displaying the instruction manual in PDF format

- InfoPrint Manager includes publications in PDF format on the DVD-ROM supplied with the product.

Displaying the instruction manual in HTML format

- The HTML version of the instruction manual is available directly from the user interface. Start the application, and then click button at the right of the banner and select **Help**.

Displaying property help

Click the ? button next to a property on the user interface to display the property help for that item.

Symbols

The following symbols are used in this manual to help you to identify content quickly.

Important

- This symbol indicates points to pay attention to when using the product. Be sure to read these explanations.

Note

- This symbol indicates helpful supplementary information that is not essential to completing a task.

Bold

Bold type indicates the names of dialogs, menus, menu items, settings, field labels, buttons, and keys.

Italic

Italic type indicates the titles of manuals and variables that you must replace with your own information.

`Monospace`

`Monospace type` indicates computer input and output.

Abbreviations

ACIF

AFP Conversion and Indexing Facility

AFP

Advanced Function Presentation

AIX®

Advanced Interactive Executive

ANSI®

American National Standards Institute

ASCII

American Standard Code for Information Interchange

BCOCA

Bar Code Object Content Architecture

CCSID

coded character set identifier

CMR

color management resource

EBCDIC

Extended Binary Coded Decimal Interchange Code

FOCA

Font Object Content Architecture

GIF

Graphical Interchange Format

GOCA

Graphics Object Content Architecture

IOCA

Image Object Content Architecture

IP

Internet Protocol

IPDS

Intelligent Printer Data Stream

JCL

job control language

JES

Job Entry Subsystem

JPEG

Joint Photographic Experts Group

MO:DCA

Mixed Object Document Content Architecture

MO:DCA-P

Mixed Object Document Content Architecture for Presentation

MVS™

Multiple Virtual Storage

PCL

Printer Command Language

PDF

Portable Document Format

PTOCA

Presentation Text Object Content Architecture

RAT

resource access table

TIFF

Tagged Image File Format

XML

Extensible Markup Language

Trademarks

RICOH InfoPrint Manager™ and RICOH ProcessDirector™ are trademarks of Ricoh Company, Ltd. in the United States, other countries, or both.

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

- AFP Font Collection for S/390®
- AIX®
- IBM®
- MVS™
- OS/390®
- POWER®
- PowerHA®
- PowerVM®
- S/390®
- Semeru Runtime®
- Tivoli®
- z/OS®

Adobe®, Adobe® Illustrator®, Adobe® PDF, Adobe® PDF Print Engine, Adobe® RGB (1998), Acrobat® Reader®, and PostScript® are registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

AlmaLinux OS™ is a trademark of AlmaLinux OS Foundation.

ANSI® and numerous other identifiers containing "ANSI" are registered trademarks, service marks, and accreditation marks of the American National Standards Institute (ANSI).

Apache® and Tomcat® are registered trademarks of the Apache Software Foundation in the United States and/or other countries.

Artifex® and Ghostscript® are registered trademarks of Artifex Software, Inc.

Citrix® and Citrix Virtual Apps and Desktops™ are either registered trademarks or trademarks of Cloud Software Group, Inc., and/or its subsidiaries in the United States and/or other countries.

CUPS™ and TrueType® are trademarks or registered trademarks of Apple, Inc. in the United States, other countries, or both.

Docker® is a registered trademark of Docker, Inc. in the United States and/or other countries.

eDirectory®, NetIQ®, and Novell® are registered trademarks or trademarks of Micro Focus International plc and/or any of its group companies or affiliates in the United Kingdom, United States and other countries.

Epson® is a registered trademark of Seiko Epson Corporation in the U.S. and other countries.

Fiery® is a registered trademark of Fiery, LLC in the U.S. and/or certain other countries.

GNOME® and GTK® are registered trademarks of the GNOME Foundation.

HP® is a registered trademark of HP Inc.

Intel® is a registered trademark of Intel Corporation in the United States and/or other countries.

Keycloak™ is a trademark of The Linux Foundation® in the United States and/or other countries.

Kodak® is a registered trademark of Eastman Kodak Company.

Lexmark® is the registered trademark of Lexmark International, Inc. in the United States and/or other countries.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

OpenSSL® is a registered trademark owned by the OpenSSL Software Foundation.

Oracle®, Java®, and OpenJDK® are registered trademarks of Oracle and/or its affiliates.

Microsoft®, Microsoft Edge®, Windows®, Windows Server®, Active Directory®, Hyper-V®, OpenType®, Visual C++®, and Visual Studio® are trademarks of Microsoft Corporation in the United States, other countries, or both.

Mozilla® and Firefox® are registered trademarks of the Mozilla Foundation in the U.S. and other countries.

Okta® is a registered trademark of Okta, Inc. in the U.S. and other countries.

Red Hat®, RHEL®, and Red Hat Enterprise Linux® are registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Rocky® Linux® is the registered trademark of the Rocky Enterprise Software Foundation in the U.S. and other countries.

Samba® is a registered trademark of Software Freedom Conservancy, Inc.

SAP®, SAP S/4HANA®, SAP® R/3®, SAP® NetWeaver®, and ABAP® are the registered trademarks of SAP SE or its affiliates in Germany and in several other countries.

SUSE®, openSUSE®, and SUSE Linux Enterprise Server® are registered trademarks of SUSE LLC or its subsidiaries or affiliates.

Sentinel® is a registered trademark of Thales DIS CPL USA, Inc.

Unicode® is a registered trademark of Unicode, Inc. in the United States and other countries.

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

VMware® is a registered trademark of VMware, Inc. in the United States and other jurisdictions.

Xerox® is a registered trademark of Xerox Corporation in the United States and/or other countries.

Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights to those marks.

1. Administrative Procedures: Configuring InfoPrint Manager

- **Using Multiple Network Adapter Cards**
- **Creating and managing servers**
- **Installing and configuring printer devices and related hardware**
- **Working with SNMP printers**
- **Managed IPDS Dialog support**
- **Configuring media**
- **InfoPrint Email support**
- **Using CUPS DSS with SELinux set in enforcing mode**
- **User-exit programs (PSF only)**
- **Creating and managing auxiliary sheets and user-exit programs**
- **Gathering accounting and auditing data about print jobs**
- **Creating and managing resource-context objects**
- **Using InfoPrint Manager notifications**
- **Managing security in InfoPrint Manager for Linux**
- **Customizing error logs in the InfoPrint Manager server**
- **Setting InfoPrint Manager environment variables**
- **Monitoring memory and space on your InfoPrint Manager server**
- **Using InfoPrint Manager with SELinux set in enforcing mode**
- **Changing the server host name and pdserver name**
- **Using the InfoPrint Manager System Migration Utility (ISMU)**
- **Configuring InfoPrint systems for interoperability**
- **Using the LPD Gateway with InfoPrint Manager for Linux**
- **Using the Internet Printing Protocol (IPP) with InfoPrint Manager for Linux**
- **Using the Virtual Printer Daemon (VPD) Gateway with InfoPrint Manager for Linux**
- **Using AFP Upload**
- **Using the SystemD services**

Using Multiple Network Adapter Cards

If you are concerned that your local area network (LAN) might not have enough bandwidth to handle the additional traffic that installing additional printers can create, you can create a secondary network that is essentially dedicated to driving printers. To set up this configuration, the system that your InfoPrint Manager server is installed on will have to have two network adapter cards installed in it, one for each network. Print data comes into the InfoPrint Manager server over the LAN and is transferred to the secondary network to be printed.

When you set up the secondary network, keep these points in mind:

- In the system that your InfoPrint Manager server is installed on, the primary network adapter card should be the one that connects to your existing network.
- In the system that your InfoPrint Manager server is installed on, the secondary network adapter card connects to a network that contains **only** printers. This network is isolated; no IP forwarding is recommended.
- The IP addresses for printers on the secondary network must not be resolvable on the primary network.

  It is recommended that you use private IP addresses for your secondary network card and your printers, for example, addresses that follow either of these formats:

  – Class A IP address: 10.xx.xxx.xxx

  - Class B IP address: 192.168.xxx.xxx
- In the properties for the secondary network adapter card, you should not set a Default Gateway, DNS, or WINS. If these fields are filled in when you configure the card, clear the default values.
- On the printers that you connect to the secondary network, do not specify a Default Gateway. If you must specify a Default Gateway, enter the IP address of the secondary network adapter card.

Creating and managing servers

The InfoPrint server manages and controls all other InfoPrint objects in your InfoPrint system. During InfoPrint installation, one server was created that contains a logical destination, a queue, and an actual destination (a physical printer). Use the information in this chapter to add additional servers and to perform server configuration and maintenance tasks.

Preparing to create additional servers in a distributed environment

If you are configuring for a distributed environment and need to define many instances of the InfoPrint server process on the same machine, use the **fstshow** command to see what intercommunication ports are available on your server.

For example, entering `fstshow -o` on a command line produces a display similar to the contents of this table:

```
#########################################################
#   IN-USE?    PORT#    PID     DIR-ID     NAME
#     Y        2050    23266      2       server1
#     Y        2051    24272      0       notifyd
#     N        2052               99
#     N        2053               99
#     N        2054               99
#     N        2055               99
#     N        2056               99
#     N        2057               99
#     N        2058               99
#     N        2059               99
#########################################################
```

Once you have a list of the server intercommunication ports that are either in use or available (`2050` through `2059` in the display above), you can see if you have enough ports to start more InfoPrint Manager server processes. If not, you can use the **pdinitports** utility to initialize a larger range of port numbers for interprocess communication. Note that InfoPrint Manager processes, such as **notifyd** and the SAP callback daemons also use server intercommunication ports.

Because InfoPrint defaults to reserving only 10 ports, to define more than 10 server processes, you must use the **pdinitports** utility to initialize more than 10 ports. For example, to initialize InfoPrint to use the server interprocess communication ports 2050 through 2069 for twenty server processes, enter this at the command line:

```
pdinitports 2050 2069
```

For a complete description of the **pdinitports** utility, see *RICOH InfoPrint Manager: Reference*. Once you have completed this process, you can begin Creating additional servers or starting existing servers, p. 17.

Creating additional servers or starting existing servers

You can create a new server or start an existing server through the InfoPrint Manager Management Interface (IPMMI). If you are not familiar with how to start the IPMMI, see "Using the InfoPrint Interfaces" in *RICOH InfoPrint Manager for Linux: Getting Started*.

To create a new server:

1. To start the InfoPrint Manager Management Interface (IPMMI) , go to **Applications** → **InfoPrint Manager** → **Management Interface** on Red Hat Enterprise Linux (RHEL) or **Computer** → **More Applications** → **InfoPrint Manager** → **Management Interface** on SUSE Linux Enterprise Server (SLES).
2. From the InfoPrint Manager Management Interface, on the left pane right-click on **Servers**. Select **New Server** → **Menu**. The **Create Server** dialog window opens.
3. Enter the name of the InfoPrint server in the **Name** field.
4. To use a port number other than the default of **6874** as the communications port for InfoPrint, enter the port number in the **PORT number** field.

   When selecting your own port number, always increment the choice by two because the InfoPrint server uses two numbers: the one you specify and the next one, for example, 6874 and 6875.
5. Choose one the start mechanism from **Options**. You can configure the new server to start **now** or **now and at reboot**.
6. If you want the InfoPrint server to run in a locale other than your current locale, enter the locale name in the **LOCALE** field. Possible values: en_US (default), fr_FR, de_DE, it_IT, es_ES or ja_JP.
7. Press **OK**. The result of this step depends on the **Options** selected.

To start an existing server:

1. From the `InfoPrint Manager Management Interface`, go to the **Servers** panel and select the server instance you want to start.
2. Right-click on the server and select **Start**.
3. Enter the name of the InfoPrint server in the * **SERVER name** field. By default, InfoPrint fills in this field with the name of the Linux system on which you are running the IPMMI session.

Note

1. Normally you will only run one InfoPrint server on an Linux system.
2. You also use this procedure to start a server that you have shut down.

Configuring servers with a mixed language environment

If your InfoPrint environment is a mixed language environment with separate locales, you must use Creating additional servers or starting existing servers, p. 17 to create separate servers running in each language of your environment. Each of these servers can be named for the locale they support (such as `English`, `French`, `Italian`, `German`, `Spanish`, or `Japanese`).

These InfoPrint Manager servers do not need to contain any InfoPrint Manager objects, such as queues or destinations. Instead, these servers handle the locale-specific processing of the commands and forward those commands to the InfoPrint Manager server that contains the objects.

For each InfoPrint client that is not running on the same Linux server as the InfoPrint Linux servers, such as the InfoPrint Manager Operations GUI, InfoPrint Submit Express or InfoPrint Select, you must make sure that it connects to the correct InfoPrint server by setting the **PD_SOCKET** environment variable to contain the port number of the InfoPrint Linux server supporting the locale of your client. If your client is running in a locale not supported by the InfoPrint server then connect it to an InfoPrint server running in English. To do this task, see Setting InfoPrint Manager and Linux environment variables, p. 161.

Shutting down a server

You can shut down a server immediately or after it finishes processing some or all of its current jobs.

InfoPrint Manager prevents the server from accepting new jobs while the server is shutting down and after it has shut down. Logical destinations and actual destinations in the server that were enabled at shut down time are enabled when the server is restarted. Logical destinations and actual destinations in the server that were disabled at shut down time are disabled when the server is restarted.

Note

See Creating additional servers or starting existing servers, p. 17 for the restart procedure.

You can shut down an existing server through the InfoPrint Manager Management Interface. If you are not familiar with how to start the InfoPrint Manager Management Interface, see "Using the InfoPrint Interfaces" in *RICOH InfoPrint Manager for Linux: Getting Started*.

To shut down an existing server:

1. From `InfoPrint Manager Management Interface` go to the **Servers**panel and select the server instance you want to stop.
2. Right-click on the server and select **Stop**.
3. Chose **When to stop the server: now, after current** or **after all**.
4. Press **Enter**. The result of this step depends on whether the server is running.

Note

1. Normally you will only run one InfoPrint Manager server server on an Linux system.
2. You also use this procedure to start a server that you have shut down.

Installing and configuring printer devices and related hardware

For information about printers not included in this chapter, see the Ricoh website at https://www.ricoh-usa.com/en/products/commercial-industrial-printing.

To create and configure the InfoPrint actual destinations that represent printer devices, see “Creating and managing actual destinations” in *RICOH InfoPrint Manager for Linux: Getting Started*.

Configuring InfoPrint Manager with the InfoPrint 5000 to print AFP

The InfoPrint 5000 is a high-speed, high quality, full-color digital printer that you can submit jobs to through InfoPrint Manager. The InfoPrint 5000 system processes and prints high quality color AFP output at high speeds. It also processes and prints PDF or PostScript documents that include high resolution color images.

The InfoPrint 5000 combines job management and page-level error recovery capabilities with the ability to print full-color applications and documents. This printer can support applications that use data formats, such as Encapsulated PostScript (EPS), and Portable document format (PDF), as resources for variable data AFP documents. The AFP architecture lets you use the Rasterized Image Processor (RIP) to process these resources in the printer controller and save them in the printer. This capability improves performance by eliminating the need to re-RIP these resources. It is not necessary to convert single page PDF and EPS, objects to color images before sending them to the printer. These objects can be RIPped directly in the printer controller, increasing performance.

Significant processing demands are placed on the InfoPrint Manager system when it is preparing and sending print jobs to an InfoPrint 5000. Therefore, you should dedicate one Linux server entirely to your InfoPrint 5000; this system should only contain one InfoPrint Manager server, and it should only send print jobs to one InfoPrint 5000.

Defining the InfoPrint 5000 to InfoPrint Manager

Important

Because you are dedicating an InfoPrint Manager server to your InfoPrint 5000, this procedure assumes that you have just installed InfoPrint Manager on this system. Your InfoPrint Manager server should not contain any objects (logical destinations, queues, or actual destinations) yet. If you are planning to use an existing InfoPrint Manager server to send jobs to your InfoPrint 5000, you should delete all of the objects that currently exist in it before you start this procedure.

When you define your InfoPrint 5000 to InfoPrint Manager to print AFP data, you must define it as a **PSF TCP/IP-attached** actual destination. The actual destination represents the physical output device. The InfoPrint 5000 must be a PSF TCP/IP printer. Use the **Create Printer Wizard** in the InfoPrint Manager Administration GUI to create an actual destination.

Follow this procedure to define the printer using the InfoPrint Manager Administration GUI.

Note

You create and configure the logical destination when you create the actual destination with the Create Printer wizard.

1. Start the InfoPrint Manager Administration GUI.
2. Select **Printer** → **Create** → **PSF** → **TCP/IP**.
3. In the first panel of the Create Printer Wizard, do these steps.
    1) Type a name for the actual destination in the **Name** field.
    2) Verify that the server is correct in the **Server** field.
    3) Select **InfoPrint 5000** from the **Model** drop-down list.

Note

If that model is not shown, you must update InfoPrint Manager to the current service level.

4) Type the TCP/IP address and port number for the printer in the correct fields.
5) Click **Next**.

4. In the second panel, type a name for the logical destination that will send jobs to this actual destination and click **Next**.
5. In the third panel, type a name for the queue that will submit jobs to this actual destination and click **Next**.
6. On the rest of the panels, accept the default values that InfoPrint Manager sets for AFP color printing by clicking **Next** and finally **Finish**.

Note

See the worksheets found in "Creating and managing actual destinations" in *RICOH InfoPrint Manager for Linux: Getting Started* when providing the information as prompted by the wizard.

When you define your InfoPrint 5000 to InfoPrint Manager to print PostScript or PDF, you must define it as a PSF TCP/ IP-attached actual destination. The actual destination represents the physical output device. The InfoPrint 5000 must be a PSF TCP/IP printer. Use the Create PrinterWizard in the InfoPrint Manager Administration GUI to create an actual destination.

Configuring the PSF TCP/IP printer in InfoPrint Manager

Once you have created the actual destination in InfoPrint Manager, you must do several other things before you can print. Complete the following procedures in any order. Instructions appear below.

- Verifying the size and location of the Input Manager workspace, p. 20
- Identifying where to store AFP resources, p. 21
- Specifying the color rasterized image processor (RIP) on this server, p. 22
- Changing the color rendering value, p. 23

Once you have completed these steps, see Using InfoPrint Manager with the InfoPrint 5000 to print AFP, p. 371 for additional information about sending print jobs to the InfoPrint 5000.

Verifying the size and location of the Input Manager workspace

The Input Manager is the part of InfoPrint Manager that breaks print jobs into segments to allow greater flexibility during processing. The Input Manager stores job segments in its workspace. Segments remain in the workspace either until they are manually deleted or the workspace fills up, depending on how you choose to manage the workspace. By default, the Input Manager workspace is located at `/var/psf/segments`.

Because of the size and complexity of the color print jobs you send to your InfoPrint 5000, the Input Manager workspace must have at least 18 GB of disk space available and it should be a separate filesystem. If the existing `/var/psf/segments` does not have that much disk space available, the easiest thing to do is to create a new filesystem and mount it over `/var/psf/segments`. None of the InfoPrint Manager settings need to be changed.

Note

Make sure that the new filesystem is set to mount automatically at system reboot so that it will still be available if you have to reboot your system.

If you do not want to overmount the filesystem, you can create a new filesystem and mount it on a different mount point. In this case, you must change the Input Manager settings to point to the correct location. To change the Input Manager settings, do this.

1. At an **IPMMI go to Options → Customize Input Manager**
2. Change the **PATH used for work area & percent usage threshold** to point to the new filesystem you created.
3. Confirm the changes and exit the tool.

Identifying where to store AFP resources

There are various kinds of AFP resources, such as form definitions, page definitions, overlays, and fonts. The newest kind of AFP resource is the *data object*. Like other resources, data object resources are files that you can include in AFP documents. Data object resources can be EPS, PDF, and IOCA files. In addition, if you use the transforms that come with InfoPrint Manager to convert TIF, GIF, and JPEG images into AFP first, they can be used as data object resources as well. See Transforming color images, p. 371.

In InfoPrint Manager, data object resources are grouped with BCOCA and GOCA objects, and all of those types of files are called *presentation object containers*. As with other kinds of AFP resources, you must tell InfoPrint Manager where your presentation object containers are located.

In addition, you should tell InfoPrint Manager whether your presentation object containers are **private** or **public**. Private resources cannot be reused by other print jobs; public resources can be used by anyone in a variety of print jobs. For example, you might make signature files **private**, so that no one could use them inappropriately in correspondence or on checks. In addition, you might make your company's logo **public**, so you only need to store one copy of it for everyone to use. By default, all presentation object containers are considered private.

You can define where your image resources are stored and whether they are public or not either from the AIX command line using the **pdset** command with the **resource-context-presentation-object-container** and **public-context-presentation-object-container** attributes or using the InfoPrint Manager Administration GUI.

Follow this procedure to set the attributes using the InfoPrint Manager Administration GUI.

1. Start the InfoPrint Manager Administration GUI.
2. Select **Options** → **Customize**.
3. In the **Customize** dialog, click the **Printer** tab.
4. In the **Action** column, find **Change AFP Resources**, **Change Tuning**, and **Change AFP Color**, and select the check boxes in the **Menu** column next to it.
   This step adds this task to the **Printer** menu. If you want to add other tasks to the Printer menu or to any other menu, you can do it now or you can wait until later.
5. Click **OK** to apply the settings and close the dialog.
6. Select the printer that you created in Defining the InfoPrint 5000 to InfoPrint Manager, p. 19.
7. Select **Printer** → **AFP Resources** to open the **Change AFP Resources** dialog.

8. In the **Location of presentation object containers** field, type the path where your presentation object containers are stored. If they are stored in more than one location, list all of the paths separated with colons (:), for example:
   `/resources/containers:/dept123/pres-obj-containers:/dept456/pres-obj-containers`
9. Fill in any other fields that are appropriate. For example, if you use form definitions, fill in the **Location of form definitions** field.
10. Click **OK** to apply the settings and close the dialog.
11. If all of your presentation object containers are private, continue with step 13. If some or all of your presentation object containers are public, select **Printer** → **Change Tuning** to open the **Change Tuning** dialog.
12. In the **Public presentation object containers** field, type the path where your public resources are stored. If your resources are stored in more than one location, list them all here separated with colons (:), for example:
    `/resources/containers:/dept123/public-pres-obj-containers`
13. Click **OK** to apply the settings and close the dialog.

Specifying the color rasterized image processor (RIP) on this server

Important

You only need to do this step if you will be submitting color PostScript print jobs to your InfoPrint 5000.

The RIP generates the data stream that InfoPrint Manager sends to the printer. By default, InfoPrint Manager uses a RIP that produces monochrome (black and white) documents. Therefore, you must specify the color RIP from the **ps2afp** data stream transform before you send color documents to your InfoPrint 5000.

To make the necessary configuration changes and start the color RIP, use this procedure:

1. You first need to stop the ps2afpd daemon:
   1. Find the process id number: **`ps | grep ps2afp`**
   2. Stop the daemon: **`kill -9 <process id number>`**
2. Edit the **ps2afpd.cfg** daemon transform configuration file that is located in the **/usr/lpp/psf/ps2afp/** directory on the server using your editor of choice.
3. Change the **ps_output_type** value to `FS45`, as shown in this example:

```
ps_output_type = FS45  # -a          type of AFP image to generate
```

4. Save and close the file.
5. Start the daemon again:

```
/usr/lpp/psf/bin/ps2afpd -C /usr/lpp/psf/ps2afp/ps2afpd.cfg
```

If you ever want to send monochrome print jobs to your InfoPrint 5000, you can stop the color RIP and start the monochrome RIP by changing the **ps_output_type** value back to the default value of `IO1_G4`, as shown in this example:

```
ps_output_type = IO1_G4  # -a          type of AFP image to generate
```

Changing the color rendering value

While the InfoPrint 5000 produces vibrant full-color images, including images that contain gradients and shading, it is not able to reproduce every shade of every color. The *color rendering intent* value (**color-rendering-intent** attribute) determines what InfoPrint Manager does when it encounters a shade in a PostScript image that the InfoPrint 5000 cannot reproduce exactly.

There are two possible values for color rendering intent:

Relative

When InfoPrint Manager encounters a color that InfoPrint 5000 cannot reproduce, it substitutes the closest color value that it *can* reproduce.

Perceptual

When InfoPrint Manager encounters a color that InfoPrint 5000 cannot reproduce, it adjusts all of the colors in the image so that they maintain their color relationships to each other. The result is an image that is pleasing to the eye, but that might not look exactly like the original image.

The default value is **Relative**.

You can set the *color rendering intent* value on a print job, a default document, or on an actual destination using either the command line or the InfoPrint Manager Administration GUI. Follow this procedure to set the attribute using the InfoPrint Manager Administration GUI.

1. Start the InfoPrint Manager Administration GUI.
2. If you have not already customized the InfoPrint Manager Administration GUI to add the **Change AFP Color** task to the **Printer**, **Printer** → **Job and Document Defaults**, **Job**, and **Logical** → **Job and Document Defaults** menus, complete these steps:
    1. Select **Options** → **Customize**.
    2. In the **Customize** dialog, click the **Printer** tab.
    3. In the **Action** column, find **Change AFP Color** and **Job and Document Defaults Change AFP Color** and select the check box in the **Menu** column next to it.
    4. Repeat steps 2.1 through 2.3 about the **Job** and **Logical** tabs.
    5. Click **OK** to apply the settings and close the dialog.
3. Select the object that you want to set the *color rendering intent* value for by clicking it once.
4. Use the menu items you added in step 2 to open the appropriate **Change AFP Color** dialog:
    - For an actual destination, select **Printer** → **Change AFP Color**.
    - For a default job on an actual destination, select **Printer** → **Job and Document Defaults** → **Change AFP Color**.
    - For a default job on a logical destination, select **Logical** → **Job and Document Defaults** → **Change AFP Color**.
    - For a job that has already been submitted, select **Job** → **Change AFP Color**.
5. In the **Change AFP Color** dialog, find the **Color rendering intent** field, select a value from the drop-down list, and click **OK**.

There are three methods that you can use to change the color rendering value for your print job.

1. Set the value in the PostScript data stream when you create the print job.

2. Add a line to the configuration file for the InfoPrint Manager PostScript to AFP transform.
   1) On your InfoPrint Manager server, locate the `ps2afpd.cfg` file.
   2) Open the file in a text editor.
   3) Add this line to the file: `pragma colorRendering=perceptual`
   4) Save and close the file.

   All of the PostScript files that are sent through the **ps2afp** transform will now use perceptual color rendering.
3. Add a flag to the **ps2afp** command when you issue it. This option only applies if you are using the InfoPrint transforms independently of InfoPrint Manager; for example, to convert your files into AFP before you submit your print jobs so they print more quickly. To change the color rendering value, add this flag to the ps2afp command:

   –pragma colorRendering=perceptual

   For example, to convert a PostScript file called mybook.ps to AFP using perceptual color rendering, you would issue this command:

```
ps2afp -pragma colorRendering=perceptual mybook.ps
```

   See *RICOH InfoPrint Manager: Reference* for additional information about using the ps2afp transform.

Specifying the color rendering intent value on the other-transform-options attribute

The attribute **other-transform-options** can now have a value which includes an equals sign, so **pragma** statements are supported. For example, to submit a color print job called **PS.doc** with the pragma **colorRendering**, to a printer called **color_printer** using perceptual color rendering, you could type this on the command line:

```
pdpr -d color_printer -x "other-transform-options = '-pragma colorRendering=
perceptual'" PS.doc
```

The single quotes are a syntax requirement.

This pragma can also be used on a command line invocation of a transform or it can be specified in the transform configuration file.

Configuring InfoPrint Manager to use the InfoPrint 5000 or RICOH Pro VC series models to print PostScript and PDF

You can use hot folders to print PDF and PostScript files on the InfoPrint 5000 printer models (AD1/AD2, AD3/AD4, AD3/AD4-XR3, AS1, AS3, KM3, KM3/MD4, MD3/MD4) or RICOH Pro VC series models (RICOH Pro VC40000, RICOH Pro VC60000, RICOH Pro VC70000, and RICOH Pro VC80000) with InfoPrint Manager.

When you use hot folders, you must complete minor configuration tasks.

Verifying Samba is installed

To print to an InfoPrint 5000 printer model or RICOH Pro VC series model hot folder from InfoPrint Manager for Linux, you need the Samba for Linux to be installed. For more information on installation and configuration of Samba, go to www.samba.org.

Connecting to the InfoPrint 5000 or RICOH Pro VC hot folder

To connect to a printer hot folder from InfoPrint Manager for Linux, you need to issue a mount command.

To issue a mount command:

1. Log in as root.
2. Enter this command for models AS1 and AS3:

```
>mount -t cifs //printer_IPAdress/simplex/ /mnt -o username=Administrator
```

3. Enter this command for models AD1/AD2, AD3/AD4, AD3/AD4-XR3, KM3, KM3/MD4, MD3/MD4, RICOH Pro VC40000, RICOH Pro VC60000, RICOH Pro VC70000, and RICOH Pro VC80000:

```
mount -t cifs //printer_IPAdress/duplex/ /mnt -o username=Administrator
```

4. Press **Enter** when prompted to enter a password.

Note

If either InfoPrint Manager for Linux or the printer is shut down, you need to issue the mount command again.

Running scripts before printing PDF and PostScript files

All hot folders must be created from the InfoPrint 5000 printer operator panel. The hot folders can have different job attributes, such as the actual destination to print the job (**actual-destination-requested** job/default job attribute) or the combination of start, end, or separator sheets to use with a job (**auxiliary-sheet-selection** job/default job attribute).

To enable the InfoPrint 5000 hot folder support, run scripts on the InfoPrint Manager server to create a new BSD DSS printer (together with a new logical destination and a new or existing queue), to create the custom transforms that allows InfoPrint Manager to convert the input data to the corresponding output data format, and to assemble the list of transform objects previously created on the BSD DSS printer.

To configure the InfoPrint Manager server to enable hot folder support on the InfoPrint 5000 printer:

1. Log on to your InfoPrint Manager server.
2. Locate the scripts provided by InfoPrint Manager: `/usr/lpp/pd/bin`
3. Type this command: `run_customization_1.sh` *`nameOfServer nameOfAD nameOfQueue nameOfLD nameOfLocalFolder/yourhotfolder`*

   nameOfServer

Specify the name of the InfoPrint Manager server where to create the destination.

nameOfAD

Specify the name of the new actual destination that will be created.

nameOfQueue

Specify the name of the existing queue or a new queue.

nameOfLD

Specify the name of the new logical destination that will be created.

nameOfLocalFolder

Specify the name of the local mount point.

yourhotfolder

Specify the name of the hot folder created at the InfoPrint 5000 printer console.

4. Verify that a new actual destination and logical destination were created by checking the InfoPrint Manager Administration GUI.
5. Submit a PDF or PostScript test job to the InfoPrint 5000.

Example script

```
run_customization_1.sh sage ip5k ip5K-q ip5k-ld "/mnt/hmi_hf"
```

ipk5 is the local folder.

hmi_hf is the remote sub-folder

Configuring InfoPrint Manager to use the InfoPrint 4000 and 4100

Perform this procedure with the help of your InfoPrint service representative to make sure that the actual destination, the InfoPrint 4000 or InfoPrint 4100 printer, and the cutter-trimmer-stacker (CTS) (a post-processing device used for paper handling) all support the same media types.

1. Use the procedure in "Associating a medium with an actual destination" from the *RICOH InfoPrint Manager for AIX: Procedures* to specify the media that the actual destination representing the InfoPrint 4000 or 4100 supports.
2. On the printer, define forms that correspond to the media you specified in step 1, for example:

Medium Object Name	***Printer_Form Name***	***Printing_Form Description***
letter	na letter	2-Up letter (8.5 by 11)
legal	na legal	2-Up legal (8.5 by 14)
ledger	ledger	1-Up (11 by 17)

3. Set up the CTS for the dimensions required by the supported media. For example, to support ledger-size paper, set up the CTS for an 11-inch cut width and a 14-inch cut length. CTS can support a limited set of cut lengths and only one center slitting option.

Displaying the InfoPrint Manager Operations GUI from the Enhanced Operator Console

You can display the InfoPrint Manager Operations GUI from either the InfoPrint 4000 or InfoPrint 4100 Enhanced Operator Console by ordering and installing Feature Code 4560 (Part Number 54P1446) with your printer. For information about installing this feature on your printer, contact your service representative.

To configure this support from your InfoPrint Manager server, you must make sure that the remote execution daemon (**rexecd**) process is running on the server.

Also, you must update the **/etc/hosts** file to include the printer's IP address. To do this task, follow these steps::

1. Edit the **/etc/hosts** file with your Linux editor of choice.
2. Insert the printer's IP address in a line as shown below:

```
xx.xx.xx.xx printer # ethernet name/address
```

where *xx.xx.xx.xx* is the IP address of the printer control unit that is to display the InfoPrint Manager Operations GUI, and `printer` is a printer host name that is unique to your network.

Note

If you have multiple Enhanced Operator Consoles that can display the InfoPrint Manager Operations GUI, the **/etc/hosts** file must contain the IP address for each printer control unit.

Limitations of actual screen frequencies and angles on InfoPrint 4000 and 4100 printers

If you change the version of toner used with your printer, you should recalibrate all customized calibration curves to make sure that they print correctly. The screen frequencies shown in this section presume the use of a 600 dots per inch (DPI) printer. Frequencies at other DPI are proportionally lower. For example, an 85 lines per inch (LPI) screen used on a 300 DPI printer will be 43 LPI. Screen frequencies match the requested screen frequency as closely as possible, based on constraints of the halftone design.

To determine the actual screen frequencies and angles for both InfoPrint 4000 and InfoPrint 4100 printers, use the table below.

Actual screen frequencies for 600 DPI printers

Printer Models	Toner	Properties	Submit Screen Frequency			
			71 LPI	85 LPI	106 LPI	141 LPI
InfoPrint 4000 • IR3/IR4 • ID5/ID6 InfoPrint 4100 • HD3/HD4 • HS2	Version 3 or Version 7	Actual Screen Frequency (LPI)	74	85	96	117
		Actual Screen Angle (Default ° Set)	35.5°	45°	51.3 °	59°
		Actual Screen Angle (90° Set)	54.5°	45°	38.7°	31°
InfoPrint 4000 • IR3/IR4 • ID5/ID6 InfoPrint 4100 • HD1/HD2 • HD5/HD6 • PD1/PD2 • HS1 • HS3 • PS1	Version 2 (enhanced printing toner) or Standard Toner	Actual Screen Frequency (LPI)	71	85	103	141
		Actual Screen Angle	45°	45°	31°	45°
InfoPrint 4000 IR1/ IR2	Version 2, Version 3, or Standard Toner	Actual Screen Frequency (LPI)	71	83	106	141
		Actual Screen Angle	45°	34°	45°	45°
InfoPrint 4100 TD1-2 printer	Version 2	Actual Screen Frequency (LPI)	74	85	96	117
		Actual Screen Angle	36°	45°	51°	59°
InfoPrint 4100 • TS1 • TS2 • TS3 • TD3-4 • TD5-6	Version 7	Actual Screen Frequency (LPI)	74	85	96	117
		Actual Screen Angle	36°	45°	51°	59°

Specifying a halftone for printing on the InfoPrint Manager server

You can use a particular halftone for printing a grayscale file through either InfoPrint Submit Express interface, see Printing from InfoPrint Submit Express, p. 29 or command line, see Printing from the command line, p. 29.

Printing from InfoPrint Submit Express

You can print using a particular halftone for printing a grayscale file from a job ticket (*name*.jtk file) in InfoPrint Submit Express. Begin by selecting a value in the **Halftone** field of the **Look** tab. Note that all halftones shipped with the product have the term 'ibm' prepended to their identifier. If you are accessing this interface for the first time, all of your previous default calibrations will have 'ibm' attached to the name.

InfoPrint Manager lets you determine output appearance on a job-by-job basis by setting a default halftone in a default document or job object.

Note

InfoPrint Submit Express is distributed with the *Common Client* DVD-ROM.

Printing from the command line

You can submit a halftone calibration from the command line with the **pdpr** command:

```
pdpr -p sip20-sage -x halftone=ibm106lpi:highlight-midtones grayscale.ps
```

where the **-x** value is the customized calibration file and `grayscale.ps` is the file to be printed.

Note

When submitting a **pdpr** command from the command line, you can specify the **halftone** attribute. The actual halftone name for this example (`ibm106lpi:highlight-midtones`) can be found either by using the **InfoPrint Manager Administration Interface** to look up the halftones supported for the actual destination, or you can enter this at the command line:

```
pdls -cp -r halftones-supported ActualDestinationName
```

Using single cell halftones, available only for InfoPrint 4100 printer models, can improve the print out image quality, especially when printing small sized objects (images, line art or text).

You use the **single-cell-type** actual destination attribute to specify the single cell halftone for the printer. You specify either **elliptical** or **euclidian** as the value for **single-cell type**. You set the object threshold size with one of these document attributes: **image-small-threshold**, **line-small-threshold**, or **text-small threshold**.

You can submit a job requesting a single cell threshold for image, line art, or text from the command line by using one of these variations of the **pdpr** command:

```
pdpr -p sip20-sage -x image-small-threshold=0.5i grayscale.ps
```

```
pdpr -p sip20-sage -x line-small-threshold=0.5i grayscale.ps
```

```
pdpr -p sip20-sage -x text-small-threshold=72p grayscale.ps
```

For more details about the supported values for the **single-cell-type** actual destination attribute and the **image-small-threshold**, **line-small-threshold** and **text-small-threshold** document attributes, see *RICOH InfoPrint Manager: Reference*.

Working with SNMP printers

InfoPrint Manager supports Simple Network Management Protocol (SNMP) both version V1 and V3.

SNMP support

To use SNMP, you must set the **use-snmp** actual destination attribute to **true**.

Note

If the **destination-tcpip-internet-address** actual destination attribute is set, the **use-snmp** actual destination attribute is set to **true** by default.

About SNMP communication

For network printers that work with Simple Network Management Protocol (SNMP) and the standard printer management information block (MIB) described in RFC 1759, the InfoPrint Manager server can use SNMP to automatically:

- Set the default of several InfoPrint Manager actual destination attributes in the printer to the values returned in the standard printer MIB. The actual destination attributes pertain to trays, media, bins, plexes, and sides. This process is called *smart–defaulting* the attributes.
- Detect, report, and recover from printer problems for several conditions, even if no jobs are currently being printed.
- Assess device specific information directly.
- Execute device commands directly on the device.

InfoPrint Manager includes support for SNMP Version 3 (SNMPv3). This version enhances the security of SNMP communications between InfoPrint Manager and printers, and it includes encrypted and authenticated management traffic and protected remote configuration and device status messages.

SNMPv3 supports these levels of security:

- **Minimum**- Without authentication and privacy.
- **Medium**- With authentication, but without privacy.
- **Maximum**- With authentication and privacy.

To support these security levels, new SNMPv3 actual destination attributes are added. See Procedures, p. 35 for more information.

Smart-defaulting actual destination attributes

The InfoPrint Manager server can automatically smart-default an actual destination attribute for an SNMP printer **only** if an InfoPrint Manager user does one of these:

- Does **not** explicitly set the attribute when creating or modifying the actual destination
- Resets the attribute to the InfoPrint Manager default

Before you create the actual destination and let the server smart-default one or more actual destination attributes for a printer, we recommend that you:

1. Turn the printer on.
2. Make sure the printer is connected to the network.

The actual destination attributes that the server can smart-default are:

- **destination-model**
- **input-trays-medium** (also smart-defaults the attributes **input-trays-supported** and **media-ready**)
- **media-ready**
- **media-supported**
- **output-bin-numbers** (also smart-defaults the attribute **output-bins-supported**)
- **plexes-supported**
- **psf-tray-characteristics** (also smart-defaults the attributes **input-trays-supported** and **media-ready**)
- **sides-supported**

For media, smart-defaulting only detects the **size** of the medium. If you have different media that are the same size (for example, three different letter-size preprinted forms) and you need to differentiate between them in the attributes, you must explicitly set the attributes instead of using smart–defaulting.

Detecting, reporting, and recovering from printer problems

The InfoPrint Manager server can automatically detect, report, and recover from printer problems for several **needs key operator** and **needs attention** conditions as it periodically queries an SNMP printer, even if no jobs are currently being printed. (To turn off automatic recovery, a key operator can manually disable the actual destination.)

For example, the server detects that a printer is out of toner, disables the actual destination, requeues the jobs, and sets the actual destination state to **needs attention** or **needs key operator**. Then, when a key operator replaces the toner, the server detects that the printer is running again, sets the actual destination state to **idle**, and re-enables the actual destination.

Besides automatically managing the state of the actual destination, the InfoPrint Manager server also updates its **problem-message**, **warning-message**, and **snmp-device-conditions** attributes and sends out printer-needs-attention and printer-needs-key-operator notifications. The actual destination's state, which message attribute is updated, and what type of notification is sent depends on the severity of the condition reported by the printer, such as error, warning, or report.

This is the list of SNMP printer conditions that InfoPrint Manager reports:

- Cover open
- Input tray empty
- Interlock open
- Media empty
- Media missing
- Offline
- Paper jam
- Power off

- Service requested
- Fuser over temp
- Input tray missing
- Printer supply empty
- Printer supply missing
- Printer waste full
- Output bin full
- Output bin missing
- Fuser under temp
- Printer supply low
- Printer waste almost full
- Media low
- Moving offline
- Moving online
- Output bin near full
- Warmup

Note

Not all printers report all these conditions. For example, most printers do not report the Warning conditions.

The same condition can be reported with a different severity depending on the situation. For example, a printer might report an input tray empty condition as an error or a warning. It could be a warning if the printer had multiple input trays linked together with the same media and only one of them is empty. If a condition is reported as an error, InfoPrint Manager reports these conditions as needs-key-operator as opposed to needs-attention:

- **fuser-over-temp**
- **fuser-under-temp**
- **marker-supply-empty**
- **marker-supply-low**
- **marker-supply-missing**
- **marker-waste-full**
- **marker-waste-almost-full**
- **media-missing**
- **output-bin-missing**
- **service-requested**
- **warmup**

Regardless of the severity, InfoPrint Manager lists all conditions on the **snmp-device-conditions** attribute.

Accessing device information

You can use the InfoPrint Manager Administration GUI or the **pdls** command to access information about SNMP devices.

Besides providing information about the actual destination object, InfoPrint Manager allows you to access information directly from the printer. InfoPrint Manager uses device-specific attributes to provide this function. While device-specific actual destination attributes are accessed through an actual destination, the information comes from the physical printer. The device-specific attributes are not returned when you ask for all the actual destination's attributes. You have to request the device-specific attributes specifically or you can use `-r all-device` to query all the device-specific attributes.

This is the list of non-settable device-specific actual destination attributes:

- **device-description**
- **device-input-trays**
- **device-ip-address**
- **device-manufacturer**
- **device-marker-supplies**
- **device-media-supported**
- **device-model**
- **device-op-panel-locked**
- **device-output-bins**
- **device-printer-name**
- **device-ready-media**
- **device-serial-number**
- **device-state**
- **device-version**

This is the list of settable device-specific actual destination attributes:

- **device-contact**
- **device-location**

Note

1. Not all printers report all these attributes.
2. The information returned is not translated.

The InfoPrint Manager Administration Interface can launch a Status window that displays information returned from SNMP. The window lets you see a detailed view of the status of an SNMP printer. It includes a graphic representation of the printer and its current status, such as whether it has a paper jam or a door open.

To see the Status window from the InfoPrint Manager Administration Interface:

1. Select an SNMP printer.
2. Select **Check Status**.
3. Select **More Information**.

You can use the **pdls** command to access SNMP device information with the **-r** flag, such as output-bins, like this:

```
pdls -c destination -r device-output-bins prt5-ad
```

Note

You can filter for specific device information with the **-f** flag.

You can also use the **pdls** command to access all the information for an SNMP device by specifying the **-r** flag with the **all-device** value.

For more information about the **pdls** command, see Chapter 2, "InfoPrint Manager Tools" in the *RICOH InfoPrint Manager: Reference*.

Changing device information

You can use the InfoPrint Manager Administration GUI, the **pddeviceop** command, and the **pdset** command to change information for SNMP devices.

The value of the **snmp-active** actual destination attribute must be **true**, and the printer device associated with the actual destination must support the operation. You must have write permission to the actual destination, and depending on the SNMP version that you use, the **snmp-write-community-name** (V1) or the **snmpv3-username** (V3) actual destination attributes must be set properly for the **pddeviceop** command or GUI commands to work.

From the **Device Information** window in the InfoPrint Manager Administration GUI, you can reset the specific device, take it offline, or put it online. You can also change the contact or location information.

You use the **pddeviceop** command to reset a specific device, take it offline, put it online, as well as lock and unlock the operation panel for the specific device.

You can use the **pdset** command or the InfoPrint Manager Administration GUI to change the contact or location information for SNMP devices.

Note

You cannot combine setting device-specific and actual destination attrributes in the same **pdset** command.

For more information about the **pddeviceop** and **pdset** commands, see the "InfoPrint Manager Tools" chapter in *RICOH InfoPrint Manager: Reference*.

Setting up and using SNMP

To set up and use SNMP with InfoPrint Manager, you must check and set the appropriate InfoPrint Manager attributes when you create or modify the physical printer in InfoPrint Manager.

This section explains:

- Determining if a printer works with SNMP
- Using the InfoPrint Manager Administration Interface
- Procedures

Determining if a printer works with SNMP

Not all printers work with SNMP and the standard MIB. In addition, only printers that are attached to a network can work with SNMP. Printers that are parallel-attached cannot use SNMP.

To determine if a printer works with SNMP:

1. Set the actual destination attribute **destination-tcpip-internet-address** to the IP address of the printer to indicate that InfoPrint Manager should try to communicate with the printer using SNMP. The server sets the actual destination attribute **use-snmp** to “true”.

   Note

   For PSF TCP-IP attached printers, a destination TCP-IP port number of 5002 must be used when driving the second engine in an InfoPrint 4000 or InfoPrint 4100 dual simplex configuration for SNMP support to work. If 5002 is used for a printer other than one using SNMP, the actual destination for that printer will not work. The server will set the actual destination attribute **use-snmp** to "false".
2. If the server cannot communicate with the printer using SNMP, it **pings** the printer. If the ping command:
   - **Works**, the server concludes that the printer is not an SNMP printer. It sets **use-snmp** to “false”.
   - **Does not work**, the server concludes that the printer is turned off or is not currently connected to the network. It leaves **use-snmp** set to “true” and continues trying to establish an SNMP connection with the printer.
3. Check the actual destination attribute **snmp-active**. If it is “true”, the printer works with SNMP.

Using the InfoPrint Manager Administration Interface

To use the InfoPrint Manager Administration Interface to check and set the attributes:

1. From the menu bar, select **Printer** or **Server**, as appropriate.
2. Select **Properties**. You see the printer properties or server properties notebook.
3. Select the **SNMP** tab.
4. Check and set the appropriate attributes on the SNMP tab as explained in Procedures, p. 35.

Procedures

Do these steps to check and set the attributes:

1. Check these actual destination attributes:
   1. For SNMPv1:
      snmp-community-name
      snmp-write-community-name

Note

InfoPrint Manager keeps both attributes and uses them as follows:

- If **snmp-write-community-name** attribute is set, it is used for both reading and setting SNMP data. In this case, **snmp-community-name** attribute is ignored.
- If **snmp-write-community-name** attribute is not set, **snmp-community-name** attribute is used. In this case, any attempts to set some SNMP data on the device should fail.

2. For SNMPv3:
   snmpv3-username
   snmpv3-authentication-type
   snmpv3-authentication-password
   snmpv3-privacy-type
   snmpv3-privacy-password
   snmpv3-security-level
   snmpv3-context-name
3. For SNMPv1 and SNMPv3:
   snmp-retry-count
   snmp-timeout
   snmp-active-version

   Note

   The **snmp-active-version** attribute is read only.

   snmp-version
   snmp-version-fallback
   use-snmp

2. If the defaults in step 1 in this section are correct, set this actual destination attribute:

   destination-tcpip-internet-address

   Indicates the IP address of the SNMP printer. The field on the tab is “TCP/IP address”.

3. Check this actual destination attribute:

   snmp-active

   Indicates whether the server has been able to establish an SNMP session with the printer. The field on the SNMP tab is “SNMP is active”.

4. Set these server attributes:

   - **snmp-discovery-during-startup**
     Indicates whether the server queries the printer MIB when InfoPrint Manager starts, or whether the server gets the printer status when it periodically polls the printer. The default is “true”, but you might want to set it to “false” if you have several SNMP printers and you want to make the server start faster. The field on the SNMP tab is “Discover printers at startup”.
   - **snmp-normal-poll-interval**

Indicates the number of seconds that the server should wait after sequentially polling all the SNMP printers **without** known problems. Increasing the polling interval causes the server to use less CPU and network resources for SNMP polling, but it makes the currently shown printer status (for printers that are not currently printing) less accurate. The field on the SNMP tab is "Normal poll interval".

- **snmp-problem-poll-interval**

  Indicates the number of seconds that the server should wait after sequentially polling all the SNMP printers **with** known problems. Increasing the polling interval causes the server to use less CPU and network resources for SNMP polling, but it makes the currently shown printer status (for printers that are not currently printing) less accurate. The field on the SNMP tab is "Problem poll interval".

For more information about the SNMP actual destination attributes, see the "InfoPrint Object attributes" chapter in the *RICOH InfoPrint Manager: Reference*.

Effects on performance with SNMP

The performance areas that SNMP can influence are:

- Server start-up
- Actual destination queries
- SNMP polling
- Number of SNMP printers on a server

Server start-up

If you set **snmp-discovery-during-startup** to "true", the server start-up might be slow because the server queries the printer MIB for each SNMP enabled actual destination for the most current information. Setting this attribute to "false" can improve startup performance. However, the initial actual destination status is not current and won't be until the server polls the printer next.

Actual destination queries

On an SNMP printer, when you query an actual destination attribute that might be affected by the printer MIB (for example, **destination-state**), the default is that the server uses SNMP at the same time to get the relevant information from the printer. If the printer is turned off or not connected to the network, it might take a long time before the printer times out, which delays the response to your query.

If this is a problem, you can use the **pdls** command with the **when=now** attribute to return the current information instead of querying the printer MIB. Using this attribute can improve performance.

SNMP polling

The server periodically polls the SNMP printers to get their latest status. If a printer condition changes while the server is polling a printer (for example, a user closes an open cover), the server might

temporarily miss one of the changes. However, during the next polling cycle, the server automatically picks up any changes that it missed.

If the server cannot discover an SNMP printer when InfoPrint Manager starts, it shows the printer status from the last time the server was shut down.

Increasing the polling interval causes the server to use less CPU and network resources for SNMP polling, but it makes the currently shown printer status (for printers that are not currently printing) less accurate.

Managed IPDS Dialog support

The Managed IPDS Dialog (MID) support enables additional sharing capabilities for specific printers. Current sharing of a print device is based on the setting of an inactivity timer. MID support overcomes these current limitations:

- Release for sharing requires physical intervention to disconnect the print device, and the PSF host must re-acquire the TCP/IP connection to process additional print requests. When the original connection is released, all downloaded resources are generally lost. Furthermore, there is a possibility that another host might acquire the TCP/IP connection and prevent the current host from processing additional print requests.
- Release of the print device is controlled by a specific time value. This causes the release of the printer even when there are no additional print requests pending.

With MID support, the printer can request that the PSF host release its control of the print device (but not drop the TCP/IP connection), thus allowing the print engine to print non-IPDS data streams when IPDS activity is minimal. MID support also enables the sharing of the print mechanism with other non-IPDS data streams.

This chapter includes the following sections:

- Enabling MID support, p. 38
- Considerations to activate MID support, p. 39

Enabling MID support

Set these parameters to enable MID support:

mid-support-enabled

1. Right-click the actual destination in the InfoPrint Manager GUI and click **Properties**.
2. Click **Tuning**.
3. Enter **Yes** in the **Enable MID Support** field to activate MID. The default value is **No**.

mid-release-timer

1. Right-click the actual destination in the InfoPrint Manager GUI and click **Properties**.
2. Click **Tuning**.
3. Specify a numeric value for **MID Release Timer** from 1 to 9999. The default value is **15**.

Note

The value for the **MID-release-timer** must be less than the value for the **destination-release-timer-setting**. The MID will never be enabled if the **destination-release-timer-setting** value is less than the **MID-release-timer** value.

Considerations to activate MID support

Consider these before you activate MID support for your print device:

- Your print device must be capable of supporting MID. Contact your InfoPrint representative if you are not sure of your print device capability.
- If you want incoming non-IPDS print requests to be intermingled with IPDS print requests, it is suggested that you maintain control over the print mechanism rather than activate MID.
- Jobs must be “stacked" before the print mechanism is released. This might lead to paper loss for the print device.
- Releasing the print mechanism might prevent the IPDS host from processing subsequent print requests until the control is returned to the host (the time depends on the capacity of each print device).
- The MID support does not allow for sharing of the print mechanism between two IPDS hosts for InfoPrint printers.

Configuring media

This section describes how to create, configure, and manage media. Use the Linux command line to complete these tasks.

Determining the media that are in the servers

You can use the Linux command lineto show the media in the servers being monitored.

Using the command line

To display all the media in specified servers, use the `pdls` command. For example, to display the media in `Server1`, enter:

```
pdls -c medium -r medium-identifier,associated-server,
    medium-type Server1:
```

InfoPrint returns information similar to this:

```
A4:          medium-identifier=A4
A4:          associated-server=Server1
A4:          medium-type=
letter:      medium-identifier=letter
letter:      associated-server=Server1
letter:      medium-type=
legal:       medium-identifier=letter
```

```
legal:       associated-server=Server1
legal:       medium-type=
A4-colored:  medium-identifier=A4-colored
A4-colored:  associated-server=Server1
A4-colored:  medium-type=stationery
```


For more information, see *RICOH InfoPrint Manager: Reference* or the `pdls` man page.

Creating a media object

You can use the Linux command line to create a media object.

Using the command line

To create a medium, use the `pdcreate` command and the medium attributes. Specify:

- The name of the server where you want to create the medium
- The medium identifier
- The size of the medium, using one of these attributes:
  - `medium-dimensions` specifies the width and length of the medium in millimeters. Be sure to enter the dimensions accurately so that printed output will be aligned correctly on both sides of the medium. You must use this attribute if CUPS actual destinations are to use the medium.
  - `medium-size` specifies a predefined size by name.

For example, to create a letter-size (8.5 by 11 inches) medium called `na-letter-colored` in `Server1`, enter this command:

```
pdcreate -c medium -x "medium-size=letter" Server1:na-letter-colored
```

For more information, see *RICOH InfoPrint Manager: Reference* or the `pdcreate` man page.

Viewing or changing attributes of a medium

You can use the Linux command line to view or change attributes of a medium.

Using the command line

To view the attributes of a medium, use the `pdls` command. For example, to view all the attributes of `na-letter-colored` in `Server1`, enter this command:

```
pdls -c medium -r all Server1:na-letter-colored
```

To change the attributes of a medium, use the `pdset` command. For example, to change the description of `na-letter-colored` in `Server1`, enter this command:

```
pdset -c medium -x "descriptor='Any color but white'"
   Server1:na-letter-colored
```

For more information, see *RICOH InfoPrint Manager: Reference* or the `pdls` and `pdset` man pages.

InfoPrint Email support

InfoPrint Email can be used for interoffice communication or to send e-mail to other addresses around the world. You can take advantage of the Email option to format documents that are both viewable and printable at company workstations. Send and receive data from a wide variety of sources, whether the data is PCL, PDF, PostScript, GIF, JPEG, TIFF, or a native SAP data stream (ABAP or OTF). Incoming data can be all these types; the Email DSS converts the data to PDF for viewing. To view output, you can use Adobe Acrobat Reader (for more information and available downloads, see the Adobe website at http://www.adobe.com).

Note

1. Color PDF and PostScript jobs are converted to black and white PDF e-mails if not handled appropriately.
2. To maintain the original document file name in the PDF attachment name from a job sent to the Email DSS, see Maintaining original document file names in the PDF attachment name, p. 43.

Configuring InfoPrint Email

Before you can send e-mail through InfoPrint Manager, you must make sure your SMTP email server is correctly configured and defined to InfoPrint and create an InfoPrint e-mail destination.

When you need E-Mail DSS, with SELinux in enforcing mode on a Red Hat Enterprise Linux, you must create and install a SELinux policy module:

1. Install `checkpolicy` and `policycoreutils` rpm files if not already installed:

```
yum install checkpolicy policycoreutils
```

2. Generate SELinux policy module:

```
cd /tmp
cp /usr/lpp/pd/selinux/ipmmail.te /tmp
/usr/bin/checkmodule -M -m -o ipmmail.mod ipmmail.te
/usr/bin/semodule_package -o ipmmail.pp -m ipmmail.mod
```

3. Install the new SELinux policy module:

```
semodule -i ipmmail.pp
```

4. Remove temporary files:

```
rm -f ipmmail.pp ipmmail.mod ipmmail.te
```

To revert the SELinux changes for InfoPrint Email, run:

```
semodule -r ipmmail.pp
```

Note

For more information, see Running InfoPrint Manager with SELinux set in enforcing mode, p. 170.

Defining the SMTP Email server to InfoPrint

The InfoPrint Manager for Linux Email DSS uses a Simple Mail Transfer Protocol (SMTP) server to send e-mail. The SMTP server must have MIME support. The SMTP server used is defined by the InfoPrint Manager server attributes **smtp-server-host** and **smtp-server-port**. These attributes default to the local Linux SMTP server. If you want to use a different SMTP server, you can modify these server attributes.

Note

Your SMTP server must be correctly configured to get outside the firewall if you plan to send e-mail outside your firewall.

To change the SMTP server for the InfoPrint Linux server:

1. To set the mail server at installation, start the **InfoPrint Manager Administration Interface** by entering this from the Linux command line: `ipguiadvadm`
2. From the InfoPrint Manager Administration Servers main window, click **Server** → **Properties**.
3. From the **Server Properties** dialog, specify your installation's mail server in the **SMTP server** field and the port in the SMTP **port** field.

   Note

   The **SMTP server port** field defaults to `25`, which is the default port for the SMTP server on most AIX servers.

If you plan to use e-mail to send large PDF files consisting of 50 pages or more, you will need to expand the size of the **/var/psf/segments** file system to 500 MB.

Creating an InfoPrint e-mail destination

Before you can send e-mail output through InfoPrint Manager, you must create an e-mail destination on your InfoPrint server. Use the **Advanced InfoPrint Manager Administration Interface** to do this task.

1. Start the **Advanced InfoPrint Manager Administration Interface** by entering this from the command line: `ipguiadvadm`
2. From the main window, click **Other** → **Create** → **Email Destination**.

   Note

   If the **Email Destination** option is not available under the **Other** → **Create** path, click the **Options** → **Add/Remove Menu Items** path and check the **Create Email Destination** from the **Customize** dialog. You can specify any actions (such as delete, enable, disable) from this dialog.
3. In the first panel of the Create Email Destination Wizard:
   1) Specify a **Name** for the e-mail destination, such as `email1`.
   2) Specify the InfoPrint **Server** where you are creating the e-mail destination, such as `server1`.
4. In the second panel of the Create Email Destination Wizard, select a logical destination to feed this e-mail actual destination.

   We recommend that you create a separate logical destination object (`email-ld`), so that the same logical destination does not feed jobs to both a printer and an e-mail destination. This step will make

it easier for operators to monitor the progress of the jobs through the system and make it easier to diagnose problems.

5. In the third panel (if you have decided to create a separate logical destination object in the preceding step), you can select where to put the new logical destination object.
6. In the fourth panel of the Create Email Destination Wizard, you can decide to take the default configuration or to further customize this destination by:
   - Changing the InfoPrint logical destination to which jobs are routed (if different from the logical destination defined in the preceding step).
   - Specifying particular job batch values.
   - Modifying actual destination properties.

   For purposes of this example, we take the default and apply the standard configuration.
7. In the fourth panel of the Create Email Destination Wizard, accept the default to enable the destination so that you have an e-mail destination with a status of **ready**.

Maintaining original document file names in the PDF attachment name

To preserve the original file name from the **document-file-name** attribute of the first printable document in a job sent to the Email DSS as the file name of the email attachment, you must configure the Email DSS actual destination attribute **maintain-original-filename** to **true**.

Configuring PDF settings for email jobs

The PDF settings parameters for **email jobs** are:

AFM_PATH=<*path***>**

This parameter specifies the path location of the Adobe Font Metrics (AFM) files that the transform uses. An AFM stores font metric data for a Type 1 PostScript font. It contains the master design of a specific font, which defines the way each character of the font looks. If a path is not specified, the default path is:

There is no command line flag associated with the **AFM_PATH** parameter.

CNV_PATH=<*path***>**

This parameter specifies the path location of the CNV files that the ICU library uses to convert text from one code page to Unicode and vice versa. If a path is not specified, the default path is:

There is no command line flag associated with the **CNV_PATH** parameter.

DEFAULT_CPID=<>*intValue*

This parameter specifies a default Code Page Identifier (CPID) to be used when transforming AFP or SCS files.

If a **DEFAULT_CPID** is not specified when transforming AFP files, a default value based on the `cpdef` file will be used.

If a **DEFAULT_CPID** is not specified when transforming SCS files, CPID 500 will be used as the default.

There is no command line flag associated with the **DEFAULT_CPID** parameter.

DEFAULT_FGID=<>*intValue*

This parameter specifies a default Font Global Identifier (FGID) to be used when transforming AFP or SCS files.

If a **DEFAULT_FGID** is not specified when transforming AFP files, a default value based on the `csdef` file will be used.

If a **DEFAULT_FGID** is not specified when transforming SCS files, FGID 11 (Courier) will be used as the default.

There is no command line flag associated with the **DEFAULT_FGID** parameter.

DISABLE_MEMSREAM=(TRUE|FALSE)

When this parameter is set to **TRUE**, it specifies that during the transformation, the content of the PDF document will be written in temporary files created on disk, no matter the size of the AFP document. This parameter is useful if the output PDF document is too large to be kept in memory until the transformation is completed. By default, the PDF document content is kept in memory for an AFP document smaller than 10 MB; otherwise, temporary files are used.

There is no command line flag associated with the **DISABLE_MEMSTREAM** parameter.

ENABLE_AUTO_FONT_IMAGE=(TRUE|FALSE)

When this parameter is set to **TRUE**, it specifies that the output PDF document will contain an image for each character that was written with a raster font in the AFP document. By default, the raster fonts will be replaced with another font, and all characters will be written as text in the PDF document. However, some of the characters might look different from their raster form.

There is no command line flag associated with the **ENABLE_AUTO_FONT_IMAGE** parameter.

FOCA_FONT_PATH=<*path***>**

This parameter specifies the path where the AFP FOCA fonts are installed. The default path is:

There is no command line flag associated with the **FOCA_FONT_PATH** parameter.

FONT_PATH=<*path***>**

This parameter specifies the path location of the font files that the transform uses when writing the PDF document. If a path is not specified, the default path is:

There is no command line flag associated with the **FONT_PATH** parameter.

GOCA_PASS1=<*number***>**

The *<number>* for this parameter can be **0**,**1**, or **2**.

When processing an afp file, there are 2 passes, an image pass and a text pass.

Setting this attribute will tell when to process the GOCA objects.

- GOCA_PASS1=*0*, goca objects processing are disabled.
- GOCA_PASS1=*1*, goca objects are processed in the image pass.
- GOCA_PASS1=*2*, goca objects are processed in the text pass.

The default value is **2**.

MAX_OBJECTS=<*number***>**

This parameter specifies the maximum objects that can be listed in the PDF cross-reference table (XREF). If the AFP file contains more objects than this, the additional ones are ignored. This setting is

provided for use with the **MAX_PAGES** setting. For example, if **MAX_PAGES** is set to a number greater than 1000, the value assigned to **MAX_OBJECTS** might need to be set to a number greater than 50000.

By default, **MAX_OBJECTS** and **MAX_PAGES** are commented out, which does not limit the number of objects or pages.

There is no command line flag associated with the **MAX_OBJECTS** parameter.

MAX_PAGES=<*number***>**

This parameter specifies the maximum pages to be converted. You can use this parameter to limit the number of pages generated by the transform. For example, if the number of pages in an AFP document is not known and you want to limit disk space usage, you might want to set a limit on the number of pages generated by the transform. Limiting the number of pages can also improve transform performance.

By default, **MAX_PAGES** and **MAX_OBJECTS** are commented out, which does not limit the number of pages or objects.

There is no command line flag associated with the **MAX_PAGES** parameter.

OVERLAYEXT=<*ListOfExtenstions***>PAGESEGEXT=<***ListOfExtenstions***>FORMDEFEXT=<***ListOfExtenstions***>JFIFext=<***ListOfExtenstions***>GIFEXT=<***ListOfExtenstions***>TIFFEXT=<***ListOfExtenstions***>CHARSETEXT=<***ListOfExtensions***>CODEPAGEEXT=<***ListOfExtensions***>CODEDFONTEXT=<***ListOfExtensions***>**

These parameters specify the accepted extension for a resource type. The possible extension are separated by a comma (,). For each extension, a * (asterisk) will be replaced with the name of the searched resource. Therefore, specifying an * (asterisk) means no extension.

When searching by name in the **RESOURCE_DATA_PATH**, the transform first tries the name with no extension. If it does not succeed, it tries with every extension from the list until the resource is found.

By default, the extension list is an * (asterisk).

There is no command line flag associated with these parameters.

PDF_FONT_MAP_FILE = <*path***>**

This parameter specifies the path location of the `fontmap.lst` file that maps the Type 1 Fonts. When specified, it means that the user wants to embed Type 1 Fonts into the PDF document. If a path is not specified, the default path is:

There is no command line flag associated with the **PDF_FONT_MAP_FILE** parameter.

PFMPFB_DIRECTORY=<*path***>**

This parameter specifies the path location of the Adobe Type 1 outline font files that the transform uses when embedding fonts inside of the PDF document. When simple font substitution is not acceptable in the PDF output, it is possible to embed a custom Type 1 font inside the PDF for better results. If a path is not specified, the default path is

Note

Placing Type 1 font files in this directory does not mean that they are automatically placed inside of the PDF file. For a font to be embedded, it must be mapped using the transform font definition files.

There is no command line flag associated with the **PFMPFB_DIRECTORY** parameter.

PRAGMA_FDEF= <*filename*>

This parameter specifies the file name of the form definition (FORMDEF) resource to be used when transforming the AFP file. If an inline form definition is associated with the AFP document, then both the **FDEF** from the configuration file and the command line are ignored. If **FDEF** is not included in the configuration file or specified on the command line, the F1A10110 default form definition is used.

Command line flag:

```
-pragma fdef <filename>
```

Command line example:

```
-pragma fdef=myformdef.fde
```

PRAGMA_ALIGNTEXT=<*angle*>

This parameter specifies the rotation value to use when transforming the file. Valid values are **0**, **90**, **180**, or **270** (clockwise). Some AFP files might have already been formatted with a rotated orientation. If this occurs, text is aligned with the vertical axis of the paper rather than the horizontal axis. To correct this, you must use this parameter to align the text as desired.

Command line flag:

```
-pragma aligntext <angle>
```

Command line example:

```
-pragma aligntext=90
```

VERBOSE

This parameter specifies verbose information will be generated to `stderr`.

Command line flag:

```
-v
```

OUTPUTFILE=<*path+filename*>

This parameter specifies the *path* and *filename* of the output PDF file. By default, the output PDF file is placed in the same directory as the input file, and it has the same file name as the input file with the extension changed from `afp` to `pdf`. For example, when the PDF is generated from an AFP file named `afpdoc.afp`, an output file named `afpdoc.pdf` is created.

Note

If AFP is not used as the extension, a `pdf` extension is still added to the output PDF file.

Command line flag:

```
-o <path+filename>
```

Command line example:

```
-o c:\mydirectory\mystream.pdf
```

RESOURCE_DATA_PATH =<path>

This parameter specifies the path location of the resource library. For all the external resources used by the AFP document (formdef, overlay, page segment, jfif, gif, IOCA image), the transform looks for them in this specified path. If a path is not specified, the default path is:

There is no command line flag associated with the **RESOURCE_DATA_PATH** parameter.

TT_FONT_PATH=<*path*>

This parameter specifies the location of the TrueType font or the TrueType Collection used. If a path is not specified, the default path is:

Submitting e-mail jobs

After you have installed and configured the Email option, you can submit jobs to a particular output destination by using the command line anywhere that you have the **pdpr** command installed. This topic shows the command syntax and provides three examples for submitting e-mail jobs:

- Submitting a basic e-mail job, p. 47
- Submitting attributes through a file with the -Z option, p. 48
- Submitting files with a shell script, p. 48
- Submitting text files with a shell script, p. 49
- Using the InfoPrint Select client, p. 50

Submitting a basic e-mail job

For information about the ***pdpr*** command and its attributes, see the *RICOH InfoPrint Manager: Reference* or the **pdpr** man page.

To submit a basic e-mail job, you can use this syntax:

```
pdpr -p email-ld -Z emailfilename
```

For an explanation of each variable on the command line, see this table:

Variable	Definition
logical_destination Example: `-p email-ld`	Specifies the logical destination that you defined when configuring the Email option through the **Advanced InfoPrint Manager Administration Interface**. For more information about configuring both logical and actual destinations and using the **pdpr** command, see the *RICOH InfoPrint Manager: Reference*.

emailfilename
Example:
`-Z`
`/email_script/samp.script`
	Specifies a file that contains the job attributes, the name of the file you are sending, and the e-mail destination to which you are sending the e-mail.
	E-mail attributes include cover sheet information, billing code information, and other options.
	Note
	subject-text, **email-body**, **email-from-adress**, **email-to-adress**, and **email-signature** are the only required attributes.

We recommend that you create sample attribute scripts that you can customize for your particular site. While you can specify e-mail attributes on the command line, we recommend using an attribute file.

Submitting attributes through a file with the -Z option

You can create a sample attribute file using an input date type of AFP, PostScript, or Portable Document Format and submit it as the *emailfilename* for the syntax listed above:

```
-x email-from-address=root@serv.us.name.com
-x email-to-address=johndoe@us.name.com
-x subject-text="Test Script"
-x document-type=printable
   /usr/lpp/ps/sample.pdf
-x document-type=email-body
   /email_script/body.txt
-x document-type=email-signature
   /email_script/sig.txt
```

On the command line, use this syntax to use this attribute file:

```
pdpr -p logical_destination -Z emailfilename
```

where *logical_destination* is **email1-ld** and *emailfilename* could be named **/email_script/samp.script** and stored on your local system. You must have defined a **body.txt** file and a **sig.txt** file for this script to work correctly.

This attribute file sends a report to a 'johndoe' Internet ID, using a logical destination (printer) named **email1-ld** that you specify from the command line. The convention *us.name.com* indicates a company-wide addressing system. It indicates the country (*us*), the company name *(name*), and the communications name (*com*) for all fully qualified addresses in the company.

You cannot put comments in an attributes file using the **-Z** option.

This script does not print ASCII input data. To print ASCII input files, see Submitting Text Files with a Korn Shell Script, p. 49.

Submitting files with a shell script

You can create a shell script (named **sendfile** for this example) with a simpler command syntax because the **pdpr** command is included in the shell script.

Create a file that contains the **pdpr** command and all its attributes. Create a shell script and put it in a directory on the system where you have installed the **pdpr** command.

The file should look similar to this:

```
# Sample Attribute File for InfoPrint Manager Email
      # using a script to submit
      # sendemail $1 $2
      # $1 is required as the name of the file to send.
      # $2 is optional as the recipient of the email.
      #
      #! /usr/bin/ksh
      if [[ -a $1 && -n $1] ]
      then
      if [ $2 ]
      then
```

```
      WHOTO=$2
      else
      WHOTO="johndoe@us.name.com"
      fi
      /usr/lpp/pd/bin/pdpr -p email1-ld \
      -r brief \
      -x email-to-address=$WHOTO \
      -x email-from-address=root@serv.us.name.com \
      -x "subject-text='Your requested Email file'" \
      -x document-type=printable $1 \
      -x document-type=email-body /email_script/body.txt \
      -x document-type=email-signature /email_script/sig.txt
      else
echo " sendfile: could not locate file [$1]"
      fi
```

On the command line, this shell script uses this syntax:

```
sendfile outputfilename [destination]
```

where *outputfilename* specifies the file that you are sending (**sample.pdf**) and destination is the e-mail address (**johndoe**) to which you are sending the file.

This script sends an e-mail to either the *destination* value that you specify as the second argument on the command line, or to 'johndoe' Internet ID, if you use the default. The script uses a logical destination (printer) named **email1-ld** on your InfoPrint server.

The convention *us.name.com* indicates a typical Internet address for a business.

Submitting text files with a shell script

You can email text files as they are, ripped at destination, or you can create a shell script (named **sendtxtfile** for this example) to submit ASCII text files as e-mail.

Create a file that contains the **pdpr** command and all its attributes. Create a shell script and put it in a directory on the system where you have installed the **pdpr** command.

The file should look similar to this:

```
# Sample Shell
# Script for InfoPrint Manager Email Text
      # Uses a script to submit:
      # sendtxtemail $1 $2
      # $1 is required as the name of the file to send.
      # $2 is optional as the recipient of the email.
      #
      # This script takes an ASCII file as input and creates
      #PostScript output that is submitted with the pdpr command
      # as email.
      #! /usr/bin/ksh
      if [[ -a $1 && -n $1] ]
      then
      if [ $2 ]
      then
      WHOTO=$2
      else
      WHOTO="johndoe@us.name.com"
      fi
      # Make temporary PostScript file
      enscript -p /tmp/$$email.txt $1
```

```
/usr/lpp/pd/bin/pdpr -p email1-ld \
-r brief \
-x email-to-address=$WHOTO \
-x email-from-address=root@serv.us.name.com \
-x "subject-text='Your requested Email file'" \
-x document-type=printable /tmp/$$email.txt \
-x document-type=email-body /email_script/body.txt \
-x document-type=email-signature /email_script/sig.txt
# Delete temporary PostScript file
rm /tmp/$$email.txt
else
echo " sendtxtfile: could not locate file [$1]"
fi
```

On the command line, this shell script uses this syntax:

```
sendtxtfile outputfilename [destination]
```

where *outputfilename* specifies the file that you are sending (**email.txt**) and destination is the e-mail address **(johndoe**) to which you are sending the file.

This script sends an e-mail to either the destination value that you specify as the second argument on the command line, or to 'johndoe' Internet ID, if you use the default. The script uses a logical destination (printer) named **email-ld** on your InfoPrint server. This file uses the **enscript** command to convert the text file to PostScript, stores that PostScript file in the **/tmp** directory on the submitting system, and then deletes that file after the **pdpr** command has submitted the PostScript file to the InfoPrint server.

The convention *us.name.com* indicates a typical Internet address for a business.

Using the InfoPrint Select client

After you have successfully installed InfoPrint Select and defined an e-mail destination (see the "Using InfoPrint Select to submit a job" chapter in *RICOH InfoPrint Manager for Linux: Getting Started* for more information), you can use this procedure to send e-mail.

Note

To configure the port for the InfoPrint server e-mail destination on your client workstation, the e-mail logical destination (in our example, **email-ld**) on the InfoPrint Manager server must have a value filled in for the **client-driver-names** logical destination attribute. If this value is not filled in, you will not be able to display printers from the InfoPrint Manager server when you click **Refresh Printers Selection List**.

You must also set the **destination-support-system** logical destination attribute to **email** for InfoPrint Select to send e-mail.

1. Specify **Print** from the screen menu.
2. From the **Print** window, make sure that the **Name** field contains the name of the logical destination that you defined to the InfoPrint Manager server, then click **OK**.
3. From the **InfoPrint Email** pop-up window, specify the attribute information in the fields as indicated.

Color Email

Sending documents as email attachments
If you have color Portable Document Format (PDF) or PostScript files, you can send these files to target e-mail address through InfoPrint Manager and receive color output.

If you want to submit color emails, you should set **color-bits-per-plane** attribute to **8**.

Make sure that your color PDF files pass through the Email DSS without being transformed no matter what method you use to submit the job to the InfoPrint Manager server. To do this, specify job attribute `job-ripped-by-server = false`. Or, you can set the actual destination to not rip PDF document formats. However, that would apply to all PDF jobs submitted.

Any job or document attributes that require the Email DSS to transform the data are not supported. For a list of these attributes and for more information about the `job-ripped-by-server` attribute, see *RICOH InfoPrint Manager: Reference*.

Note

Color Email is not supported if your e-mail job sent to an Actual Destination contains any of the attributes specified in the lists that follow. If these attributes or conditions are sent to an Actual Destination, the file will print as black-and-white (MO:DCA-P) output.

Submitting color e-mail attributes through a file

You can print a color e-mail job with a batch file that contains these:

- Job name
- E-mail body text
- E-mail to address
- E-mail from address

Here is the sample file:

```
-x job-name="Titleist"
-x job-ripped-by-server=false
-x email-from-address=gopher1@gopher.com
-x email-to-address=gopher@gopher.com
-x subject-text="testing"
-x document-type=email-body
-f d:\bsi\data\body.txt
-x document-format=pdf
-x document-type=printable
```

To submit this job to the InfoPrint server, specify this on a command line:

```
c:\Infoprint\pdpr -p email -Z d:\bsi\options\em_rip.txt  \
d:\bsi\data\LogoOnly.pdf
```

where **em_rip.txt** is the **-Z** file above.

Sending fax documents as email attachments via email DSS

Email DSS accepts fax jobs. If job attribute **fax-number** is set at job submission, InfoPrint Manager forwards the fax document as an email attachment to a recipient with the **email-to-address** composed by as **fax-number@fax-to-email-domain** from the **fax-number** and **fax-to-email** attribute values. For more information about sending fax documents as email attachments, see *RICOH InfoPrint Manager: Reference*.

Using CUPS DSS with SELinux set in enforcing mode

Using CUPS DSS with SELinux set in enforcing mode

This section describes special considerations for working with CUPS DSS with SELinux set in enforcing mode.

Running InfoPrint Manager with SELinux set in enforcing mode

When you print using CUPS DSS, with SELinux in enforcing mode on a Red Hat Enterprise Linux, you must create and install a SELinux policy module:

1. Install `checkpolicy` and `policycoreutils` rpm files if not already installed:

```
yum install checkpolicy policycoreutils
```

2. Generate SELinux policy module:

```
cd /tmp
cp /usr/lpp/pd/selinux/ipmcups.te /tmp
/usr/bin/checkmodule -M -m -o ipmcups.mod ipmcups.te
/usr/bin/semodule_package -o ipmcups.pp -m ipmcups.mod
```

3. Install the new SELinux policy module:

```
semodule -i ipmcups.pp
```

4. Remove temporary files:

```
rm -f ipmcups.pp ipmcups.mod ipmcups.te
```

Reverting the SELinux changes for CUPS DSS

To revert the SELinux changes for InfoPrint Manager, run:

```
semodule -r ipmcups.pp
```

User-exit programs (PSF only)

This section describes the sample user-exit programs supplied with InfoPrint Manager and how you can create your own user-exit programs.

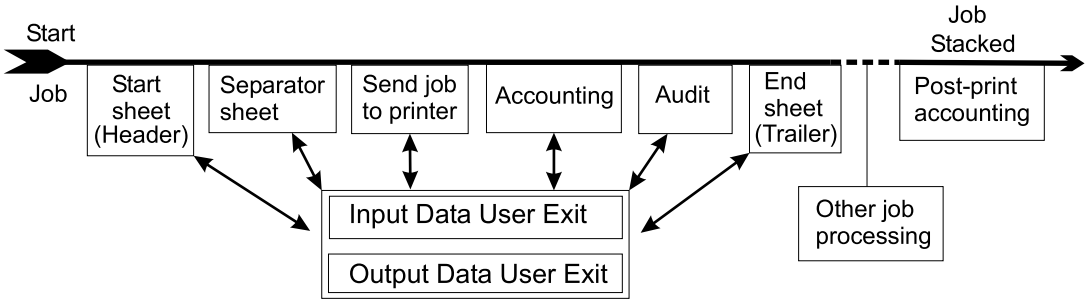
What type of user-exits does InfoPrint Manager support?

InfoPrint Manager supports dynamically loaded user-exit programs. These programs are loaded with the InfoPrint Manager executable program during initialization and are called at various points during job processing.

InfoPrint Manager allows eight different types of user-exits. InfoPrint Manager calls the user-exits in this order:

- Header (Start) page user-exit
- Separator page user-exit
- Input data user-exit
- Output data user-exit
- Accounting user-exit
- Audit user-exit
- Trailer (End) page user-exit
- Post-print accounting user-exit

Sequence of user-exit programs for InfoPrint Manager



What sample user-exit programs are available?

InfoPrint Manager supplies sample user-exit programs for the header, separator, and trailer pages, for accounting and audit data, and for post-print accounting data. The sample header, separator, and trailer page user-exit programs generate Advanced Function Presentation (AFP) data stream pages. The sample accounting and audit user-exit programs generate ASCII data in a report file format or write to a log file. The sample post-print accounting user-exit program only writes to a log file. InfoPrint Manager also supplies sample user-exit programs for input data and output data; however, these user-exit programs do not do any function. You can use all of these sample programs as they are or you can modify them.

If you choose to modify the existing user-exit programs, the source code for each one is located in the `/usr/lpp/psf/usr/lpp/psf/exits` directory. InfoPrint Manager supplies these sample programs:

Sample user-exit programs

Type of user-exit	File Name	Description
Header Page	ainuxhdr.c	Generates brief style start sheet.
	ainuxhdr2.c	Generates full style start sheet.
	ainuxhdrp.c	Generates start sheet with job ticket information.
	ainuxhdrx.c	Generates start sheet without vertical lines.
	pduxblkh.c	Generates a blank start sheet.
Separator Page	ainuxsep.c	Generates brief style separator sheet
	ainuxsep2.c	Generates full style separator sheet.
	ainuxsepp.c	Generates separator sheet with job ticket information.
	ainuxsepx.c	Generates separator sheet without vertical lines.
	pduxblks.c	Generates a blank separator sheet.
Trailer Page	ainuxtlr.c	Generates brief style end sheet.
	ainuxtlr2.c	Generates full style end sheet.
	ainuxtlrp.c	Generates start sheet with job ticket information.
	ainuxtlrx.c	Generates end sheet without vertical lines.
	pduxblkt.c	Generates a blank end sheet.
Accounting	ainuxacc.c	Generates brief style accounting sheet.
	ainuxacc2.c	Generates full style accounting sheet.
	ainacclog.c	Generates accounting log information.
	ainuxaccp.c	Generates accounting sheet log with job ticket information.
Post-print Accounting	ainuxjobcompletion.c	Generates accounting log information about printed ("stacked") sheets.
Audit	ainuxaud.c	Generates brief style audit sheet.

Type of user-exit	File Name	Description
	ainuxaud2.c	Generates full style audit sheet.
	ainuxaudp.c	Generates audit sheet with job ticket information.
	ainaudlog.c	Generates audit log information.
Input Data	ainuxind.c	Provides template for writing your own user-exit program.
Output Data	ainuxout.c	Provides template for writing your own user-exit program.

See User-exit programs for the line-data transform, p. 71 for sample user-exit programs used with the line-data transform.

Creating and using your own user-exit programs

You can create your own user-exit program at any time; however, you must do additional tasks to activate your program. You must have a C++ compiler installed for the Linux operating system. We recommend the GCC compiler, as the sample **Makefile** files are already customized for it.

Creating your own user-exit programs

If you have written any user-exit programs that use ICONV, you must alter them to use UCONV instead. See the *RICOH InfoPrint Manager: Reference* for more information.

You can create your own user-exit program in one of these ways:

- Copy one of the sample user-exit programs, rename it, and modify it using a text editor.
- Write your own user-exit program. You can include the variable data shown in the following sections, for example, `USERID`; but InfoPrint Manager cannot process variable data that is not shown in this chapter.

Compiling and installing the user-exit program

When you finish creating your own user-exit program, you must compile the program code and then install it in the correct directory. You do this by using the **make** command.

When the **make** command executes, it reads the **Makefile** file, supplied with InfoPrint Manager, that is located in the same directory as the sample programs. The **Makefile** file contains information about the specified target files and the commands necessary to update the target files.

Before you put your user-exit program into general use at your site, you should test it first to make sure that it runs correctly. Because your user-exit program becomes part of the InfoPrint Manager executable code, a defective user-exit program could cause problems with InfoPrint Manager.

Compile and install your user-exit program

To compile and install your user-exit program, follow these steps:

Note

You must have **root** authority to use the **make** command in this directory.

1. If you already have an existing executable user-exit program in the **/usr/lpp/psf/bin** directory for the user-exit program you are adding, do these steps first; otherwise, go to step 2.
   1) Disable all the logical destinations associated with the queues that are to receive jobs that will use the new user-exit program.
   2) Delete the executable user-exit program from the **/usr/lpp/psf/bin** directory.
2. To make the directory in which the user-programs are stored the current directory, enter:

```
cd /usr/lpp/psf/exits
```

3. On the command line, enter:

```
make
```

   The **make** command reads the **Makefile** file supplied with InfoPrint Manager, compiles and generates the user-exit program.

4. On the command line, enter:

```
make install
```

   The **make install** command copies the executable user-exit program from the **/usr/lpp/psf/exits** directory into the **/usr/lpp/psf/bin** directory so that the code executes with InfoPrint Manager.

5. Enable all of the logical destinations that you disabled.

Note

1. If you do not want to copy the executable user-exit program into the **/usr/lpp/psf/bin** directory, you can copy them into another directory. However, if you do this, you need to add the path to the **PATH** statement in the **/etc/environment** file so that InfoPrint Manager can locate the module.
2. You can only specify one user-exit program for each function name provided by InfoPrint Manager per queue.
3. Make sure the function name of the user-exit is one of the following where the name is in uppercase letters, because that is the entry point in the module:
   - ACCOUNTING
   - AUDIT
   - JOBCOMPLETION
   - HEADER
   - SEPARATOR
   - TRAILER
   - INDATA
   - OUTDATA
4. If you do not want to use the same name for the user-exit program, you must edit the **Makefile** file and change the name of the user-exit program supplied with InfoPrint Manager to the new name. For example, to call your new header user-exit program **header**, you must edit the **Makefile** file and change the name of the existing header user-exit program, **ainuxhdr** to **header**. In addition, you must change the source code file name from **ainuxhdr.c** to **header.c**. You can also add new exit targets in the **Makefile** file.
5. You must mark the function as an exported symbol during the link-editing step. See the InfoPrint Manager supplied **/usr/lpp/psf/exits/Makefile** for more information.

Activating the user-exit program

Before you can use the new user-exit program, you must activate the program.

From a command line, complete these steps:

1. Enter this command: `pdls -caux -rall servername:`, where *servername* refers to the name of the server you are using.

   This command displays the attributes for all current auxiliary sheet objects available on the server. You can also view attributes for a specific auxiliary sheet object. For example, to view attributes for the accounting log, enter this command:

```
pdls -caux -rall servername:accounting-log
```

2. Use the **pdset** command to change the values of any of the auxiliary sheet objects, as necessary.

To do this task from the InfoPrint Manager Administration Interface, you must select the printer that you want to associate with an auxiliary sheet (in this case, `prt1`) and then use the **Printer —> Properties** menu to access the **Printer Properties** notebook. To complete this task, see the online help topic **Using auxiliary-sheet objects** in the InfoPrint Manager Administration Interface.

Do this procedure from the command line.

1. To create a new auxiliary-sheet object, type:

```
pdcreate -c aux-sh -x psf-exit-prog-name=/usr/lpp/psf/exit_name
        server1:auxiliary_sheet_name
```

2. Shut down the actual destination (in this case, `prt1`).

```
pdshutdown -cp prt1
```

3. Associate this new auxiliary-sheet object (in this case, an audit exit) with the actual destination.

```
pdset -cp -x dest -x audit-exit=auxiliary_sheet_name prt1
```

Note

For the post-print accounting exit, instead of this step you will set the attribute in the accounting auxiliary sheet object for the actual destination, using this command:

```
pdset –c auxiliary-sheet –x psf-post-print-accounting-program-
name=/usr/lpp/psf/exit_name servername: accounting-log
```

By default, the accounting-log auxiliary sheet object is available for all destinations. This object can point to any combination of the accounting and post-print accounting exits. For example, you can set either one, both, or neither of these attributes at the same time:

```
psf-exit-program-name=name of accounting exit
psf-post-print-accounting-program-name=name of post-print
                                accounting exit
```

If you want some printers to have different accounting configurations than others, you must create a new auxiliary sheet object for each configuration.

4. Enable the actual destination (in this case, `prt1`).

```
pdenable prt1
```

See Creating and managing auxiliary sheets and user-exit programs, p. 80 for the procedures to activate the various user-exit programs.

User-exit program structures

All input and output variables used in InfoPrint Manager user-exits are defined through either of these two files:

- The **/usr/lpp/psf/exits/ainuexit.h** file
- The **/usr/lpp/psf/exits/ainurpt.h** file

The **ainuexit.h** file contains definitions for the input, output, header, trailer, separator, accounting, audit, and post-print accounting exits, while the **ainuprt.h** file formats data for these executable reporting utilities that apply to the accounting, audit, and post-print accounting exits:

- Accounting Log
  - **ainurpt1**
  - **ainurpt2**

  - **ainurpt3**
- Audit Log
  - **ainurpt4**
  - **ainurpt5**
  - **ainurpt6**
- Post-print Accounting Log
  - **ainurpt7**
  - **ainurpt8**
  - **ainurpt9**

These user-exit program structure files are written in the C programming language. The declarations and statements in these files show the structure of the InfoPrint Manager user-exit programs, which are imbedded as part of the user-exit program.

Common input and output fields

The following fields can be found in the user-exit programs. For a detailed list of input and output fields according to user-exit, see the **/usr/lpp/psf/exits/ainuexit.h** file. This information can be specified from:

- Migrated jobs
- Download for z/OS jobs
- Fields set in the InfoPrint Manager Administration Interface Job and Document Defaults notebook
- Fields in the InfoPrint SMIT administrator's interface `Submit an InfoPrint Job` panel

UserID

Identifies the user ID of the person who submitted the job. If you log in as yourself and then switch to **root**, the user ID is **root**.

NodeID

Identifies the name of the system (host). The backend program sets the NodeID.

Jobname

Identifies the name of the job. This is the same as the file name in the job.

Job submitters can also specify the job name with the InfoPrint Manager job attribute **job-name=** `name` when they submit a job.

If the job name is not specified for files defined as OTF, PCL, PostScript, ditroff, line data, unformatted ASCII, double-byte character set (DBCS) ASCII, EUC, PDF, JPEG, GIF, or TIFF files when the job is submitted, InfoPrint Manager creates a temporary file for the transform output. InfoPrint Manager uses the temporary file name for the job name.

SpoolID

Identifies the PSF job ID, which is an integer. InfoPrint Manager sets the SpoolID.

PrinterName

Identifies the first 64 characters of the InfoPrint Manager destination name.

hab

This function is not used.

Date

Identifies the date printed.

Time

Identifies the time printed.

Distribution

Identifies the distribution information provided when the job is submitted.

PagePointer

Points to a buffer containing the data produced by any of these exits:

- Header page
- Trailer page
- Separator page
- Accounting
- Audit

PageSize

Indicates the size of the data returned by these exits:

- Header page
- Trailer page
- Separator page
- Accounting
- Audit

This field is initially set to 0 before the exit is called. If no data is generated by the exit, this field remains set at 0.

PageType

Indicates the type of header or trailer page data (if any) that the exit generates. The field is initially set to 0 (AFP data stream) before the exit is called. Valid values are:

0

AFP data stream

1

ASCII data

Job Completion

Indicates the status returned by the exit. This field is initially set to 0 before the exit is called. Valid return codes are:

0

Good completion, all done, accept resultant page.

-1

Bad completion, non-fatal error, no page generated.

-5

Bad completion, fatal error. The exit is disabled until you shut down the actual destination.

-8

Bad completion, terminate the current job.

-12

Bad completion: terminate the job and put it in `pending` state. In addition, InfoPrint disables the actual destination.

Name

Identifies the name information provided when the job is submitted.

The **Name** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Userid

Identifies the userid information provided when the job is submitted.

The **Userid** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 10 characters.

Nodeid

Identifies the nodeid information provided when the job is submitted.

The **Nodeid** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 10 characters.

Address1

Identifies the first line of address information provided when the job is submitted.

The **Address1** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 57 characters.

Address2

Identifies the second line of address information provided when the job is submitted.

The **Address2** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 57 characters.

Address3

Identifies the third line of address information provided when the job is submitted.

The **Address3** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 57 characters.

Address4

Identifies the fourth line of address information provided when the job is submitted.

The **Address4** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 57 characters.

Account

Identifies the account information provided when the job is submitted.

The **Account** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 20 characters.

Building

Identifies the building information provided when the job is submitted.

The **Building** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Department

Identifies the department information provided when the job is submitted.

The **Department** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Programmer

Identifies the programmer information provided when the job is submitted.

The **Programmer** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Room

Identifies the room information provided when the job is submitted.

The **Room** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Title

Identifies the title information provided when the job is submitted.

The **Title** field value can be any null-terminated (X'00') character string. This value can be any null-terminated (X'00') character string no greater than 24 characters.

Passthru

Identifies any other information provided when the job is submitted that will be passed through the user-exit to the backend program.

These **Passthru** flags are supported:

class

Identifies the one-character class attribute.

destination

Identifies the one-to 64-character destination attribute.

forms

Identifies the one-to 64-character forms attribute.

segmentid

Identifies the one-to ten-character segmentation identifier.

ServerJobID

Identifies the job identifier.

The **Passthru** field value can be any null-terminated (X'00') character string no greater than 1,024 characters.

LongPrinterName

The full name of the InfoPrint destination.

Note

Because InfoPrint Manager only allows 64 character names, PrinterName and LongPrinterName are the same.

The header page and trailer page user-exit programs

Because the InfoPrint Manager start page and trailer page user-exit programs require the same inputs and produce the same outputs, the program descriptions are similar.

The source code (in the C programming language) for the sample header page user-exit programs and trailer page user-exit programs listed on page What sample user-exit programs are available?, p. 53 are located in the **/usr/lpp/psf/exits** directory. These sample programs generate an AFP data stream page.

The data structures for the header and trailer page user-exit programs are included with the source code in the **/usr/lpp/psf/exits/ainuexit.h** file. The code for these structures is shown in User-exit program structures, p. 58.

The declaration of these exits is:

```
void HEADER (HEADER_EXITDATA *exitdata)
void TRAILER (TRAILER_EXITDATA *exitdata)
```

The `HEADER_EXITDATA` and the `TRAILER_EXITDATA` input/output parameters contain all of the input and output data needed to communicate between InfoPrint Manager and the user-exit programs.

See Common input and output fields, p. 59 and the **ainuexit.h** file for explanations of the function and structure of the various components of this user-exit.

After you create your header page or trailer page user-exit program, you must use the **make** command to compile the code (see Compiling and installing the user-exit program, p. 55).

The separator page user-exit program

The InfoPrint Manager separator page user-exit program is called before copies of a job (including the very first copy) print.

The source code (in the C programming language) for the sample separator page user-exit programs listed on page What sample user-exit programs are available?, p. 53 are located in the **/usr/lpp/psf/exits** directory. These sample programs generate an AFP data stream page.

The data structures for the separator page user-exit program are included with the source code in the **/usr/lpp/psf/exits/ainuexit.h** file. The code for these structures is shown in User-exit program structures, p. 58.

The declaration of this exit is:

```
void SEPARATOR (SEPARATOR_EXITDATA *exitdata)
```

The `SEPARATOR_EXITDATA` input/output parameter contains all the input and output data needed to communicate between InfoPrint Manager and the user-exit program.

See Common input and output fields, p. 59 and the **ainuexit.h** file for explanations of the function and structure of the various components of this user-exit.

This field is also found in the separator page user-exit program and provides output information:

Copy

Indicates which copy is associated with this call to the exit. Initially set to 1, it increments by one each time the exit is called. The sample separator page user-exit program generates one separator page for the first copy and all subsequent copies of the job.

After you create your separator page user-exit program, you must use the **make** command to compile the code (see Compiling and installing the user-exit program, p. 55).

The accounting, post-print accounting and audit user-exit programs

The InfoPrint Manager accounting and audit user-exit programs require the same inputs and produce the same outputs so the program descriptions are the same. The post-print accounting user-exit program requires more inputs and produces more outputs.

The source code (in the C programming language) for the sample user-exit programs listed on page What sample user-exit programs are available?, p. 53 are in the **/usr/lpp/psf/exits** directory. The sample programs store the data into a file you can access with these executable reporting utilities:

ainurpt1

Provides accounting data for the actual destinations defined, based upon the destination ID.

ainurpt2

Provides accounting data for the actual destinations defined, based upon the userid.

ainurpt3

Provides detailed accounting data for the destinations defined, based upon the particular userid supplied.

ainurpt4

Provides audit data for the actual destinations defined, based upon the destination ID.

ainurpt5

Provides audit data for the actual destinations defined, based upon the userid.

ainurpt6

Provides detailed audit data for the actual destinations defined, based upon the particular userid supplied.

ainurpt7

Provides post-print accounting data stored in **jobcompletion.log**, sorted by actual destination.

ainurpt8

Provides summary post-print accounting data stored in **jobcompletion.log**, sorted by job submitter.

ainurpt9

Provides detailed entries of times and pages printed for a specific job submitter in **jobcompletion.log**.

The source code for these reporting utilities is also located in the **/usr/lpp/psf/exits** directory.

These executable reporting utilities are located in the **/usr/lpp/psf/bin** directory. By specifying these executable reporting utilities at the Linux command line, you can display data by either destination ID or userid. For example, to report on print requests submitted by a specific user to a actual destination for which the accounting user-exit has been activated, you can specify **ainurpt3** and then provide the userid to the program.

Note

You must activate these exits to generate the log files used by these report programs. For more information about activating these exits, see Activating the user-exit program, p. 57.

The data structures for the accounting, audit, and post-print accounting user-exit programs are included with the source code in the **/usr/lpp/psf/exits/ainuexit.h** and **/usr/lpp/psf/exits/ainurpt.h** files. The code for these structures is shown in User-exit program structures, p. 58.

The declarations of these exits are:

```
void ACCOUNTING (ACCOUNTING_EXITDATA *exitdata)
void AUDIT (AUDIT_EXITDATA *exitdata)
void JOBCOMPLETION (JOBCOMPLETION_EXITDATA *exitdata)
```

After you create your user-exit program, you must use the Linux **make** command to compile the code (see Compiling and installing the user-exit program, p. 55).

See Common input and output fields, p. 59 and the **ainuexit.h** file for explanations of the function and structure of the various components of this user-exit.

The `ACCOUNTING_EXITDATA`, `AUDIT_EXITDATA`, and `JOBCOMPLETION_EXITDATA` input/output parameters contain all the input and output data needed to communicate between InfoPrint Manager and the user-exit programs.

These fields are also found in the accounting, audit, and post-print accounting user-exit programs:

Pages Printed (accounting and audit user-exit only)

Shows the total number of impressions processed for this job.

Bin One Sheets Processed (accounting and audit user-exit only)

Shows the total number of processed sheets that were selected from the primary bin.

Bin Two Sheets Processed (accounting and audit user-exit only)

Shows the total number of processed sheets that were selected from bins other than the primary bin.

User Pages Stacked by Bin (post-print accounting user-exit only)

Shows the total number of stacked pages in the user print file. It does not include system pages, such as the header page, in the count.

User Sheets Stacked by Bin (post-print accounting user-exit only)

Shows the total number of stacked sheets in the user print file. It does not include system pages, such as the header page, in the count.

Pages Stacked by Bin (post-print accounting user-exit only)

Shows the total number of pages stacked for this job by bin.

Sheets Stacked by Bin (post-print accounting user-exit only)

Shows the total number of sheets stacked for this job by bin.

Data Object Resources

Shows the total number of Data Object Resources, which can consist of PDF pages, IOCA images, and Encapsulated PostScript (EPS) files, used in this job. Data Object Resources are also known as presentation-object containers (for more information, see the **presentation-object-container** attributes in the *RICOH InfoPrint Manager: Reference*).

Note

The counters do not include resources contained in the input file. The counters only include resources used before the accounting or audit user-exit is called. Therefore, the resources required for printing error messages and the trailer page do not appear in the resource totals.

Job Copies Requested (post-print accounting user-exit only)

Shows the total number of **results-profile** job copies requested in the job.

Fonts used

Shows the total number of fonts used in this job.

Note

The counters do not include resources contained in the input file. The counters only include resources used before the accounting or audit user-exit is called. Therefore, the resources required for printing error messages and the trailer page do not appear in the resource totals.

Overlays used

Shows the total number of overlays used in this job.

Note

The counters do not include resources contained in the input file. The counters only include resources used before the accounting or audit user-exit is called. Therefore, the resources required for printing error messages and the trailer page are not included in the resource totals.

Segments used

Shows the total number of page segments used in this job.

Note

The counters do not include resources contained in the input file. The counters only include resources used before the accounting or audit user-exit is called. Therefore, the resources required for printing error messages and the trailer page are not included in the resource totals.

ExtStart Date

Identifies the date the job started processing.

Start Date

The date the job started processing.

Start Time

The time the job started processing.

ExtStop Date

The date the job finished processing.

Stop Date

The date the job finished processing.

Stop Time

The time the job finished processing.

Completion Date (post-print accounting user-exit only)

The date all pages in the job printed and stacked.

Note

If errors or operator commands affect the job so that all pages are not sent to the printer, this field indicates the time when these pages have been printed and stacked.

Completion Time (post-print accounting user-exit only)

The time all pages in the job printed and stacked.

Note

If errors or operator commands affect the job so that all pages are not sent to the printer, this field indicates the time when these pages have been printed and stacked.

DORU

The data object resources used (DORU) by the exit.

DOFU

The data object fonts used (DOFU) by the exit.

The input data user-exit program

The InfoPrint Manager input data user-exit program can be used to:

- Monitor the incoming print data stream to InfoPrint Manager
- Discard an input record before it is processed
- Replace an existing input record with one or more records (which might include the original record or a modified version of the original record)
- Continue processing with the input record unchanged

This exit can be called for all parts of the job (see Sequence of user-exit programs for InfoPrint Manager, p. 53).

Note

The input data user-exit program only processes MO:DCA data.

If you are using the exit to insert extra input records, the original input record that is passed to the exit can be processed before or after the extra input records. This means that the extra input records can be inserted before or after the current structured field depending on where the current structured field is placed in the list of structured fields to insert.

Important considerations

- Code processing becomes an integral part of PSF and can affect performance and reliability.
- User-exits should be coded by experienced programmers who are aware of overall system implications.
- Any delays caused by this exit will translate into a PSF processing delay and **could affect printer performance**.
- This exit is not passed any structured fields that are contained in resources that are processed. This includes: form definitions, overlays, page segments, fonts, data object resources, internal medium maps, and data object fonts.
- Code processing must be able to handle PSF repositioning.

Repositioning can occur as a result of errors or operator commands. PSF can reposition items forward into the print job, skip structured fields, or reposition items back into previously processed data. If PSF repositions back into the print file, it is the responsibility of the exit to return the same data in the same sequence to PSF, just as it did when the structured field was originally passed to the exit.

Because of the constraints outlined above, you should first consider using the transform subsystem to process data, because the transform subsystem does not have these limitations. For example, a transform object would have access to all of the input data (no restrictions) and could add, delete, and modify both resources and pages. In addition, the transform subsystem processing is done before printing begins, so it would have no impact on printer performance or behavior.

The input data user-exit is not capable of identifying individual document boundaries in a data stream. For example, your job might have 5 separate documents that each contain 10 pages of data. This is viewed by the exit as 50 pages of data, not 5 units of 10. The transform subsystem does not have this limitation.

The PSF Input Record Exit is responsible for managing any memory it allocates. It can reuse the memory across calls through the use of a work area pointer provided in **`ainuexit.h`**. PSF does not free any memory that might be pointed to by fields in either the `INDATA_EXITDATA` structure or the `PSFINSERTRECS` structure. The exit can point to memory that it has defined or allocated. The exit can reuse allocated memory or free memory when it is appropriate. PSF might run out of memory if the PSF Input Record Exit mismanages memory.

The source code (in the C programming language) for the sample input data user-exit program listed on page What sample user-exit programs are available?, p. 53 is in the **/usr/lpp/psf/exits/ainuxind.c** file. This sample program consists of a return code and performs no function.

The data structures for the input data user-exit program are included with the source code in the **/usr/lpp/psf/exits/ainuexit.h** file.

The declaration of this exit is:

```
void INDATA (INDATA_EXITDATA *exitdata)
```

The **INDATA_EXITDATA** input/output parameter contains all of the input and output data needed to communicate between InfoPrint Manager and the user-exit program.

These fields are also found in the input data user-exit:

version

This variable was formerly called `reserved1`. It is designed for use as a migration aid for future variants of this exit. It contains a version identifier for this structure:

- X'00'= previous version (before August, 2005)
- X'01'= August, 2005 version adding support for deletion and insertion of input records

Copy

Indicates which copy is associated with this call to the exit. Initially set to 1, it increments by one each time the exit is called.

DataSize

Contains the length of the input data record

DataPointer

Points to the input data record to be processed.

DataType

Always set to 0 (AFP; MO:DCA data only).

Note

If the exit is being called to indicate end of job processing:

- DataSize is set to 0,
- DataPointer is set to NULL
- DataType is set to 0 — no record is passed in at that time

The **PSFINSERTRECS** structure contains information about a structured field that is to be inserted at the request of the input data exit. The **PSFINSERTRECS** structure contains these fields:

version

Version identifies to PSF the version of the **PSFINSERTRECS** structure that is being supplied. The exit must supply a value of 0x01 or the record (including the `next_p` pointer) is ignored.

next_p

Contains the pointer to the next record to insert.

record_p

Contains the pointer to the MO:DCA (AFP) structured field to be inserted.

Important

See more data definitions and comments in the `ainuexit.h` source file located in **/usr/lpp/psf/exits/**.

After you create your input data user-exit program, you must use the **make** command to compile the code (see Compiling and installing the user-exit program, p. 55).

Note

The input information fields used by some of the sample programs (see Common input and output fields, p. 59 for a complete list) are available through the alternate sample user-exits. InfoPrint Manager does not provide an alternate input data user-exit.

The output data user-exit program

The InfoPrint Manager output data user-exit program is used to monitor the outgoing print data stream from InfoPrint Manager. This exit is called at the end of the job before the error messages and the trailer page.

The source code (in the C programming language) for the sample output data user-exit program listed on page What sample user-exit programs are available?, p. 53 is in the **/usr/lpp/psf/exits/ainuxout.c** file. This sample program consists of a return and performs no function.

The data structures for the output data user-exit program are included with the source code in the **/usr/lpp/psf/exits/ainuexit.h** file. The code for these structures is shown in User-exit program structures, p. 58.

The declaration of this exit is:

```
void OUTDATA (OUTDATA_EXITDATA *exitdata)
```

The **OUTDATA_EXITDATA** input/output parameter contains all the input and output data needed to communicate between InfoPrint Manager and the user-exit program.

These fields are also found in the output data user-exit program:

Copy

Indicates which copy is associated with this call to the exit. Initially set to 1, it increments by one each time the exit is called.

DataSize

Indicates the number of bytes in the buffer.

DataPointer

Points to a buffer containing the outgoing print data stream.

After you create your output data user-exit program, you must use the Linux **make** command to compile the code (see Compiling and installing the user-exit program, p. 55).

Note

The output information fields used by some of the sample programs (see Common input and output fields, p. 59 for a complete list) are available through the alternate sample user-exits. InfoPrint Manager does not provide an alternate Output Data user-exit.

User-exit programs for the line-data transform

InfoPrint Manager provides several sample user-exit programs for the line-data transform. Use of the user-exits is optional. You specify the names of the exit programs with the `inpexit`, `indxexit`, `outexit`, and `resexit` keywords. Each of these keywords is described in the *RICOH InfoPrint Manager: Reference*.

InfoPrint Manager provides these sample programs:

/usr/lpp/psf/acif/apkinp.c

Input-record user-exit

/usr/lpp/psf/acif/apkout.c

Output-record user-exit

/usr/lpp/psf/acif/apkres.c

Resource exit

In addition, InfoPrint Manager provides these input-record user-exits programs to translate input data streams:

/usr/lpp/psf/acif/apka2e.c

Converts ASCII stream data to EBCDIC stream data.

/usr/lpp/psf/acif/asciinp.c

Converts unformatted ASCII data that contains carriage returns and form feeds into a record format that contains an American National Standards Institute (ANSI) carriage control character. This exit encodes the ANSI carriage control character in byte 0 of every record.

/usr/lpp/psf/acif/asciinpe.c

Converts unformatted ASCII data into a record format as does **asciinp.c**, and then converts the ASCII stream data to EBCDIC stream data.

The C language header file for all ACIF exit programs is provided in **/usr/lpp/psf/acif/apkexits.h** along with the build rules for the ACIF user-exits in **/usr/lpp/psf/acif/Makefile**

For more information about compiling user-exit programs, see Compiling and installing the user-exit program, p. 55.

Input record exit

The line-data transform provides an exit that enables you to add, delete, or modify records in the line-data input file. The program invoked at this exit is defined by the value of the `inpexit` keyword of the **line2afp** command.

This exit is called after each record is read from the input file. The exit can request that the record be discarded, processed, or processed with control returned to the exit for the next input record. The largest record that InfoPrint Manager can process is 32756 bytes. This exit is not called when the line-data transform is processing resources from directories.

Sample input record exit C language header, p. 72 contains a sample C language header that describes the control block that is passed to the exit program.

Sample input record exit C language header

```
/*********************************************************************/
/* INPEXIT STRUCTURE                                                 */
/*********************************************************************/
typedef struct _INPEXIT_PARMS /* Parameters for input record exit    */
{
   char           *work;      /* Address of 16-byte static work area */
   PFATTR         *pfattr;    /* Address of print file attribute info*/
   char           *record;    /* Address of the input record         */
   unsigned short in_CCSID;   /* input CCSID for translating     @06a*/
   unsigned short out_CCSID;  /* output CCSID for translating    @06a*/
   unsigned short recordln;   /* Length of the input record          */
   unsigned short reserved2;  /* Reserved for future use             */
   char           request;    /* Add, delete, or process the record  */
   char           eof;        /* EOF indicator                       */
} INPEXIT_PARMS;
```

The address of the control block containing these parameters is passed to the input record exit:

work (Bytes 1-4)

A pointer to a static, 16-byte memory block. The exit program can use this parameter to save information across calls (for example, pointers to work areas). The 16-byte work area is aligned on a full word boundary and is initialized to binary zeros before the first call. The user-written exit program must provide the code required to manage this work area.

pfattr (Bytes 5-8)

A pointer to the print file attribute data structure. See Attributes of the line-data input file, p. 79 for more information about the format of this data structure and the information it contains.

record (Bytes 9-12)

A pointer to the first byte of the input record including the carriage control character. The record is located in a buffer that is located in storage allocated by the line-data transform but the exit program is allowed to modify the input record.

in_CCSID (Bytes 13-14)

The value from the **INPCCSID** parameter on the **line2afp** command.

out_CCSID (Bytes 15-16)

The value from the **OUTCCSID** parameter on the **line2afp** command.

recordln (Bytes 17-18)

Specifies the number of bytes (length) of the input record. If the input record is modified, this parameter must also be updated to reflect the actual length of the record. If **recordln** is 0 when the exit program returns, **line2afp** will terminate with a return code of 99.

reserved2 (Bytes 19-20)

These bytes are reserved for future use.

`request (Byte 21)`

Specifies how the line-data transform processes the record. On entry to the exit program, this parameter is X'00'. When the exit program returns control to line-data transform, this parameter must have the value X'00', X'01', or X'02', where:

X'00'

Specifies that the line-data transform should process the record.

X'01'

Specifies that the line-data transform should not process the record.

X'02'

Specifies that the line-data transform should process the record and then return control to the exit program to allow it to insert the next record. The exit program can set this value to save the current record, insert a record, and then supply the saved record at the next call. After the exit inserts the last record, the exit program must reset the **request** byte to X'00'.

A value of X'00' on entry to the exit program specifies that the record be processed. If you want to ignore the record, change the `request` byte value to X'01'. If you want the record to be processed and you want to insert an additional record, change the `request` byte value to X'02'. Any value greater than X'02' is interpreted as X'00', and the exit processes the record.

Note

Only one record can reside in the buffer at any time.

`eof (Byte 22)`

An end-of-file (**eof**) indicator. This indicator is a one-byte character code that specifies whether an **eof** condition has been encountered.

When **eof** is signaled (**eof** value='Y'), the last record has already been presented to the input exit, and the input file has been closed. The pointer **record** is no longer valid. Records cannot be inserted when **eof** is signaled. These are the only valid values for this parameter:

Y

Specifies that **eof** has been encountered.

N

Specifies that **eof** has not been encountered.

This end-of-file indicator allows the exit program to do some additional processing at the end of the file. The exit program cannot change this parameter.

Using the line-data transform user input record exits

apka2e

The **apka2e** input-record exit program translates data that is encoded in one code character set (CCSID) to another. You should use this exit when your job uses fonts such as GT12, which has code points that are different from your data file. The default translation is from ASCII (code set 00850) to EBCDIC (code set 00037) if the **INPCCSID** and **OUTCCSID** parameters are not specified on the **line2afp** command. **INPCCSID** specifies the input code page, and **OUTCCSID** specifies the output code page.

To execute the **apka2e** input record exit program, set the following keywords and values in your line-data transform keyword file. The **line2afp** `parmdd` keyword identifies the keyword file.

```
inpexit=/usr/lpp/psf/bin/apka2e
cc=yes
cctype=z
```

asciinp

The **asciinp** input-record exit program transforms an ASCII data stream into a record format that contains a carriage control character in byte 0 of every record. If byte 0 of the input record is an ASCII carriage return (X'0D'), byte 0 is transformed into an ASCII space (X'20') that causes a data stream to return and advance one line; no character is inserted. If byte 0 of the input record is an ASCII form feed character (X'0C'), byte 0 is transformed into an ANSI skip to channel 1 command (X'31') that serves as a form feed in the carriage-control byte.

To execute the **asciinp** input record exit program, set the following keywords in your line-data transform keyword file. The **line2afp** `parmdd` keyword identifies the keyword file:

```
inpexit=/usr/lpp/psf/bin/asciinp
cc=yes
cctype=z
```

asciinpe

The **asciinpe** input-record exit program translates data that is encoded in one code character set (CCSID) to another. You should use this exit when your job uses fonts such as GT12, which has code points that are different from your data file. The default translation is from ASCII (code set 00850) to EBCDIC (code set 00037) if the **INPCCSID** and **OUTCCSID** parameters are not specified on the **line2afp** command. **INPCCSID** specifies the input code page, and **OUTCCSID** specifies the output code page.

While the **asciinp** and **asciinpe** input-record exits do not recognize other ASCII printer commands, you can modify these exits to account for these:

- backspacing (X'08')
- horizontal tabs (X'09')
- vertical tabs (X'0B')

For more information about using and modifying these programs, see the prologue of the **asciinp.c** source file that is provided with InfoPrint Manager in the **/usr/lpp/psf/acif** directory.

Output record exit

Using the output-record exit, you can modify or ignore the records the line-data transform writes into the output file. The program invoked at this exit is defined by the `outexit` keyword of the **line2afp** command.

The exit receives control before a record (structured field) is written to the output document file. The exit can request that the record be ignored or processed. The largest record that the exit can process is 32752 bytes, not including the record descriptor word. The exit is not called when the line-data transform is processing resources.

Sample output record exit C language header, p. 75 contains a sample C language header that describes the control block passed to the exit program.

Sample output record exit C language header

```
typedef struct _OUTEXIT_PARMS /* Parameters for resource record exit  */
{
   char           *work;      /* Address of 16-byte static work area  */
   PFATTR         *pfattr;    /* Address of print file attribute      */
                              /* information                          */
   char           *record;    /* Address of the record to be written  */
   unsigned short recordln;   /* Length of the output record          */
   char           request;    /* Delete or process the record         */
   char           eof;        /* Last call indicator                  */
  }OUTEXIT_PARMS;
```

The address of the control block containing these parameters is passed to the output record exit:

work (Bytes 1-4)

A pointer to a static, 16-byte memory block. The exit program can use this parameter to save information across calls (for example, pointers to work areas). The 16-byte work area is aligned on a full word boundary and is initialized to binary zeros before the first call. A user-written exit program must provide the code required to manage this work area.

pfattr (Bytes 5-8)

A pointer to the print file attribute data structure. See Attributes of the line-data input file, p. 79 for more information about the format of this data structure and the information it contains.

record (Bytes 9-12)

A pointer to the first byte of the output record. The record is located in a 32KB (where KB equals 1024 bytes) buffer. The buffer is located in storage allocated by the line-data transform, but the exit program is allowed to modify the output record.

recordln (Bytes 13-14)

Specifies the length, in bytes, of the output record. If the output record is modified, this parameter must also be updated to reflect the actual length of the record.

request (Byte 15)

Specifies how the line-data transform processes the record. On entry to the exit program, this parameter is X'00'. When the exit program returns control to the line-data transform, this parameter must have the value X'00' or X'01', where:

X'00'

Specifies that the line-data transform should process the record.

X'01'

Specifies that the line-data transform should ignore the record.

A value of X'00' on entry to the exit program specifies that the record be processed. If you want to ignore the record, change the `request` byte value to X'01'. Any value greater than X'01' is interpreted as X'00'; the exit processes the record.

Note

Only one record can reside in the buffer at any time.

eof (Byte 16)

An end-of-file (**eof**) indicator. This indicator is a one-byte character code that signals when the line-data transform has finished writing the output file.

When **eof** is signaled (**eof** value='Y'), the last record has already been presented to the output exit. The pointer **record** is no longer valid. Records cannot be inserted after **eof** is signaled. These are the only valid values for this parameter:

Y

Specifies that the last record has been written.

N

Specifies that the last record has not been written.

This end-of-file flag, used as a last-call indicator, allows the exit program to return control to the line-data transform. The exit program cannot change this parameter.

Resource exit

The line-data transform provides an exit that enables you to "filter" (exclude) resources. This exit is useful in controlling resources at the file-name level. For example, assume you only wanted to use those fonts that are not shipped with InfoPrint Manager. You could code this exit program to contain a table of all fonts shipped with InfoPrint Manager and filter those from the resource file. Security is another consideration for using this exit because you could prevent certain named resources from being included. The program invoked at this exit is defined by the `resexit`keyword of the **line2afp** command.

This exit receives control before a resource is read from a directory. The exit program can request that the resource be processed or ignored (skipped), but it cannot substitute another resource name in place of the requested one. If the exit requests that any overlay be ignored, the line-data transform automatically ignores any resources the overlay might have referenced (that is, fonts and page segments).

Sample resource exit C language header, p. 76 contains a sample C language header that describes the control block that is passed to the exit program.

Sample resource exit C language header

```
/*********************************************************************/
/* RESEXIT STRUCTURE                                                 */
/*********************************************************************/
typedef struct _RESEXIT_PARMS /* Parameters for resource record exit */
{
   char           *work;      /* Address of 16-byte static work area */
   PFATTR         *pfattr;    /* Address of print file attribute info*/
   char           resname[8]; /* Name of requested resource (8 byte) */
   char           restype;    /* Type of resource                    */
   char           request;    /* Ignore or process the resource      */
   char           eof;        /* Last call indicator                 */
   unsigned short resnamel;   /* Length of resource name         @05A*/
   char           pad1[3];    /* padding byte                    @05A*/
   char           resnamf[250]; /* Rsrc name if more than 8 bytes@05A*/
} RESEXIT_PARMS;
```

The address of the control block containing these parameters is passed to the resource-record exit:

work (Bytes 1–4)

A pointer to a static, 16-byte memory block. The exit program can use this parameter to save information across calls (for example, pointers to work areas). The 16-byte work area is aligned on a full word boundary and is initialized to binary zeros before the first call. A user-written exit program must provide the code required to manage this work area.

pfattr (Bytes 5–8)

A pointer to the print file attribute data structure. See Attributes of the line-data input file, p. 79 for more information about the format of this data structure and the information presented.

resname (Bytes 9–16)

Specifies the name of the requested resource. The exit program cannot modify or change this value.

restype (Byte 17)

Specifies the type of resource the name refers to. This is a one-byte hexadecimal value where:

X'03'

Specifies a GOCA (graphics) object

X'05'

Specifies a BCOCA (barcode) object

X'06'

Specifies an IOCA (IO image) object

X'40'

Specifies a font character set

X'41'

Specifies a code page

X'FB'

Specifies a page segment

X'FC'

Specifies an overlay

The line-data transform does **not** call this exit for these resource types:

- Page definition
  The page definition (**pagedef** keyword) is a required resource for transforming a line-data file.
- Form definition
  The form definition (**formdef** keyword) is a required resource for printing a transformed line-data file.
- Coded fonts
  The line-data transform must process coded fonts to determine the names of the code pages and font character sets they reference. This is necessary in creating Map Coded Font-2 (MCF-2) structured fields.

request (Byte 18)

Specifies how the line-data transform processes the resource. On entry to the exit program, this parameter is X'00'. When the exit program returns control to the line-data transform, this parameter must have the value X'00' or X'01' where:

X'00'

Specifies that the line-data transform should process the resource.

X'01'

Specifies that the line-data transform should ignore the resource.

A value of X'00' on entry to the exit program specifies that the resource be processed. If you want to ignore the resource, change the `request` byte value to X'01'. Any value greater than X'01' is interpreted as X'00' and the resource is processed.

`resnamel`

Specifies the length of the meaningful characters in the **resnamef** field.

`pad1`

A padding byte to get the **resnamef** on a word boundary.

`resnamef`

The actual name of the resource if it is greater then 8 characters in length. If the name is 8 characters or less in length, the name is stored in the **resname** field instead.

`eof (Byte 19)`

An end-of-file (**eof**) indicator. This indicator is a one-byte character code that signals when the line-data transform has written the last record.

When **eof** is signaled (**eof** value = 'Y'), the last record has already been presented to the resource exit. The pointer **record** is no longer valid. Records may not be inserted after **eof** is signaled. These are the only valid values for this parameter:

Y

Specifies that the last record has been written.

N

Specifies that the last record has not been written.

This end-of-file flag, used as a last-call indicator, returns control to the line-data transform. The exit program cannot change this parameter.

Non-zero return codes

If the line-data transform receives a non-zero return code from any exit program, the line-data transform issues message 0425-412 and terminates processing.

Attributes of the line-data input file

Note

This data structure is provided for informational purposes only.

The line-data transform provides information about the attributes of the line-data input file in a data structure available to the line-data transform user-exits. Sample print file attributes C language header, p. 79 shows the format of this data structure.

Sample print file attributes C language header

```
typedef struct _PFATTR  /* Print File Attributes                    */
{
   char     c[3] ;      /* Carriage controls? - "YES" or "NO "      */
   char     cctype[1];  /* Carriage control type - A(ANSI),         */
                        /* M(Machine), Z(ASCII)                     */
   char     chars[20];  /* CHARS values, including commas           */
                        /* (eg. GT12,GT15)                          */
   char     formdef[8]; /* Form Definition (FORMDEF)                */
   char     pagedef[8]; /* Page Definition (PAGEDEF)                */
   char     prmode[8];  /* Processing mode                          */
   char     trc[3];     /* Table Reference Characters - "YES"       */
                        /* or "NO "                                 */
} PFATTR;
```

The address of the control block containing these parameters is passed to the user-exits:

cc (Bytes 1-3)

The value of the cc keyword as specified with the **line2afp** command. The line-data transform uses the default value, **yes**, if this keyword is not explicitly specified.

cctype (Byte 4)

The value of the cctype keyword as specified with the **line2afp** command. The line-data transform uses the default value, **z** for ANSI carriage-control characters that is encoded in ASCII, if this keyword is not explicitly specified.

chars (Bytes 5-24)

The value of the chars keyword as specified with the **line2afp** command, including any commas that separate multiple font specifications. Because the chars keyword has no default value, this field contains blanks if no values are specified.

formdef (Bytes 25-32)

The value of the formdef keyword as specified with the **line2afp** command. Because the formdef keyword has no default value, this field contains blanks if no value is specified.

pagedef (Bytes 33-40)

The value of the pagedef keyword as specified with the **line2afp** command. Because the pagedef keyword has no default value, this field contains blanks if no value is specified.

prmode (Bytes 41-48)

The value of the prmode keyword as specified with the **line2afp** command. Because the prmode keyword has no default value, this field contains blanks if no value is specified.

trc (Bytes 49-51)

The value of the `trc` keyword as specified with the **line2afp** command. The line-data transform uses the default value, **no**, if this keyword is not explicitly specified.

Creating and managing auxiliary sheets and user-exit programs

Besides logical destinations, queues, and actual destinations, the InfoPrint server also controls auxiliary-sheet objects. This section contains information and procedures for creating and configuring these objects.

Preparing to work with auxiliary sheets

These sections describe auxiliary-sheet terms, default auxiliary sheets, and user-exit programs supplied by InfoPrint and provide examples of printed auxiliary sheets.

Terms relating to auxiliary sheets

To understand auxiliary sheets and how InfoPrint generates the information that appears on them, you need to understand these terms:

Auxiliary sheet

Represents a particular sheet of paper, blank or otherwise, that can precede a job, separate documents in a job, or follow the job. PSF and Linux actual destinations print auxiliary sheets.

Auxiliary-sheet object

Represents a set of user-exit programs that you can specify for InfoPrint PSF, and E-mail actual destinations. User-exit programs process information that can be printed on an auxiliary sheet or recorded in an accounting or audit log. Both the actual destination and the auxiliary-sheet objects must reside in the same InfoPrint server.

Note

Linux actual destinations print auxiliary sheets, but they do not support auxiliary-sheet objects.

Two auxiliary-sheet styles supplied with InfoPrint generate a file that you can use with an executable reporting utility to view job statistics. See Gathering accounting and auditing data about print jobs, p. 94 for more information about viewing statistics.

Auxiliary-sheet object name

Identifies the auxiliary sheet. The names of some of the default auxiliary sheets indicate the style of the auxiliary sheet: `brief`, `full`, or `blank`.

User exit

A point during InfoPrint processing that lets you run a user-exit program and return control of processing to InfoPrint after the user-exit program ends. InfoPrint provides data at each exit point that can serve as input to the user-exit program. InfoPrint can use the output from the user-exit program for InfoPrint processing; for example, a user-exit program can pass instructions about how to format a start sheet to InfoPrint. InfoPrint provides user exits for these InfoPrint auxiliary-sheet objects:

- Start sheet

- End sheet
- Separator sheet
- Accounting log
- Accounting sheet
- Audit log
- Audit sheet
- Input data stream
- Output data stream
- Post-print accounting (log file only)

Note

E-mail destinations support only the log, input data stream, and output data stream user exits.

See Sequence of user-exit programs, p. 87 for more information about user exits and user-exit programs.

User-exit program

Defines the data that prints and creates the format of the auxiliary sheet style. InfoPrint extends its function by providing user-exit programs that you can use to format start, end, or separator sheets for the job, track accounting and audit information, or extract information about the incoming and outgoing job data streams.

Form definition

Defines the characteristics of the form or printed media, which includes: overlays used, the paper source (for cut-sheet printers), duplex printing, text suppression, fonts, and the position of composed-text data on the form.

See *RICOH InfoPrint Manager: Reference* for more information about form and page definitions.

Edge marks

Marks that print at the edges of auxiliary sheets and help to identify job boundaries in printed output.

Interrupt Message Pages

Interrupt Message Pages are pages that are similar to the message pages that print at the end of a print job, but Interrupt Message Pages are printed at the point of an error where a possible data loss has occurred. See Using Interrupt Message Pages, p. 91 for more information.

InfoPrint default auxiliary-sheet objects

InfoPrint Manager provides default auxiliary-sheet objects which it creates when you first create a server. InfoPrint Manager automatically identifies these as supported values for PSF actual destinations. The default auxiliary-sheet objects with their associated user-exit programs and form definitions are shown in the next table.

Default auxiliary sheet objects

Auxiliary sheet object	Form definition	Description	User exit program	Additional Information
none	n/a	InfoPrint generates no auxiliary sheet.	n/a	n/a
brief	F1A10110	Start sheet	ainuxhdr	
		Separator sheet	ainuxsep	
		End sheet	ainuxtlr	
		Accounting exit	ainuxacc	
		Audit exit	ainuxaud	
full	F1A10110	Start sheet	ainuxhdr2	
		Separator sheet	ainuxsep2	
		End sheet	ainuxtlr2	
		Accounting exit	ainuxacc2	
		Audit exit	ainuxaud2	
blank	F1A10110	Start sheet	pduxblkh	Blank sheet of paper
		Separator sheet	pduxblks	
		End sheet	pduxblkt	
accounting-log	n/a	Accounting exit	ainacclog	Writes accounting information to the `/var/psf/accounting.log` file. Does not produce a printed sheet.
		Audit exit	ainaudlog	Writes audit information to the `/var/psf/audit.log` file. Does not produce a printed sheet.
		Post-print accounting exit	ainuxjob completion	Writes post-print accounting information to the `/var/psf/jobcompletion.log` file if the auxiliary sheet object's *psf-post-print-accounting-program-name* is set to ainuxjobcompletion. Does not produce a printed sheet.
job-ticket	n/a	Start sheet	ainuxhdrp	
		Separator sheet	ainuxsepp	
		End sheet	ainuxtlrp	

Auxiliary sheet object	Form definition	Description	User exit program	Additional Information
		Accounting exit	ainuxaccp	Writes accounting information to the `/var/psf/podaccount.log` Does not produce a printed sheet.
		Audit exit	ainuxaudp	
64xx, 65xx printers	F1A10110	Start sheet	ainuxhdrx	
		Separator sheet	ainuxsepx	
		End sheet	ainuxtlrx	

You can create your own auxiliary-sheet styles using the InfoPrint **pdcreate** command on the command line. See Creating new auxiliary-sheet objects using the pdcreate command, p. 89 for the procedure.

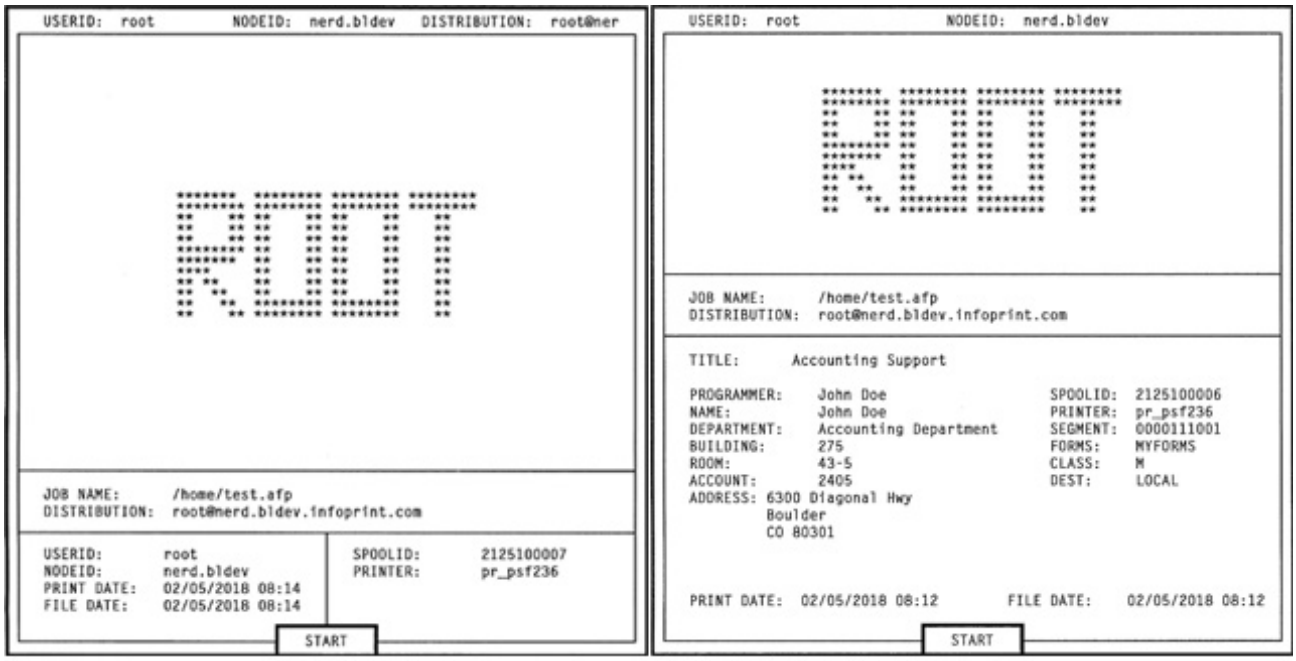
Examples of Brief-Style and Full-Style Auxiliary Sheets

These examples of auxiliary-sheet printouts were produced using the InfoPrint `brief` and `full` default auxiliary-sheet objects.

Start Sheet Examples

In Brief and Full Styles of Start Sheets, p. 83, the user-exit program **ainuxhdr** generated the brief style of the start sheet and user-exit program **ainuxhdr2** generated the full style of the start sheet.

Brief and Full Styles of Start Sheets

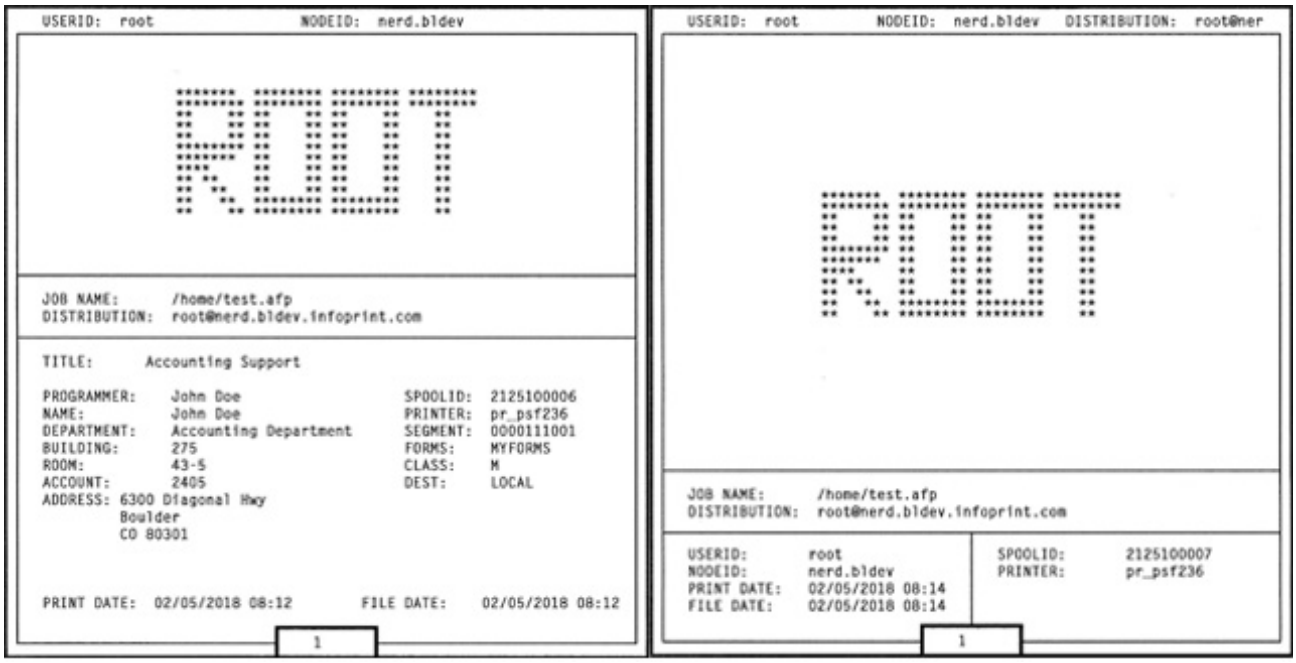


Separator Sheet Examples

In Brief and Full Styles of Separator Sheets, p. 84, the user-exit program **ainuxsep** generated the brief style of the separator sheet and user-exit program **ainuxsep2** generated the full style of the separator sheet.

1

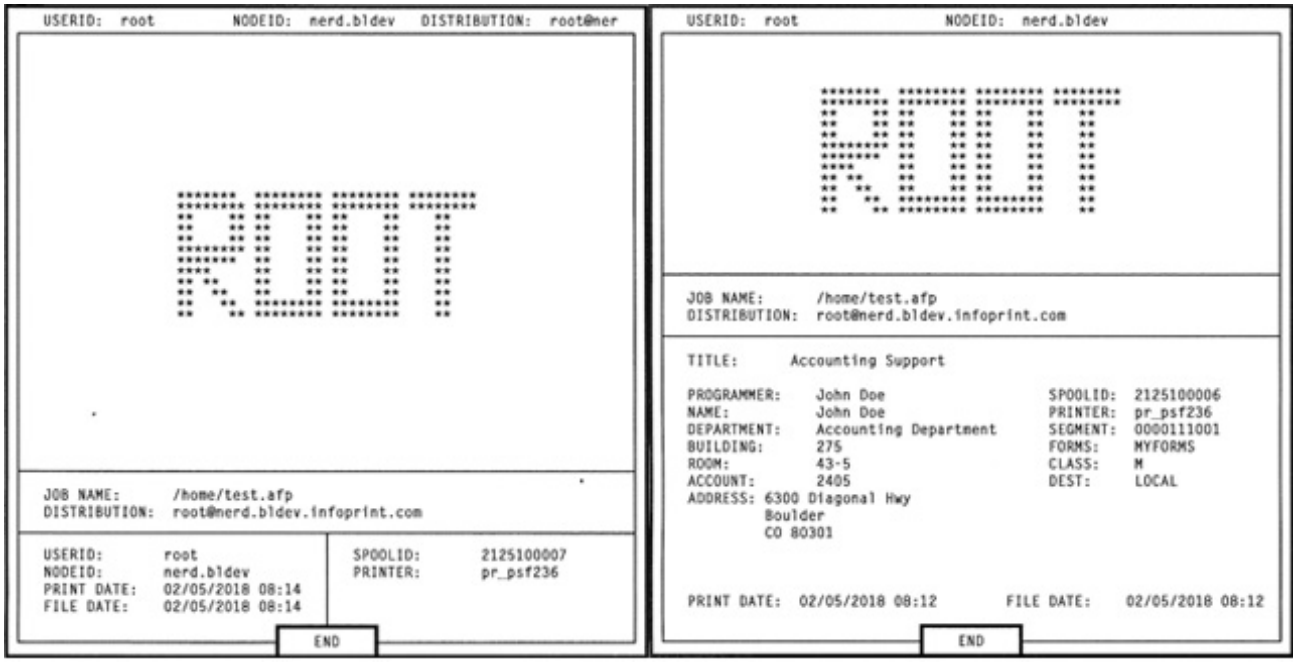
Brief and Full Styles of Separator Sheets



End Sheet Examples

In Brief and Full Styles of End Sheets, p. 85, the user-exit program **ainuxtlr** generated the brief style of the end sheet and user-exit program **ainuxtlr2** generated the full style of the end sheet.

Brief and Full Styles of End Sheets



Accounting Sheet Examples

In Brief and Full Styles of Accounting-Log Sheets, p. 86, the user-exit program **ainuxacc** generated the brief style of the accounting-log sheet and user-exit program **ainuxacc2** generated the full style of the accounting-log sheet.

Brief and Full Styles of Accounting-Log Sheets

```
ACCOUNTING:
UserID=root
NodeID=starwars
JobName=/info/afp/twopage.afp
SpoolID=1366
PrinterName=psf4317
PagesPrinted=3
BinOneSheets=3
BinTwoSheets=0
FontsUsed=2
ResidentFontsUsed=0
OverlaysUsed=0
HardSegmentsUsed=0
SoftSegmentsUsed=0
DataObjectResources=0
StartDate=11/01/2000
StartTime=14:50:14
EndDate=11/01/2000
EndTime=14:50:15
FileDate=05/05/2000
FileTime=07:59:13
ExtendedSpoolID=
```

```
ACCOUNTING:
System   UserID=root
System   NodeID=starwars
JobName=/info/afp/twopage.afp
SpoolID=1367
PrinterName=psf4317
PagesPrinted=3
BinOneSheets=3
BinTwoSheets=0
FontsUsed=2
ResidentFontsUsed=0
OverlaysUsed=0
HardSegmentsUsed=0
SoftSegmentsUsed=0
DataObjectResourcesUsed=0
StartDate=11/01/2000
StartTime=14:51:46
EndDate=11/01/2000
EndTime=14:51:48
FileDate=05/05/2000
FileTime=07:59:13

Name=
Supplied  Userid=
Supplied  Nodeid=
Address1=
Address2=
Address3=
Address4=
Account=
Building=
Department=
Programmer=
Room=
Title=
Segmentid=
Class=
Forms=
Destination=
Passthru=
ExtendedSpoolID=
```

IPRWAUXAU

Audit Sheet Examples

In Brief and Full Styles of Audit Sheets, p. 87, the user-exit program **ainuxaud** generated the brief style of the audit sheet and the user-exit program **ainuxaud2** generated the full style of the audit sheet.

Brief and Full Styles of Audit Sheets

```
AUDIT:
UserID=root
NodeID=starwars
JobName=/info/afp/twopage.afp
SpoolID=1366
PrinterName=psf4317
PagesPrinted=3
BinOneSheets=3
BinTwoSheets=0
FontsUsed=2
ResidentFontsUsed=0
OverlaysUsed=0
HardSegmentsUsed=0
SoftSegmentsUsed=0
DataObjectResources=0
StartDate=11/01/2000
StartTime=14:50:14
EndDate=11/01/2000
EndTime=14:50:15
FileDate=05/05/2000
FileTime=07:59:13
ExtendedSpoolID=
```

```
AUDIT:
System   UserID=root
System   NodeID=starwars
JobName=/info/afp/twopage.afp
SpoolID=1367
PrinterName=psf4317
PagesPrinted=3
BinOneSheets=3
BinTwoSheets=0
FontsUsed=2
ResidentFontsUsed=0
OverlaysUsed=0
HardSegmentsUsed=0
SoftSegmentsUsed=0
DataObjectResourcesUsed=0
StartDate=11/01/2000
StartTime=14:51:46
EndDate=11/01/2000
EndTime=14:51:48
FileDate=05/05/2000
FileTime=07:59:13

Name=
Supplied  Userid=
Supplied  Nodeid=
Address1=
Address2=
Address3=
Address4=
Account=
Building=
Department=
Programmer=
Room=
Title=
Segmentid=
Class=
Forms=
Destination=
Passthru=
ExtendedSpoolID=
```

IPRWAUXAU

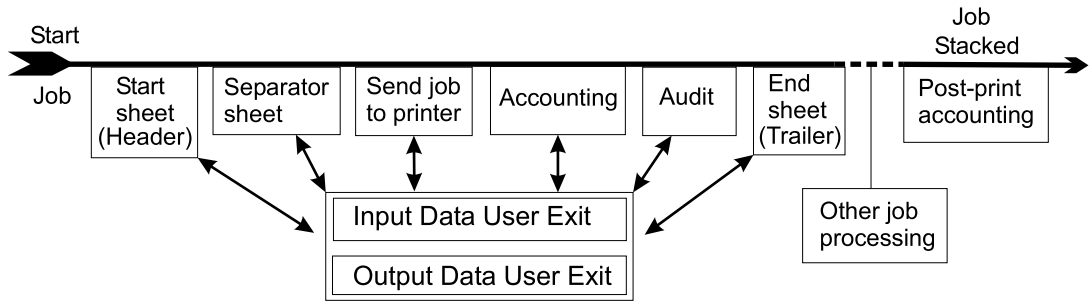
Sequence of user-exit programs

User exits for InfoPrint, p. 87 shows the order in which InfoPrint calls the user-exit programs during job processing. You cannot change this order.

Note

Separator sheets are printed only between copies of a job, not between documents in a job.

User exits for InfoPrint



Directory locations of user-exit programs supplied with InfoPrint

When you create the auxiliary-sheet object, you specify the name of the user-exit program and other options, such as the form definition, if the program produces printed output. You identify which auxiliary-sheet objects you want associated with each actual destination using the InfoPrint Manager Administration Interface attributes notebook for each actual destination. InfoPrint loads the user-exit programs that are called out into memory and runs the programs when you submit a job to that printer.

The following table shows the sample executable user-exit programs that InfoPrint supplies.

User-exit programs

Program	*Description*
Start Sheets	
/usr/lpp/psf/bin/pduxblkh	Prints a blank start sheet.
/usr/lpp/psf/bin/ainuxhdr	Prints a brief-style start sheet.
/usr/lpp/psf/bin/ainuxhdr2	Prints a full-style start sheet.
/usr/lpp/psf/bin/ainuxhdrp	Prints job ticket information about the start sheet.
/usr/lpp/psf/bin/ainuxhdrx	Prints a start sheet without vertical lines for 64xx & 65xx printers.
Separator Sheets	
/usr/lpp/psf/bin/pduxblks	Prints a blank separator sheet.
/usr/lpp/psf/bin/ainuxsep	Prints a brief-style separator sheet.
/usr/lpp/psf/bin/ainuxsep2	Prints a full-style separator sheet.
/usr/lpp/psf/bin/ainuxsepp	Prints job ticket information about the separator sheet.
/usr/lpp/psf/bin/ainuxsepx	Prints a separator sheet without vertical lines for 64xx & 65xx printers.
End Sheets	
/usr/lpp/psf/bin/pduxblkt	Prints a blank end sheet.
/usr/lpp/psf/bin/ainuxtlr	Prints a brief-style end sheet.
/usr/lpp/psf/bin/ainuxtlr2	Prints a full-style end sheet.
/usr/lpp/psf/bin/ainuxtlrp	Prints job ticket information about the end sheet.
/usr/lpp/psf/bin/ainuxtlrx	Prints an end sheet without vertical lines for 64xx & 65xx printers.
Accounting	
/usr/lpp/psf/bin/ainuxacc	Prints a brief-style accounting sheet.
/usr/lpp/psf/bin/ainuxacc2	Prints a full-style accounting sheet.

Program	*Description*
/usr/lpp/psf/bin/ainuxaccp	Writes accounting information to `/var/psf/podaccount.log`.
/usr/lpp/psf/bin/ainacclog	Writes accounting information to `/var/psf/accounting.log`.
Post-print Accounting	
/usr/lpp/psf/bin/ainuxjobcompletion	Writes accounting information to `/var/psf/jobcompletion.log`.
Audit	
/usr/lpp/psf/bin/ainuxaud	Prints a brief-style audit sheet.
/usr/lpp/psf/bin/ainuxaud2	Prints a full-style audit sheet.
/usr/lpp/psf/bin/ainuxaudp	Prints job ticket information about the audit sheet.
/usr/lpp/psf/bin/ainaudlog	Writes accounting information to `/var/psf/audit.log`.
Input Data	
/usr/lpp/psf/bin/ainuxind	Performs no tasks; use it as a model for writing your own user exit program.
Output Data	
/usr/lpp/psf/bin/ainuxout	Performs no tasks; use it as a model for writing your own user exit program.

In addition, you can use the source code of the sample user-exit programs to write your own user-exit programs. InfoPrint provides the source code in the **usr/lpp/psf/exits** directory.

Creating and configuring auxiliary-sheet objects (PSF only)

When you first create a server, InfoPrint creates the default auxiliary-sheet objects described in InfoPrint default auxiliary-sheet objects, p. 81. The following topics describe how you can create and configure auxiliary-sheet objects.

Creating new auxiliary-sheet objects using the pdcreate command

Create auxiliary-sheet objects from the command line; you cannot create them through the InfoPrint Manager Administration Interface. Use **-c auxiliary-sheet** with the **pdcreate** command and specify the server name and an auxiliary-sheet object name to create a new auxiliary-sheet object. Specify a name that uniquely identifies the auxiliary-sheet object in the server. Two servers can each contain an auxiliary-sheet object with the same name, but possibly with different values for their attributes.

You also specify the **psf-exit-form-definition**, **psf-exit-page-mark**, and **psf-exit-program-name** attributes with the command to specify the form definition, whether edge marks print, and the user-exit program used.

For example, to create an auxiliary-sheet object named `custom-start` contained in `Server1` that specifies the `ainuxhdr` user-exit program, the `F1A10110` form definition, and includes edge marks, enter the command:

```
pdcreate -c auxiliary-sheet -x "psf-exit-program-name=ainuxhdr
        psf-exit-form-definition=F1A10110 psf-exit-page-mark=true"
        Server1:custom-start
```


See "Attributes for Auxiliary Sheet Objects" in the "Attributes for auxiliary-sheet objects" chapter of *RICOH InfoPrint Manager: Reference* for more information.

Associating auxiliary-sheet objects with PSF printers

Use the **InfoPrint Manager Administration Interface** to associate auxiliary sheet objects with PSF printers. Start the InfoPrint Manager Administration Interface and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer** → **Properties** to open the **Printer Properties** notebook. To complete this task, see the online help in the **InfoPrint Manager Administration Interface**, starting with the topic *Using auxiliary-sheet objects*.

Associating auxiliary sheets for accounting or audit information

You can use the **InfoPrint Manager Administration Interface** to specify auxiliary sheet objects to collect accounting and audit information about jobs processed (as opposed to printed) by a selected PSF printer. Start the InfoPrint Manager Administration Interface and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer** → **Properties** to open the **Printer Properties** notebook. To complete this task, see the online help in the **InfoPrint Manager Administration Interface**, starting with the topic *Using auxiliary-sheet objects*.

To activate the accounting user-exit program , specify one of the accounting user-exit programs names as the value for the psf-exit-program-name (for example, `psf-exit-program-name=ainacclog`) in the auxiliary sheet object associated with the destination. To activate the audit user-exit program, specify one of the audit user-exit program names as the value for the psf-exit-program-name (for example, `psf-exit-program-name=ainaudlog`) in the auxiliary sheet object associated with the destination.

Each PSF actual destination can have one, both, or neither or the two accounting methods (accounting and post-print accounting) active. If the accounting setup is intended to be the same for all PSF printers, this can all be done with one auxiliary sheet object, which can either be one you have created or the default accounting-log auxiliary sheet object supplied by InfoPrint.

Associating the post-print accounting user-exit program with an actual destination

You can use the **InfoPrint Manager Administration Interface** to specify auxiliary sheets to collect post-print accounting information about jobs processed (as opposed to printed) by a selected PSF printer. Start the InfoPrint Manager Administration Interface and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer** → **Properties** to open the **Printer Properties** notebook. To complete this task, see the online help in the **InfoPrint Manager Administration Interface**, starting with the topic *Using auxiliary-sheet objects*. Then set the attribute of the auxiliary sheet object using the command shown in Activating the user-exit program, p. 57.

To activate the post-print accounting exit program, specify `psf-post-print-accounting-program-name=ainuxjobcompletion` in the auxiliary sheet object associated with the destination.

Each PSF printer actual destination can have one, both, or neither of the two accounting methods (accounting and post-print accounting) active. If the accounting setup is intended to be the same for all PSF printers, this can all be done with one auxiliary sheet object, which can either be one you have created or the default accounting-log auxiliary sheet object supplied by InfoPrint.

Associating an input data user-exit program with an actual destination

You can use the input data user-exit program to monitor the contents of the data stream coming into InfoPrint Manager from the **InfoPrint Manager Administration Interface**. Start theInfoPrint Manager Administration Interface and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer** → **Properties** to open the **Printer Properties** notebook. To complete this task, see the online help in the **InfoPrint Manager Administration Interface**, starting with the topic *Using auxiliary-sheet objects*.

Associating an output data user-exit program with an actual destination

You can use the output data user-exit program to monitor the contents of the data stream created by InfoPrint from the **InfoPrint Manager Administration Interface**. Start the InfoPrint Manager Administration Interface and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer** → **Properties** to open the **Printer Properties** notebook. To complete this task, see the online help in the **InfoPrint Manager Administration Interface**, starting with the topic *Using auxiliary-sheet objects*.

Using Interrupt Message Pages

Interrupt Message Pages provide a way for printer operators to find the cause and location of a printing problem. Interrupt Message Pages are pages that are similar to the message pages that print at the end of a print job, but Interrupt Message Pages are printed at the point of an error where a possible data loss has occurred. This will allow the operators to make sure that no data is missing, or to compensate for issues during the post-processing phase of the print job.

Interrupt Message Pages are for use with PSF TCP/IP attached printers using the PSF DSS or the Distributed Print Function (DPF). They are associated with an Actual Destination (AD), and cannot be changed for each job. Interrupt Message Pages do not work with other PSF attachment types. Interrupt Message Pages and finishing processes are incompatible and we do not recommend combining these functions.

Scenario/Condition	Produce interrupt message page?
Pause/Resume job	Yes
Pause/Resume printer	No
AD shutdown - *Synchronous* (same as pause job)	Yes
AD shutdown - *Now* (stop as soon as possible)	No

Scenario/Condition		Produce interrupt message page?
Recoverable data and hardware exceptions (including paper jams, pre-/post-processor exceptions)		Yes
Back/Forward spacing		Yes
Back/Forward spacing with no pages committed		No
Print a page range		No
Cancel job	- Pdrm/pddelete	No
	- Cancel button on the printer	Yes
Stop button on the printer		No - on `ready/not ready`
Communication lost		No

Enabling Interrupt Message Pages

Follow this procedure to enable a printer to produce Interrupt message pages:

1. Open the InfoPrint Manager Administration GUI.
2. Right-click the printer that you want to enable and select **`Properties`**. The properties notebook will open for that printer.
3. Click **`Auxiliary/Separator Sheets`**.
4. Click **`Yes`** next to **`Interrupt message page enabled`**.
5. If you want to use a form definition that is different than the default, enter the name of the definition in the **`Interrupt message page form definition`** entry field.
6. If you want to use page marks, click **`Yes`** next to **`Interrupt message page edge marks`**. The **`Interrupt message page edge marks`** setting prints a mark on the edge of the forms so that the mark is visible in a stack of paper when viewed from the side. Continuous forms printers that are folding the forms will print the mark where the fold in the pages occurs.

Interrupt Message Pages have a default message page form definition that is already assigned and is designed to print on all paper sizes. This form definition is named **`F1MG0110`**, and has these default settings:

- One copy
- No overlay
- Use paper from bin 1

There is also a new form definition named **`F1MGIMP`** that is identical to the default form definition, except that it specifies offset stacking.

You can create your own form definition, but be aware that doing so might alter the ability of your form definition to print on all paper sizes. To create a form definition, you must use the Page Printer Formatting Aid (PPFA) feature. See the *InfoPrint Page Printer Formatting Aid: User's Guide and Reference*, S550-0801 for more information.

If you are creating your own form definition and wish to use offset stacking, you can use PPFA and specify `JOG YES` in the form definition.

Using Interrupt Message Report

Interrupt Message Report includes all interrupt error messages already printed on separate Interrupt Message Pages to provide printer operators with a summary of the most critical error messages from the entire processing phase of the print job.

Interrupt Message Report is printed at the end of a print job, after the message pages.

The messages occurred during a print job are listed in this order:

1. Interrupt Message Page
2. Message Page
3. Interrupt Message Report

Enabling Interrupt Message Report

Follow this procedure to enable a printer to produce Interrupt Message Reports:

1. Open the **InfoPrint Manager Administration GUI.**
2. Right-click on the printer that you want to enable and select `Properties`. The properties notebook will open for that printer.
3. Select `Auxiliary/Separator Sheets`.
4. Click `Yes` next to `Interrupt message report enabled`.

   Note

   - Make sure that Interrupt Message Pages is enabled so that you are able to generate the Interrupt Message Report. For more information, see Enabling Interrupt Message Pages, p. 92.
5. To use a form definition that is different than the default, enter the name of the definition in the `Interrupt message report form definition` entry field.

Interrupt Message Report has a default message report form definition that is already assigned and is designed to print on all paper sizes. This form definition is named `F1PIRM1` and has these default settings:

- Greyed- out background overlay
- Bottom page overlay section with the text "INTERRUPT MESSAGE REPORT"

Allowing the auxiliary-sheet-selection job attribute with the BSD DSS

BSD actual destinations do not generate auxiliary sheets. The system where the printer device is attached can create auxiliary sheets, but you cannot control them using InfoPrint Manager attributes.

For example, an auxiliary sheet, such as a header sheet, could be printed if the job was sent to a remote queue that is actually printing auxiliary sheets.

BSD actual destinations do not support the **auxiliary-sheet-selection** attribute. Many print command options generate an **auxiliary-sheet-selection** attribute, and the BSD actual destinations will reject these jobs. Some examples for the Linux client follow:

Command	Action
`lpr -Pbsdld file`	Sets **auxiliary-sheet-selection=start** in the job.
`lpr -Pbsdld -h file`	Sets **auxiliary-sheet-selection=none** in the job.

You can use this procedure to allow BSD DSS actual destinations to accept jobs with an **auxiliary-sheet-selection** attribute. Although this procedure allows BSD DSS destinations to accept jobs with the **auxiliary-sheet-selection** specified, the BSD DSS does not process or print auxiliary sheets.

1. Use the **pddisable** command to disable the BSD actual destination.
2. Issue this **pdset** command:

```
pdset -cp -x "auxiliary-sheet-selections-supported=none
             job-attribute-supported+=auxiliary-sheet-selection" bsdad
```

   where `bsdad` is the BSD actual destination on which you are working.

3. Determine the queue associated with the destination, and then issue this **pdset** command to unregister the actual destination from its queue:

```
psdet -cp -x "associated-queue==" bsdad
```

4. Issue this **pdset** command to re-register the actual destination to its queue:

```
psdet -cp -x "associated-queue=bsdq" bsdad
```

   where bsdq is the queue identified in the previous step. This command tells the queue about the new actual destination attribute settings so it can schedule jobs correctly.

5. Use the **pdenable** command to enable the BSD actual destination.

Note

You could modify this procedure to specify other values for **auxiliary-sheet-selection** since the BSD DSS ignores these values as well.

Gathering accounting and auditing data about print jobs

InfoPrint allows you to gather accounting information about print jobs from two sources:

- The InfoPrint Manager Linux server that applies to all device support systems (DSS). For more information, see Working with InfoPrint Manager server accounting information, p. 95.
  You should use accounting information from the InfoPrint Manager if you:
  - need to track accounting information for all types of InfoPrint DSSs.
  - want to take advantage of the **pdaccount** command to summarize data in a spread sheet program.
- The PSF auditing and accounting user-exit programs (applies only to PSF). For more information, see Working with PSF accounting, post-print accounting, and auditing user-exit programs, p. 103.

You should use accounting information from the PSF auditing and accounting user-exit programs if you:

– use only PSF attached printers.
– need more detailed information than that provided by the **pdaccount** command.

Working with InfoPrint Manager server accounting information

While all DSSs can store information about jobs processed, some DSSs are more accurate than others. All actual destination accounting logs are enabled through the server and actual destination **log-accounting-data** attribute. When you specify `log-accounting-data=true`, InfoPrint stores data into server accounting logs (one for each destination), which you can either view directly from the command-line or through the **pdaccount** command. The **pdaccount** command lets you create views of summary information in a comma-delimited format (`Printer1,12997,1989787763`). This information can be saved in a file, which can be imported into a spread sheet to calculate the cost of consumables, the usage rate of printers, or the cost of printing by individual user. The summary information includes both the total pages and total octets (the number of bytes) printed, either by destination or by user as requested for a specified time period. You can also obtain the complete accounting records for a specified time period, instead of the summary data.

How does the pdaccount command collect InfoPrint Manager server accounting information?

To activate InfoPrint Manager server account log support, you must change the `log-accounting-data` value. By default, the InfoPrint Manager value is `log-accounting-data=false`, and the actual destination default is to use the server setting. As a result, you must set the **log-accounting-data** attribute as follows:

- To activate accounting for all destinations on a server:
  Set the server's `log-accounting-data` attribute to `True` (if an actual destination hasn't specified this attribute, the server's `log-accounting-data` attribute is used)
- To activate accounting for a specific actual destination:
  Set the `log-accounting-data` attribute to `True` for that specific actual destination. Otherwise, the value of the `log-accounting-data` attribute for the server is used.
- To deactivate accounting for all destinations on a server:
  Set the `log-accounting-data` attribute to `False` (if an actual destination hasn't specified this attribute, the server's `log-accounting-data` attribute is used).
- To deactivate accounting for a specific actual destination:
  Set the `log-accounting-data` attribute to `False` for that specific actual destination. Otherwise, the value of the `log-accounting-data` attribute for the server is used.

You can change the `log-accounting-data` attribute with a `pdset` command or the InfoPrint Manager Administration GUI. For more information about the InfoPrint Manager Administration GUI, see *RICOH InfoPrint Manager for Linux: Getting Started*.

By default, the `pdaccount` command obtains this accounting information from one or more InfoPrint Manager servers:

- **actual destination name**
- **job owner**
- **global job-id**
- **submission time**
- **completion time**
- **pages-completed**
- **octets-completed**
- **data stream**
- **job name**

To use the accounting log version, you must set the **accounting-log-version** value. By default, the **accounting-log-version** value is not set. To use the same **accounting-log-version** value for all actual destinations on a server, set the server's **accounting-log-version** attribute to the desired text value and make sure that the **accounting-log-version** attribute is not set on the actual destinations.

The **accounting-log-version** attribute is logged only if it has a value. By setting the **accounting-log-version** attribute, a new field is appended after the currently fixed set of logged attributes which shifts the previous layout of additional accounting data.

The actual destinations attribute **additional-accounting-log-attributes** contains the job and document attributes that you define to record in the accounting log for each destination. You can use the actual destination attribute **additional-accounting-log-attributes** to define additional job and document attributes that you want included in the log. These attributes are:

- **accounting1-text**
- **accounting2-text**
- **accounting3-text**
- **account-text**
- **actual-destinations-requested**
- **address1-text**
- **address2-text**
- **address3-text**
- **address4-text**
- **auxiliary-sheet-selection**
- **base-printer**
- **building-text callback-number**
- **chars**
- **color-bits-per-plane**
- **color-mapping-table**
- **color-profile**
- **color-rendering-intent**
- **color-toner-saver**
- **copy-count**
- **creation-time**
- **default-input-tray**

- default-medium
- department-text
- destination-accounting1-text
- destination-accounting2-text
- destination-accounting3-text
- destination-company-text
- destination-locations-requested
- destination-models-requested
- destination-name-requested
- destination-pass-through
- destinations-used
- document-comment
- document-content
- document-file-name
- document-finishing
- document-type
- email-from-address
- email-to-address
- font-resolution
- form-definition
- halftone input-tray-select
- job-batch
- job-client-id
- job-comment
- job-copies-completed
- job-deadline-time
- job-discard-time
- job-finishing
- job-media-sheet-count
- job-originator
- job-page-count
- job-priority
- job-retain-until
- job-retention-period
- media-sheets-completed
- modification-time
- mvs-destination
- mvs-forms
- mvs-segment-id
- name-of-last-accessor
- name-text
- node-id-text

- **number-of-documents**
- **number-up**
- **octet-count**
- **originating-company-text**
- **output-format**
- **page-count**
- **page-definition**
- **page-select**
- **plex**
- **print-quality**
- **processing-time**
- **programmer-text**
- **promotion-time**
- **queue-assigned**
- **record-count**
- **resource-context**
- **results-profile**
- **room-text**
- **sheet-range**
- **sides**
- **started-printing-time**
- **subject-text**
- **title-text**
- **user-id-text**
- **user-locale**
- **user-name**
- **x-image-shift**
- **x-image-shift-back**
- **y-image-shift**
- **y-image-shift-back**

Note

You only get the **accounting-log-version** attribute if you specify `pdaccount –t all`.

The user-defined accounting information is added at the end of the current accounting information log. Only the attributes for the first printable document of the job are logged.

If you wanted to request summary information group by destinations on an InfoPrint Manager server named `Server1` from 8 a.m. on 9/20/99 to 8 a.m. on 9/27/99, specify this command from a command line:

```
pdaccount  -t destination  -s '08:00:00 09/20/99'  -e '08:00:00 09/27/99'
Server1
```

InfoPrint Manager displays the information in a format similar to this:

```
Destination Name,Pages Completed,Octets Completed
```

```
Printer1,12997,1989787763
Printer2,2455,17676836
Printer3,86673,189808083
```

where `Printer1` specifies the actual destination, `12997` indicates the pages completed, and `1989787763` indicates the octets completed in the specified week.

For more information about using the `pdaccount` command, see *RICOH InfoPrint Manager: Reference* or the **pdaccount** man page.

The various DSSs for actual destinations report `pages-completed` differently. For example, actual destinations created using the PSF DSS report the number of Intelligent Printer Data Stream (IPDS) pages printed, based upon the number of `begin-page` and `end-page` structured field pairs in the data. As a result, if you print a 2–up document as duplex, you could get a page count of 4 for a single sheet of paper. However, if you created an actual destination through the CUPS DSS, it counts each side-sheet printed as a single page. As a result, if you print a 2-up document as duplex, you will get a page count of 2 for each sheet printed out.

How do you manage the contents of the InfoPrint server accounting logs?

Once you specify `log-accounting-data=true` on either an InfoPrint server or an actual destination, these accounting logs start growing without bound. The accounting logs must be managed carefully to avoid potential performance problems. Periodically, the log files must be either purged or moved to a different directory. If you do not do either of these tasks, your installation will run out of space on the file system, preventing you from printing any more jobs. To avoid this problem, you should purge or move the accounting and audit logs as part of your system maintenance.

The accounting logs reside in a subdirectory of the server's directory. This is a description of the names and location of the files, as well as their default permissions:

```
$PDBASE/ servername/accounting.logs
```

where `servername` indicates the name of your InfoPrint server. InfoPrint provides two accounting logs for each destination:

- `actual_destination_name.accounting.log.v1` storing accounting information for successful jobs
- `actual_destination_name.accounting.log.error` storing accounting information for error jobs

where `actual_destination_name` specifies the actual destination where the job is processed. Locating them in a separate directory makes them easier for you to manage.

InfoPrint restricts direct access to the logs by setting the permissions on the `accounting.log` directory to `rwx` (read, write, execute) for `root` and the system group. The logs themselves will be created with root having read and write access and the system group having read access.

To retain a record of InfoPrint server accounting data, you need to move the existing log files to a new directory. Use this procedure:

1. Go to the directory where the log files are located by specifying:

```
cd $PDBASE/server1/accounting.logs
```

2. Move the **prt1.accounting.log.*** files to a directory named **1qrt2000** by specifying:

```
mv prt1.accounting.log.* /1qrt2000
```

This command moves the existing files to the new directory and removes them from the **$PDBASE/server1/accounting.logs** directory. When the next job is processed on **prt1**, InfoPrint creates a new accounting log file.

We recommend you move the data to another file system rather than deleting it from the system. This lets you store accounting data and delete it from the system when you no longer need to refer to it. To delete the files without storing the data in another directory,

1. Specify

```
cd $PDBASE/server1/accounting.logs
```

This command moves you into the correct path.

2. Enter

```
rm prt1.accounting.log.*
```

When the command-line prompt returns, you know that the files have been deleted from the system.

3. Enter

```
ls
```

This command displays the contents of your directory for you to verify that the files have been deleted. When the next job is submitted for printing, InfoPrint creates a new accounting log file and writes the data into that file.

The `pdaccount` command only finds and processes the information in the original files. Once you move or rename the files, you cannot use the `pdaccount` command to summarize the data. You might want to use the `pdaccount` command to find and process the information and summarize it, store the summaries in a different location, and then delete the logs.

Important

Requesting information for **all** or **error** accounting records can consume a lot of the server's memory since all the data is read into memory to be sent to you. If you have a lot of data to retrieve, you should consider requesting it for smaller time periods and concatenating the returned data.

For example, to retrieve all the accounting records for a one month period, you could issue four **pdaccount** commands like this:

- `pdaccount -t all -s '00:00:00 AM 03/01/00' -e '00:00:00 AM 03/08/00' Server A >March1.accting.data`
- `pdaccount -t all -s '00:00:00 AM 03/08/00' -e '00:00:00 AM 03/15/00' Server A >March2.accting.data`
- `pdaccount -t all -s '00:00:00 AM 03/15/00' -e '00:00:00 AM 03/22/00' Server A >March3.accting.data`
- `pdaccount -t all -s '00:00:00 AM 03/22/00' -e '00:00:00 AM 03/31/00' Server A >March4.accting.data`

Usage considerations by DSS for accounting information

This section contains more information about server accounting information by DSS. It also gives suggestions for working with the various DSSs when using the accounting information in InfoPrint Manager.

BSD DSS

The page count recorded in the accounting log for destinations using the BSD DSS is based on these criteria:

- If the user specifies a **job-page-count** job attribute or a **page-count** document attribute with the print job, InfoPrint Manager records that number in the accounting log.
- If the user has not specified **job-page-count** or **page-count**, and the data being printed is PCL, PDF, or PostScript, InfoPrint Manager attempts to calculate the page count and records the calculated page count in the accounting log.

CUPS DSS

- If **wait-for-job-completion** is set to **True** and the job completes successfully or is canceled, then the page count shown in the accounting log is the number of pages stacked at the printer. For example, a three-page document printed simplex records three pages in the accounting log and three pages are stacked; a four-page duplex document records four pages in the accounting log even though only two sheets are stacked. Depending on the printer and the back end used, jobs held at the printer log zero pages completed.

  Note

  – When you use CUPS DSS, make sure that the cups-libs version is 1.4.2 or higher for accurate accounting information.

  – If your physical printer has a time-out setting, make sure it is disabled or the time-out value is high enough when you specify **True** or **Yes** for **wait-for-job-completion**. A low time-out value causes incorrect information and the back end fails.

  – If your physical printer does not support nested PJL JOB commands, you must configure the actual destination **printer-command** attribute. For more information, see **wait-for-job-completion**.

- If **wait-for-job-completion** is set to **False**, the page count is not based on the number of pages stacked on the printer. In this case, InfoPrint Manager uses these methods to calculate a page count:

  1. The printer backend can return a non-zero page count that is based on the number of pages it is sending to the printer. If this number is returned, InfoPrint Manager uses it for the page count that is reported in the accounting log.
  2. If the printer back end returns a zero for the page count and if the user has specified a **job-page-count** with the job, InfoPrint Manager reports that page count in the accounting log.
  3. If the printer back end returns a zero for the page count and the user has not specified a **job-page-count** and the job is PCL, PDF or PostScript, then InfoPrint Manager tries to calculate a page count and records that calculated page count in the accounting log.

- **Limitation:** The octet count reported for canceled or paused jobs might not be accurate.

DFE DSS

- If the **wait-for-job-completion** attribute is set to **True** and the job completes successfully or is canceled, then the page count shown in the accounting log is the number of pages stacked at the printer. For example, a three-page simplex document records three pages in the accounting log and three pages are stacked; a four-page duplex document records four pages in the accounting log even though only two sheets are stacked.
- If the **wait-for-job-completion** attribute is set to **False**, the page count is not based on the number of pages stacked on the printer. In this case, InfoPrint Manager uses these methods to calculate a page count:

  1. The page count recorded in the accounting log is the number of pages PSF reports it has sent to the printer driver, except for the case when the **document-formats-ripped-at-destination** actual destination attribute is set.
  2. If the **document-formats-ripped-at-destination** actual destination attribute is set to the input datastream and the job is PCL, PDF or PostScript, then InfoPrint Manager attempts to calculate a page count and records that calculated page count in the accounting log. The only exception occurs when the user specifies a **job-page-count** attribute and InfoPrint Manager records the user-specified value of that attribute.

- **Limitations:** The page count and octet count reported in the accounting log for canceled jobs is not accurate.

Email DSS

The page count recorded in the accounting log for destinations using the Email DSS reflects the number of pages generated and sent to the email recipient by InfoPrint Manager. It is not based on the number of pages received by the email recipient.

IPP DSS

The page count recorded in the accounting log for destinations using the IPP DSS is based on these criteria:

- Depending on the capabilities of your particular IPP printer, if the printer reports **pages-completed** information through the Internet Printing Protocol and the job completes successfully, that information is used. The auxiliary sheet is not included in the count.
- If the user specifies a **job-page-count** job attribute or a **page-count** document attribute with the print job, InfoPrint Manager records that number in the accounting log. The auxiliary sheet is not added to the count.
- If the user has not specified **job-page-count** or **page-count**, and the data being printed is PCL, PDF, or PostScript, InfoPrint Manager retrieves the page count from the printer, if the particular printer supports that function. The auxiliary sheet is not included in the count.
- **Limitations:** Because all information is obtained through the Internet Printer Protocol, the attributes required for **pages-completed** might not be supported by a particular printer model.

Note

For accurate results with this DSS, make sure that you set the sides default (whether `simplex` or `duplex`) in the printer to match the value on the **sides** actual destination attribute.

PSF DSS with Command attachment type

The page count recorded in the accounting log for destinations using the command is the number of pages PSF generates and sends to the printer. It is not based on the number of pages received or printed by the printer. The page count does include start pages, separator pages, and end pages.

PSF DSS with Other-Driver attachment type

- If **wait-for-job-completion** is set to **True** and the job completes successfully, then the page count shown in the accounting log is the number of pages stacked at the printer. For example, a three-page simplex document records three pages in the accounting log and three pages stacked; a four-page simplex or duplex document reports four pages in the accounting log even though only 2 sheets are stacked.

  The page count recorded in the accounting includes the auxiliary sheets.

  A duplex four-page job with a header sheet provides a page count of five, because the auxiliary sheets are included in the count and both sides of each page are counted.
- If **wait-for-job-completion** is set to **False**, the page count is not based on the number of pages stacked on the printer. In this case, InfoPrint Manager uses these methods to calculate a page count:

  1. The page count recorded in the accounting log is the number of pages PSF reports it has sent to the printer driver, except for the case when the **document-formats-ripped-at-destination** actual destination attribute is set.
  2. If the **document-formats-ripped-at-destination** actual destination attribute is set to the input datastream and the job is PCL, PDF or PostScript, then InfoPrint Manager attempts to calculate a page count and records that calculated page count in the accounting log. The only exception occurs when the user specifies a **job-page-count** attribute and InfoPrint Manager records the user-specified value of that attribute.
- **Limitations:** The page count and octet count reported in the accounting log for canceled and paused jobs is not accurate.

PSF DSS with TCP/IP attachment type

- On jobs that complete successfully, the page count shown in the accounting log is the number of AFP pages stacked on the printer, including start pages, separator pages, and end pages.

  Note

  For n-up documents, up to *n* pages are counted for each sheet-side that prints.
- **Limitations:**
  - When printing a multiple-document job in which all the documents are ASCII, the accounting log only shows pages for the last document printed.
  - If a job is paused or canceled, the page count shown in the accounting log might not be accurate.
  - On paused jobs, the octet count is always zero (0).

Working with PSF accounting, post-print accounting, and auditing user-exit programs

The accounting, post-print accounting, and audit PSF DSS user-exit programs record data about jobs. For each job printed, InfoPrint Manager activates the accounting and audit user-exit programs *before* processing the print job's trailer page. InfoPrint Manager activates the post-print accounting user-exit

program *when* the job stacks. More than one InfoPrint Manager actual destination can use the same user-exit program to write into the same log file.

What do these user-exit programs provide?

The accounting and auditing user-exit programs record statistical data on jobs processed by InfoPrint actual destinations. PSF DSS accounting, post-print, and audit user-exit programs, p. 104 shows the most common accounting and auditing user-exit programs, the files where they record data, and the types of data that they record.

PSF DSS accounting, post-print, and audit user-exit programs

Program	*Log File*	*Records*
/usr/lpp/psf/bin/ainuxaccp	/var/psf/podaccount.log	Job submitter, job ID, actual destination, pages per job, pages per actual destination, pages per output bin, data object resources, start date, start time, copies per job
/usr/lpp/psf/bin/ainacclog	/var/psf/accounting.log	Job submitter, job name, actual destination, pages per job, pages per actual destination, jobs per output bin, data object resources, start date, start time, fonts, resident fonts, overlays
/usr/lpp/psf/bin/ ainuxjobcompletion	/var/psf/ jobcompletion.log	Job submitter, job ID, job name, spool ID, actual destination, global ID, user pages stacked by tray, user sheets stacked by tray, actual pages stacked by tray, actual sheets stacked by tray, copies requested, fonts and resident fonts used, overlays used, hard and soft segments used, processing start and stop date/ time, job completion date/time, file date/time, extended user ID, extended job ID, address, account, building, department, programmer, room, title, job ticket name

Program	Log File	Records
/usr/lpp/psf/bin/ainaudlog	/var/psf/audit.log	Job submitter, job name, actual destination, pages per job, pages per actual destination, jobs per output bin, data object resources, start date, start time, fonts, resident fonts, overlays

Note

`/usr/lpp/psf/bin/ainacclog` and `/usr/lpp/psf/bin/ainaudlog` record the same information in different files.

This information is specified in flat ASCII format, with a single line for each job that is processed. You can use this information to determine billing procedures, to assess the workload of different output devices, or to determine how resources are used at your installation.

The **ainuxjobcompletion** exit is called when the printer reports back to InfoPrint Manager that all of the pages for a job have printed. This includes a partial job (due to a data stream error or because the job was paused/canceled or forward-spaced) and also any extra pages produced as a result of a backspace or other recovery. For the last job that prints on a continuous forms printer, the exit is not called until an NPRO (non-process run out) is done, either by an operator or automatically by InfoPrint Manager.

Only one record is written for a job that prints, unless a job is paused. In this case, a record is written for the part of the job that was printed up to the pause point; later, if the job is resumed another record is written for the rest of the job.

Note

If InfoPrint Manager does not receive feedback from the printer that all pages have stacked, such as when the printer is shut down in the middle of a job or when an error such as a server failure occurs, no record is written.

A new procedure has been added to the InfoPrint Manager Administration GUI to show you how to activate the default PSF **ainuxjobcompletion** accounting logs. The **ainuxjobcompletion** accounting exit interface can be used for PSF-driven IPDS printers. It provides actual (versus projected) accounting data and provides additional information about completed print jobs.

Customers who upgrade an existing InfoPrint Manager server to the latest version will continue to find the previous accounting exit interface intact. Accounting programs that count on the previous accounting interface, which include the InfoPrint supplied samples, will function without modification.

Customers installing an InfoPrint Manager version for the first time will see the post-print accounting exit activated and the accounting data written to `/var/psf/jobcompletion.log`.

The intent is to minimize upgrade issues for existing customers while ensuring that new customers begin with the preferred accounting exit. The default installation for new machines will not activate both accounting exits. However, when the latest version is installed, both new and upgraded customers can choose to have the old, preferred, or both accounting exits active.

If an upgraded installation wants to activate the preferred (new) exit, the administrator must set the **psf-post-print-accounting-program-name** auxiliary sheet attribute in the appropriate auxiliary sheet object to provide the name of either the InfoPrint supplied sample exit or a custom version of the

preferred exit. The administrator can also choose to deactivate the old exit by resetting the value of the **psf-exit-program-name** auxiliary attribute in the appropriate auxiliary sheet object to **<not set>**.

If a new installation wants to deactivate the preferred exit and instead run the old exit, the administrator must reset the **psf-post-print-accounting-program-name** auxiliary sheet attribute in the appropriate auxiliary sheet object to **<not set>**. The administrator can also choose to activate the old exit by setting the value of the **psf-post-print-accounting-program-name** auxiliary sheet attribute in the appropriate sheet object to the name of either the InfoPrint supplied sample exit or a custom version of the old exit.

In both cases, the administrator can also choose to have both exits active by setting the **psf-exit-program-name** and the **psf-post-print-accounting-program-name** auxiliary sheet attributes to the names of the two accounting programs (old and preferred). The administrator can also choose to deactivate both exits by leaving both attributes unset.

How do you format PSF DSS accounting, post-print accounting, and audit data for viewing?

To view PSF DSS accounting and audit data, you can use one of these executable reporting utilities that are provided with InfoPrint:

ainupod1

Reports accounting data stored in `podaccount.log`, sorted by job submitter.

ainupod2

Reports accounting data stored in `podaccount.log`, sorted by job ID.

ainupod3

Reports times and pages printed by the job submitter from `podaccount.log`.

ainurpt1

Reports accounting data stored in `accounting.log`, sorted by actual destination.

ainurpt2

Reports accounting data stored in `accounting.log`, sorted by job submitter.

ainurpt3

Reports times and pages printed by the job submitter from `accounting.log`.

ainurpt4

Reports audit data stored in `audit.log`, sorted by actual destination.

ainurpt5

Reports audit data stored in `audit.log`, sorted by job submitter.

ainurpt6

Reports times and pages printed by the job submitter from `audit.log`.

ainurpt7

Reports post-printing accounting data stored in **jobcompletion.log**, sorted by actual destination.

ainurpt8

Reports summary post-printing accounting data stored in **jobcompletion.log**, sorted by job submitter.

ainurpt9

Reports detailed entries of times and pages printed for a specific job submitter in **jobcompletion.log**.

For examples using these utilities, see *RICOH InfoPrint Manager: Reference* or the man pages for the utilities.

How do you manage the contents of the PSF DSS accounting data file?

An installation with a regular schedule of printing jobs will accumulate accounting and audit data that must be managed carefully to avoid potential performance problems. Periodically, the log files must be either purged or moved to a different directory. If you do not do either of these tasks, your installation will run out of space on the file system, preventing you from printing any more jobs. To avoid this problem, you should purge or move the accounting and audit logs as part of your system maintenance.

To retain a record of PSF DSS accounting and audit data, you need to move the existing log files to a new directory. Use this procedure:

1. Go to the directory where the log file, for example, **podaccount.log**, is located by specifying:

```
cd /var/psf
```

2. Move the **podaccount.log** file to a directory named **1quarter96** by specifying:

```
mv podaccount.log /1quarter96
```

This command moves the existing file to the new directory and removes it from the **/var/psf** directory. The next time a job is submitted for printing from InfoPrint Submit, InfoPrint will create a new **podaccount.log** file in the **/var/psf** directory and write a single line of data for each job.

We recommend you move the data to another file system rather than deleting it from the system. This lets you store accounting data and delete it from the system when you no longer need to refer to it. If you want to delete the **podaccount.log** file without storing the data in another directory,

1. Specify

```
cd /var/psf
```

This command moves you into the correct path.

2. Enter

```
rm podaccount.log
```

When the command-line prompt returns, you know that the **podaccount.log** file has been deleted from the system.

3. Enter

```
ls
```

This command displays the contents of your directory for you to verify that the file has been deleted. When the next job is submitted for printing, InfoPrint creates a new **podaccount.log** file and writes the data into that file.

Creating and managing resource-context objects

Resource-context objects identify the locations of different types of Advanced Function Presentation (AFP) resources that jobs sent to PSF physical printers can reference. Resource-context objects let you encapsulate the directory path name in one place. Then you specify the resource-context object name rather than directory names with documents and actual destinations when specifying where to find resources. Later, if the location of a resource changes, you only need to modify the path name specified in the corresponding resource-context object.

The different types of AFP resources are:

Color management resources (CMRs)

CMRs are the foundation of color management in AFP print systems. They provide all the color management information, such as ICC profiles and halftones, that an AFP system needs to process a print job and maintain consistent color from one device to another.

Note

1. CMRs are stored in resource libraries that use resource access tables (RATs). When you update an existing CMR or add a new CMR to a resource library, the RAT is updated. You must restart PSF so the changes take effect.
2. CMRs are considered data objects for accounting purposes.
3. For retention of resources across print jobs, CMRs are counted as part of the **maximum-presentation-object-containers-to-keep**.

Data object font resources

A data object font resource can be an OpenType font collection, an OpenType set of linked fonts, or OpenType fonts that are installed in one or more directories on your InfoPrint Manager server. Data object font resources are the only fonts registered through a RAT that are located in the same directory as the fonts. For more information about using data object font resources with InfoPrint Manager, see Using OpenType Fonts, p. 335.

Data objects

Data objects can be IOCA files or types of files natively supported by a particular model of printer (for example, an EPS file on an InfoPrint 5000). In addition, if you use the transforms that come with InfoPrint Manager to convert objects into AFP first, these non-native types of files can be used as data objects. In InfoPrint Manager, data objects are grouped with BCOCA and GOCA objects, and they are collectively known as *presentation object containers*.

Note

Data objects are stored in resource libraries that use RATs. When you update an existing data object or add a new data object to a resource library, the RAT is updated. You must restart PSF so the changes take effect.

Fonts

A font is a single size and typeface in a particular type family, including letters, numerals, punctuation marks, special characters, and ligatures.

Form definitions

Form definitions provide instructions on how output devices position data on the page. Form definitions can specify overlays, a paper source for cut-sheet printer devices, duplexed printing, text suppression, data position, and the number and modifications of pages.

Page definitions

Page definitions contain the formatting controls for line data. Page definitions can include controls for the number of lines per logical page, font selection, print direction, and the mapping of individual fields to positions on the logical page.

Page segments

Page segments contain text and images that you can include at any addressable point on a page or an electronic overlay.

Overlays

Overlays are collections of predefined data such as lines, shading, text boxes, or logos, that can merge with variable data on a page or a form during printing or transmission.

You use the **pdcreate** command to identify the directory or directories where the AFP resources reside and provide a name for the resource-context object. You use the **AFP Resources** tab of the actual destination and default document attributes notebooks in the InfoPrint Manager Administration GUI, or the **Command Prompt** window from an InfoPrint Manager for Windows server, to identify the resource-context object name or which paths you want to associate with that object. Jobs can also reference resource-context objects through their contained documents. See The search order for AFP resources, p. 109 for the search order InfoPrint uses to locate resource-context objects for a job. Both the actual destination and the resource-context objects that are referenced must reside in the same InfoPrint server.

You can also purchase additional resources or even create your own custom resources. To create page definitions and form definitions, you can purchase the Page Printer Formatting Aid (PPFA) for Windows, which is an optional feature of InfoPrint Manager for Windows.

The search order for AFP resources

This section describes the order and the conditions on which InfoPrint searches for AFP resources:

1. For fonts, paths specified by the `-osrchfontlib` keyword on the `enq`, `lp`, `qprt`, or `lprafp` command
2. Paths specified by the `-oreslib` keyword on the `enq`, `lp`, `qprt`, or `lprafp` command
3. The path specified by the **resource-context-user** document attribute.
4. The paths specified by one of these document attributes, depending on the type of resource:
   - **resource-context-font**
   - **resource-context-form-definition**
   - **resource-context-overlay**
   - **resource-context-page-definition**
   - **resource-context-page-segment**

Note

When one of these document attributes contains a value, InfoPrint ignores any location specified by the same attribute of a default document object if one is used with the job.

5. The path specified by the **resource-context** document attribute.
6. The path specified by the **PSFPATH** environment variable.
   If you are using OpenType Fonts, you must allow the InfoPrint Fonts product to install its resource access table for keeping track of these fonts in the default directory. On an InfoPrint Manager Windows server, if you change the location of this table from the folder where it was installed, you can update the global search path on your Windows system with the global resource path for these fonts.
7. The paths specified by the resource-context attributes of the actual destination.
8. For fonts, the directory where your fonts are installed.

Once a resource is located, the search ends, and the located resource is used to process the job, as long as the user can access the resource. If InfoPrint Manager ultimately cannot find a required resource, it prints an error message advising you of the missing resource. InfoPrint Manager reports that the job completed successfully.

If the only AFP resources you use are the AFP resources that shipped with InfoPrint Manager, there is no need to use resource-context objects. All of your resources are located automatically because these resources reside in the **reslib**, **fontlib**, and font directories described above. If, on the other hand, you have custom AFP resources, make sure that you *do not* store them in the **reslib**, **fontlib**, or font directories listed. Any custom resources stored in these locations might be corrupted by installing, uninstalling, or servicing the software. Store any custom AFP resources in a directory outside these locations and use resource-context objects to provide access to them.

When InfoPrint Manager puts together the total of all resource contexts for a particular resource type, InfoPrint Manager tries this list of extensions in order while searching each directory for a particular resource. For example, InfoPrint Manager is searching for a 300–pel raster font named C0H20000 and these conditions are true:

- Document attributes are set to `resource-context-font=\myfonts`
- `resource-context=\all_resources`
- The **$PSFPATH** environment variable has not been set.
- The **resource-context-font** actual destination attribute has not been set.

A list of files similar to this would be checked in sequence:

1. `x:\myfonts\C0H20000.300`
2. `x:\myfonts\C0H20000.FONT300`
3. `x:\myfonts\C0H20000`
4. `x:\all_resources\C0H20000.300`
5. `x:\all_resources\C0H20000.FONT300`
6. `x:\all_resources\C0H20000`
7. With InfoPrint Manager:
   1) The `/usr/lpp/psf/reslib` directory.

2) For fonts, the `/usr/lpp/afpfonts` directory.
3) For fonts, the `/usr/lpp/psf/fontlib` directory.

Once InfoPrint Manager finds a matching file, the search ends.

If InfoPrint Manager cannot find the resource in any of the paths, it stops the job and prints an error message at the end of the job.

CMRs and data objects search order

The search order for CMRs and data objects is different from the order used for other AFP resources.

CMRs and data objects must reside in the same directory as the RAT where the resources are identified. That is, the CMR RAT and the data object RAT can be located in different directories, but the resources that are associated with the specific RAT must reside in the same directory as the RAT.

The InfoPrint 5000 printer Models AS1 and AD1/AD2 and InfoPrint 4100 printers support CMRs and data objects.

The search order for CMRs

InfoPrint Manager searches for CMRs in different locations and in a different order than it uses for other AFP resources.

InfoPrint Manager searches for CMRs in these locations in this order:

1. Inline objects
2. The path specified by the **resource-context-presentation-object-container** document attribute
3. The path specified by the **resource-context** document attribute
4. The path specified by the **PSFPATH** environment variable
5. The path specified by the **resource-context-presentation-object-container** actual destination attribute
6. The path specified by the **resource-context** actual destination attribute
7. The system dependant library:

Note

1. InfoPrint searches for CMRs using the UTF-16 name provided in the data stream.
2. If a CMR changes after a job prints, recycle the actual destination by disabling and enabling the connection.

The search order for data objects

InfoPrint Manager searches for data objects in different locations and in a different order than it uses for other AFP resources.

InfoPrint Manager searches for data objects in these locations in this order:

1. Inline objects
2. The path specified by the **resource-context-page-segment** document attribute if object is IOCA that is not installed in a data object RAT

3. The path specified by the **resource-context-presentation-object-container** document attribute
4. The path specified by the **resource-context** document attribute
5. The path specified by the **PSFPATH** environment variable
6. The path specified by the **resource-context-presentation-object-container** actual destination attribute
7. The path specified by the **resource-context** actual destination attribute
8. The system dependant library:

Note

1. InfoPrint Manager searches for data objects using the UTF-16 name and the 64-character name provided in the data stream. The search process is different from previous versions. InfoPrint Manager now searches items one through four by UTF-16 name and then search items one through four by eight byte name. Then, InfoPrint Manager searches items five through eight by UTF-16 name and then by eight byte name.
2. If a data object changes after a job prints, recycle the actual destination by disabling and enabling the connection.

Reuse of CMRs and data objects across printer jobs

Using CMRs and data objects in more than one print job can improve printer performance. InfoPrint Manager includes two actual destination attributes that you can use to specify reuse: **maximum-presentation-object-containers-to-keep** and **capture-inline-cmr-resources**.

If you use the same resources (such as company logos) in different print jobs, you can improve print performance by storing those resources on the printer.

Use the **maximum-presentation-object-containers-to-keep** actual destination attribute to specify the maximum number of CMRs and data objects to store in the memory of the printer device between jobs. The default value is 100 objects. Both CMRs and data objects are included in this count.

The printers control the actual retention of resources based on the value of the **maximum-presentation-object-containers-to-keep** actual destination attribute and the available resources downloaded to the printer.

Note

If the printer is reset, the printer memory is cleared.

If you include CMRs inline with your print jobs, you can use the **capture-inline-cmr-resources** actual destination attribute to indicate whether inline CMRs can be captured and reused by PSF TCP/IP printers. Capturing a CMR means the resource is downloaded to the printer and remains resident in the printer memory for future reuse. The default for this attribute is **false** (not active).

File extensions for resources

This table contains file extensions for specific types of resources and lists the order that the extensions are tried by InfoPrint when searching for that resource.

File extensions for resources

Type of Resource	*File Extensions Searched (see note)*
BCOCA (bar code) objects	1. No file extension 2. **OBJ** 3. **OBJECT**
Code pages	1. **ECP** 2. No file extension 3. **FONT3820** 4. **FONT38PP** 5. **CDP** 6. **FONT300**
Coded fonts	1. No file extension 2. **FONT3820** 3. **FONT38PP** 4. **CFT** 5. **FONT300**
Font character sets, 240-pel resolution	1. No file extension 2. **240** 3. **FONT3820** 4. **FONT38PP** 5. **CFT** 6. **CDP**
Font character sets, 300-pel resolution	1. No file extension 2. **300** 3. **FONT300** 4. **CFT** 5. **CDP**
Outline fonts	1. No file extension 2. **OLN** 3. **FONTOLN** 4. **CFT** 5. **CDP**

Type of Resource	***File Extensions Searched (see note)***
TrueType and OpenType fonts, CMRs, and data object resources	Entire file name, including extension; comes from the Resources Access Tables (RATs) creaetd by the AFP Resource Installer. If a CMR or a data object RAT is altered, PSF must be restarted for the changes to take effect.
Form definitions	1. No file extension 2. **FDEF3820** 3. **FDEF38PP** 4. **FDE** 5. **FIL**
GOCA (graphics) objects	1. No file extension 2. **OBJ** 3. **OBJECT**
IOCA (IO image) objects	1. No file extension 2. **OBJ** 3. **OBJECT**
MO:DCA Objects	1. No file extension 2. **OBJ** 3. **OBJECT**
Overlays	1. No file extension 2. **OVLY3820** 3. **OVLY38PP** 4. **OVL** 5. **OLY** 6. **OVR**
Page definitions	1. No file extension 2. **PDEF3820** 3. **PDEF38PP** 4. **PDE**
Page segments	1. No file extension 2. **PSEG3820** 3. **PSEG38PP** 4. **PSG**

Type of Resource	*File Extensions Searched (see note)*
	5. **PSE**
Color mapping table	1. No file extension 2. **SETUP** 3. **SET**
Presentation object container For data object resources not installed in a Resource Access Table (RAT).	1. No file extension 2. **DOR**
Setup data	1. No file extension 2. **SETUP** 3. **SET** 4. **COMSETUP**

Note

1. All file extensions must be in uppercase.
2. If a file name includes a period (.), the file extension is that part of the file name that follows the period. For example, the file extension of the file name **ARTWORK.PSEG3820** is **PSEG3820**.

How InfoPrint Manager provides resource access for PSF DSS printers

By default, InfoPrint Manager allows all users to access the resource by setting all resource permissions to **-r–r–r–**. The instance of the program that tries to access a resource is created with an id of the default user, ipm1, or an **authorized user** with **sys** group set as the primary group.

PSF DSS printers limit resource access based on the userid of the person that submitted the job. The userid must have a local ID defined, or InfoPrint uses anonymous resource access (requiring a minimum of **-r–r–r–** access permissions to access the resource).

Access for PSF DSS resources is granted if the userid:

1. Exists locally (in the local **/etc/passwd** file), and they are a member of the **system** group.
2. Owns the resource.
3. Can access the resource with group permissions.
4. Can access the resource because global access is allowed for that resource.

If all these checks fail, the userid cannot access the resource and InfoPrint Manager issues an error message indicating that the resource could not be found.

Creating a new resource-context object

You create resource-context objects from the Linux command line; you cannot create them through the InfoPrint Manager Administration GUI.

To create a new resource-context object from the command line, type a **pdcreate** command using **-c resource-context** and typing:

1. The **context-address**
2. The server name
3. The resource-context object name

For example, to create a resource-context object named `Resfont1` contained in server `Server1` that identifies `x:\resource\fonts\group1` as the **context-address** or directory containing AFP font resources, enter the command:

```
pdcreate -c resource-context -x "context-address=/resource/fonts/group1"
         Server1:Resfont1
```

Note

1. Specify a name that uniquely describes the resource-context object in the server.
   All resource-context objects in a server appear in the selection lists for the fields in the **PSF Resource** tab of the **Printer Properties** window. Two servers can each contain a resource-context object with the same name, but possibly with different values for their attributes.
2. InfoPrint uses **resource-context** object names as values for the lists associated with the fields in the actual destination attributes notebook (**Printer Properties** window) or as valid values you can enter in the fields of the default document attributes notebook. Job submitters can also use them as values for attributes of a document in a job.

For more information about the attributes used in creating resource-context objects, see "Attributes for Resource Contexts" in *RICOH InfoPrint Manager: Reference*.

Changing a directory path for a resource-context object

If you need to change the path to a resource-context object, use the **pdset** command.

For example, to change the path for the resource-context object named `Resfont1` contained in server `Server1` to `/resource/fonts/group2` as the directory containing AFP font resources, enter the command:

```
pdset -c resource-context -x "context-address=/resource/fonts/group2"
         Server1:Resfont1
```

To read more about resource-context attributes, see "Attributes for Resource Contexts" in InfoPrint Manager Reference.

Assigning resource-context objects to an actual destination

You can use either the InfoPrint Manager Administration GUI or the command line to assign resource-context objects to a PSF actual destination, or e-mail destination.

Using the command line

To assign resource-context objects to a PSF actual destination, or e-mail destination from the command line, use the `pdset` command and the `resource-context-font`, `resource-context-overlay`, `resource-context-form-definition`, `resource-context-page-definition`, or `resource-context-page-segment` attribute. For example, to associate the font resource context object `Resfont1` with the PSF actual destination `Serv1-pp` residing in server `Server1`, enter this command:

```
pdset -c destination -x "resource-context-font=Resfont1"
   Server1:Serv1-pp
```

If you have not created a resource-context object, you can assign the path name of a directory containing resources to the actual destination in the same way. For example, to associate the directories `/resource/overlays/julia` and `/resource/overlays/henry`, which contain overlays, with the actual destination `Serv1-AD` residing in server `Server1`, enter this command:

```
pdset -c destination -x
 "resource-context-overlay=/resource/overlays/julia:/resource
      /overlays/henry"
   Server1:Serv1-AD
```

InfoPrint Manager searches the directories specified by the resource-context objects or the directories you specified for resources required by jobs.

Assigning a resource-context object to a default-document object

You can use either the InfoPrint Manager Administration GUI or the command line to assign resource-context objects to a default-document object.

Using the command line

To assign resource-context objects to a default-document object from the command line, use the `pdset` command and the `resource-context-font`, `resource-context-overlay`, `resource-context-form-definition`, `resource-context-page-definition`, `resource-context-page-segment`, or `resource-context` attribute. For example, to associate the font resource context object `Resfont1` with the default document `report` residing in server `Server1`, enter this command:

```
pdset -c initial-value-document -x "resource-context-font=Resfont1"
   Server1:report
```

If you have not created a resource-context object, you can assign the path name of a directory containing resources to the default document in the same way. For example, you want InfoPrint to

search for resources of any type that it cannot find in the other resource contexts associated with a default document in the directory `/resource/any`. To associate this directory with the default document `report` residing in server `Server1`, enter this command:

```
pdset -c initial-value-document -x "resource-context=/resource/any"
   Server1:report
```


You can now assign the default-document object to an actual destination (use the InfoPrint Manager Administration GUI) or inform the job submitters of the resource objects assigned to this default-document object.

Using InfoPrint Manager notifications

InfoPrint Manager can detect events (such as **job completed** or **destination needs attention**) as they occur, generate messages about those events, and deliver the messages to users who need the information. For example, users can receive messages about jobs they submit, operators can be notified about InfoPrint objects that require their attention, and administrators can be notified if someone makes a configuration change. With notifications, you do not have to query for status of the objects in your system—you just receive messages when specific events occur so you can take care of them.

You can tailor who receives these messages and how they are delivered based on the **notification profile** that is set for a particular InfoPrint object.

Understanding notification profiles

Most InfoPrint objects (the server, actual destinations, logical destinations, queues, jobs, and default jobs) have **notification profiles** associated with them. A notification profile is composed of one or more *notification profile entries*. Each notification profile entry has three main components:

Event Identifiers

Which events cause messages to be sent. See Event identifiers, p. 119 for more information.

Delivery method

How the messages are delivered. See Delivery method and delivery address, p. 119 for more information.

Delivery address

Where the messages are sent. See Delivery method and delivery address, p. 119 for more information.

In addition, a notification profile entry can include a comment (text that InfoPrint will add to the general message text) and a locale (to set the language that the message will be sent in). Comments are optional. If you do not specify a locale, InfoPrint delivers messages using the locale (language) that the user is running in.

Some objects need only one entry in their notification profiles. For example, when a person submits a job, he is probably the only one who needs to receive information about it, and he will probably be content to have all of the messages about that job delivered the same way.

For other objects, there are very good reasons to have multiple entries. For example:

- If operators manage printers from more than one workstation, you can create an entry for each workstation. That way, all of the messages go to all of the machines.
- If operators can address some messages (like **destination-needs-attention**), but an administrator needs to address others (like **object-deleted**), you can create an entry for each operator and one for the administrator. Messages are delivered to the person who can handle them.
- If an administrator likes to use InfoPrint Manager Notifications while he's at work, and wants to keep a more permanent record of how often some events occur, he can set up two entries. One delivers messages to the InfoPrint Manager Notifications client and one writes certain messages into a file that he can check periodically.

Event identifiers

The **event identifiers** component lists the InfoPrint events that trigger notifications. You can find a full list of InfoPrint notification events and their descriptions in the "Notification events and event classes" chapter of the *RICOH InfoPrint Manager: Reference*. In the descriptions, you will see what objects each event supports. If you add an event to an object that it does not support, no messages will ever be generated for that event. For example, you can add the **queue-backlogged** event to a notification profile entry for an actual destination, but **queue-backlogged** only supports the queue, job, and server objects. As a result, the actual destination notification profile will not generate any messages about that event.

That chapter also lists the InfoPrint Manager *event classes*, or groups of related notification events. When you add event identifiers to a notification profile, you can add individual events, event classes, or a combination of the two.

Each type of InfoPrint object has a default set of event identifiers. See Default notification profiles, p. 121 for a list of defaults for each object.

Delivery method and delivery address

The **delivery method** component shows how you want notification messages to be delivered. The **delivery method** and **delivery address** components are strongly related; the delivery method you select determines the type of value you use for the delivery address.

In addition, delivery method and delivery address are the *keyed components* in notification profile entries; they are the two values that InfoPrint Manager checks when you modify a notification profile. If the delivery method and delivery address that you specify match the delivery method and delivery address of an existing entry, InfoPrint Manager updates that entry. If both values do not match any of the existing entries, InfoPrint Manager adds a new entry to the profile.

There are various delivery methods defined for InfoPrint objects. All methods are valid for all objects **except** for **job-log**; **job-log** is only valid for jobs. InfoPrint notification delivery method characteristics, p. 120 lists the delivery methods, along with descriptions and how to set the delivery address for each type.

Note

One other delivery method **sapcb** exists, but it is only used internally with the InfoPrint Manager SAP Print Feature. You should never use either of those values in a notification profile entry.

InfoPrint notification delivery method characteristics

Delivery Method	Description	Delivery address
message Message is the default delivery method for all objects.	Sends a message to the InfoPrint Manager notification server, where it waits for an InfoPrint Manager Notifications client to pick it up. Once a message is picked up, it is deleted from the notification server; there are no further records of those messages.	The delivery address listed in the notification profile can be whatever you want it to be– a user id, office number, IP address, driver's license number– as long as it is unique. In order to receive messages sent to that address, you must set the delivery address listed in the **Settings** dialog in the InfoPrint Manager Notification client to the same value. The InfoPrint Manager Notifications client will only pick up messages that specify an identical delivery address.
electronic-mail (synonyms: **e-mail**, **email**)	Sends an E-mail to the address listed in the delivery address,	The E-mail address that you want notification messages to be sent to. For example, **kjones@printerco.com**.
exit	Does not actually send a message. Instead, when the event specified in the event identifier occurs, InfoPrint Manager runs an exit program or script. See Using the 'exit' delivery method, p. 121 for more information.	The full path to the notification exit program or script (with hard-coded arguments, if any).
delivery-method-wireless	Sends notifications to wireless devices, such as pagers, cellular phones, palm pilots, and laptop computers.	Because wireless notifications are delivered as emails, you must set both the **smtp-server-host** and the **smtp-server-port** server attributes on this InfoPrint server. If your InfoPrint server is behind a firewall, you might have to configure your SMTP server. For information about this task, see Configuring InfoPrint Email, p. 41.
file	Writes the message in the file you specify in the delivery address. If the file does not exist, InfoPrint Manager will create it. If the file does exist, InfoPrint Manager will overwrite it the first time it writes a message. After that, messages are added to the end of the file.	The directory path and name of the file that you want to write your notification messages in. This file must reside on the same system as your InfoPrint Manager server.

Delivery Method	Description	Delivery address
file-add-to	Adds the message to the end of the file you specify in the delivery address. If the file does not exist, InfoPrint Manager will create it.	The directory path and name of the file that you want to write your notification messages in. This file must reside on the same system as your InfoPrint Manager server.
job-log Only valid for jobs.	Writes messages into the **job-log** attribute of the job. To see the messages, enter the **pdls** command for the job-log attribute on the job you want to know about.	No value needed.
none	Does not send any notifications; turns notification off.	No value needed.

Using the 'exit' delivery method

The **exit** delivery method lets you set up automatic responses to system events beyond just sending a message. When you set the delivery method for a given event to **exit**, you use the delivery address to specify the explicit path to a script or program on your system. Whenever that event occurs, InfoPrint Manager runs the script or program you specify.

For example, if the InfoPrint Manager server runs low on disk space, the **disk-space-low** event generates a message for the system administrator. The administrator then has to do something to clean up the disk so InfoPrint Manager can continue to run effectively. However, using the **exit** delivery method can automate that process by launching a "disk clean up" script. The administrator can still receive the message, but will know that the script is already working on the problem.

By default, when you use the exit delivery method, InfoPrint Manager only passes two values along to the script or program: the message number of the event that occurred and the text of the message. As a result, the command line that the exit delivery method executes is composed of the path that you specified in the delivery address, the last three digits of the message number, and the text of the message.

Therefore, if you create a notification profile entry for the **job-completed** event, setting the delivery method to `exit` and the delivery address to `/infoprint/exits/myscript`, when a job finishes printing, InfoPrint Manager will generate and execute a command line similar to this one:

```
/infoprint/exits/myscript 280 "5010-280 Finished processing job ofc,
Job 20 (3520800004)."
```

The scripts or programs you write might require other information, such as the destination name or the printer model. To pass those values, you can include them in the delivery address of the notification profile entry; they will be added to the end of the command line when InfoPrint Manager creates it.

Default notification profiles

By default, the person who creates an object (server, destination, queue, job) receives the notification messages about that object. The default notification profile settings provide information such as object

status, error conditions, or configuration changes. Some of these events require the person managing the object to intervene; for example, someone might need to clear a paper jam.

By default, the events listed in this table generate messages for the corresponding objects. These messages are delivered to the user who created the object: job submitters receive messages about their print jobs; the administrator receives messages about destinations and servers that he or she created. The default delivery method is **message**, so the notifications are sent to the InfoPrint Manager Notification client on the system that the person used to create the object.

Default notification profile settings

Object	Events
Actual destination	destination-needs-administrator
	destination-needs-attention
	destination-needs-operator
	destination-timed-out
	object-cleaned
	object-deleted
Default Job	No defaults set. See Using notification profiles with default jobs, p. 123 for more information.
Job	destination-needs-attention
	destination-needs-operator
	destinations-not-ready-for-job
	document-aborted-by-destination
	document-aborted-by-server
	document-cancelled-at-destination
	job-aborted-by-server
	job-cancelled-by-operator
	job-cannot-be-scheduled
	job-completed
	past-discard-time
Logical destination	object-cleaned
	object-deleted
Queue	object-cleaned
	object-deleted
	queue-backlogged
Server	internal-server-error

	low-on-memory
	memory-exhausted
	object-cleaned
	object-deleted
	out-of-disk-space
	out-of-dpf-haids
	out-of-dpf-job-numbers
	out-of-dpf-raids
	sapcbd-down
	server-shutdown-complete

Determining how much notification administrators, operators, and job submitters require takes some planning and, as you familiarize yourself with InfoPrint, you should discover when you require more or less notification. You can start by using the defaults and see how many and what kind of messages you receive. If you find that you need more or less notification, or to modify which users receive notifications, or if you want messages to be sent using a different delivery method, you can change the notification profiles.

Using notification profiles with default jobs

When someone submits a print job, InfoPrint Manager sends messages about that job based on a default notification profile. As a result, the delivery method is **message**, the delivery address is "**the user ID of the person who submitted the job@the address of the system the job was submitted from**," and the event identifiers are those listed in Default notification profile settings, p. 122. (If you submit jobs using InfoPrint Select, the default notification profile is different. See the "Working with InfoPrint Select Notifications" section in *RICOH InfoPrint Manager for Linux: Getting Started* for additional information.)

If you want to change the settings in the default notification profile, you can create a notification profile on a default job. There are several things to keep in mind when you set up a notification profile on a default job:

- You can only set one default job on each logical destination. As a result, all of the jobs will use the same notification profile.
- If you set a delivery address in the notification profile for the default job, all messages will go to that address. However, if you leave the delivery address blank, InfoPrint Manager will fill it in with the address that the job was submitted from (you can't specify this address; it is sent with the job). This option only works with the **message** and **e-mail** delivery methods and is the only time that the notification profiles are merged.
- If you submit a job with an explicit notification profile (for example, by including the **notification-profile** attribute when you submit a job using **pdpr**), that profile overrides the one set in the default job.

Note

If you submit jobs using InfoPrint Select, an explicit notification profile is sent with every job. If you want to use a notification profile on a default job with InfoPrint Select,see the "Working with InfoPrint Select Notifications" section in *RICOH InfoPrint Manager for Linux: Getting Started* for information about overriding Select notifications.

Changing notification profiles

You can use the properties notebooks in the InfoPrint Manager Administration GUI to change notification profiles for jobs, default jobs, servers, queues, and actual destinations.

For jobs, default jobs, servers, queues, and actual destinations

Open the properties notebook and click the **Notification** tab for an actual destination, a job, or a default job. For a server, click the **Other** tab. On the page that opens, you can see the notification profile entries that already exist, including:

- How messages about that object are delivered
- Who they are sent to
- What events generate messages
- Any comments that are included with the messages

Use the **Add**, **Modify**, and **Delete** buttons to make changes to the notification profile. See the online help for the InfoPrint Manager Administration GUI for additional information. You can also change notification for jobs in the InfoPrint Manager Operations GUI.

Note

If you change the delivery method, make sure you change the delivery address appropriately.

For logical destinations

The only way to change a notification profile for a logical destination is to use the **pdset** command. The procedures below provide examples of some common modifications you might want to make. You can use these procedures to change the notification profiles for other InfoPrint objects as well, but you might find it easier to use the properties notebooks in the InfoPrint Manager Administration GUI.

This procedure describes how to modify notification profiles by:

- Adding or modifying events or users in a notification profile , p. 125
- Removing users from a notification profile , p. 127
- Changing the delivery method, p. 127

If you need to change other parts of a notification profile, see *RICOH InfoPrint Manager: Reference* for information about the **pdset** command and the **notification-profile** attribute.

Important

When you modify a notification profile, InfoPrint Manager compares the new delivery method and delivery address to the delivery methods and delivery addresses in the existing entries and behaves accordingly:

- If you specify both of the values and both match the values in an existing entry, the modification replaces that entry.
- If you only specify one of the values, InfoPrint Manager fills in the other value with the default values. Then, it checks to see if any of the existing entries match that combination. If they match, InfoPrint Manager replaces that entry. If they do not match, InfoPrint Manager creates a new entry.
- If you specify both values but only one matches, InfoPrint Manager creates a new entry.
- If you specify both values and neither one matches, InfoPrint Manager creates a new entry.

When you delete a notification profile entry, the delivery method and delivery address in your command must match those in an existing entry exactly.

Viewing an existing notification profile

If you want to see what values are set for a notification profile, follow these steps.

1. Type the following command. Fill in the values for the italicized items as defined below.

```
pdls -c Type of Infoprint object -r notification-profile Objectname
```

Type of InfoPrint object

What type of object you want to do this command on, not the specific object. For example, type **destination** not **printer1-ld**.

Objectname

The name of the destination or queue that this notification profile refers to.

2. Press **Enter**.

The notification profile will be displayed.

Note

If you have messages about a particular object sent to more than one person, the notification profile will display multiple entries, one for each person.

Adding or modifying events or users in a notification profile

Follow these steps to add or remove events in the notification profile for a particular delivery address, or to add another entry to the notification profile, so that messages will be sent to another address as well.

1. Type the following command. Fill in the values for the italicized items as defined below.

```
pdset -c Type of Infoprint object-x
“notification-profile+={event-identifiers= event(s) delivery-method=
method delivery-address=”address“}” Objectname
```

Type of InfoPrint object

What type of object you want to perform this command on, not the specific object. Type **destination**, **queue**, **job**, or **server**.

event-identifiers

The events that cause InfoPrint Manager to send a message. You can have messages sent for any of the events that are listed in the events-supported attribute for the server. For a list of these values, see the "Attributes for Servers" section in the "InfoPrint object attributes" chapter of *RICOH InfoPrint Manager: Reference*. Not all events support all objects. See the "Notification events and event classes" chapter in the *RICOH InfoPrint Manager: Reference* for lists of what objects each event supports.

Note

You must type all of the events for this object that you want the user you specify in the delivery address to receive, even if they already appear in the notification profile. Any events that you do not type will not be listed in the notification profile.

delivery-method

How you want messages to be sent. See InfoPrint notification delivery method characteristics, p. 120 for information about the various delivery methods.

delivery-address

Where you want to have messages sent. See InfoPrint notification delivery method characteristics, p. 120 for information about what kinds of delivery address value to use with each delivery method. Enclose the value in double quotes.

Note

You cannot specify more than one delivery address here. If you want another user to receive messages about the same (or different) events, finish this procedure, then enter the command again with that user's delivery address.

Objectname

The name of the destination or queue that this notification profile refers to.

2. Press **Enter**.
3. Repeat the process for other users and objects.

Example

The default notification profile for queues is set to send you messages when the **queue-backlogged**, **object-cleaned**, and **object-deleted** events occur. You need to change the notification profile for the print queue **printer1-q** so that you receive messages when the **queue-no-longer-backlogged** and **queue-state-changed** events occur as well. Your backup person does not currently receive any messages about queues, but needs to receive all of the same messages you do.

Issue this command to change your notification profile. Remember to list the event-identifiers for **all** of the events that you want this user to receive notifications about. This list of event-identifiers replaces the existing list entirely.

```
pdset -c queue -x “notification-profile+={event-identifiers=object-cleaned
object-deleted queue-backlogged queue-no-longer-backlogged
queue-state-changed  delivery-address=”administrator@mydesk.office.com“}”
printer1-q
```

Then, issue this command to have the **same** queue messages sent to your backup person.

```
pdset -c queue -x “notification-profile+={event-identifiers=object-cleaned
```

```
object-deleted queue-backlogged queue-no-longer-backlogged
queue-state-changed  delivery-address=”backup@hisdesk.office.com“}”
printer1-q
```

Use the Viewing an existing notification profile, p. 125 procedure to make sure the changes took effect.

Removing users from a notification profile

Follow these steps to remove an entire entry from a notification profile.

1. Use the Viewing an existing notification profile, p. 125 procedure to find the entry you want to remove.
2. Type the following command. Type the correct **delivery-method** and the **delivery-address** exactly as it is displayed for the entry that you want to remove.

```
pdset -c Type of InfoPrint object -x "notification-profile-={delivery-method=
electronic-mail or message delivery-address=“address”}"Objectname
```

3. Press **Enter**.
4. Use the Viewing an existing notification profile, p. 125 procedure to verify that the entry has been removed.
5. Repeat the process for other entries.

Example

You and two other people currently receive messages about the logical destination **printer1-ld**. You do not want to receive them any longer. You see the notification profile, and see these three entries:

```
printer1-ld: notification-profile=

{event-identifiers = object-cleaned object-deleted delivery-method =
message delivery-address = “admin@desk1.office.com” locale = en_US}

{event-identifiers = object-cleaned object-deleted delivery-method =
message delivery-address = “helpdesk@desk2.office.com” locale = en_US}

{event-identifiers = object-cleaned object-deleted delivery-method =
message delivery-address = “backup@desk3.office.com” locale = en_US}
```

Issue this command to remove your entry from the notification profile.

```
pdset -c destination -x "notification-profile-={delivery-method=message
 delivery-address = “admin@desk1.office.com”}” printer1-ld
```

Note

If you display the notification profile to verify that the entry has been removed and find that it has not been removed, enter the command again, but this time instead of only specifying the **delivery-address**, type the entire notification profile exactly as it appears.

Changing the delivery method

You can choose to receive your notification messages in a variety of ways, as listed in InfoPrint notification delivery method characteristics, p. 120. Follow these steps to change the delivery method from the default to some other value.

1. Use the Viewing an existing notification profile, p. 125 procedure to find the entry you want to change.
2. Type the following command. Fill in the values for the italicized items exactly as they appear in the entry you want to change. Replace the **delivery-method** and **delivery-address** values as specified in InfoPrint notification delivery method characteristics, p. 120.

```
pdset -c Type of InfoPrint object
-x “notification-profile+={event-identifiers=
event(s) delivery-method=method delivery-address=”address“
locale=yourlocale}”Objectname
```

3. Press **Enter**.
4. Use the Viewing an existing notification profile, p. 125 procedure to verify that the entry was added.
5. Follow the procedure for removing a user, p. 127 to remove the old entry.

Example

You want to receive your messages about **printer1-q** in your E-mail instead of through InfoPrint Manager Notifications. You view the notification profile, and see this entry:

```
printer1-q: notification-profile={event-identifiers = object-cleaned
object-deleted queue-backlogged delivery-method = message delivery-address
= “admin@desk1.office.com” locale = en_US}
```

Issue this command to add another entry to the notification profile with a different delivery method and delivery address.

```
pdset -c queue -x “notification-profile+={event-identifiers=object-cleaned
object-deleted queue-backlogged delivery-method=electronic-mail
delivery-address=”admin@mycompany.com“ locale=en_US}” printer1-q
```

Then view the notification profile, p. 125 again and see two entries:

```
printer1-q: notification-profile=  {event-identifiers = object-cleaned
object-deleted queue-backlogged delivery-method = message delivery-address =
“admin@desk1.office.com” locale = en_US}  {event-identifiers = object-cleaned
object-deleted queue-backlogged delivery-method = electronic-mail
delivery-address = “admin@mycompany.com” locale = en_US}
```

Follow the procedure for removing a user, p. 127 to delete the entry that would send messages to InfoPrint Manager Notifications.

Getting help for notification messages

If you receive a notification message that you do not understand, InfoPrint Manager provides a way to get more information about the situation. In some cases, the explanation even suggests some things you can do to resolve the problem.

Other notification methods

This section describes the other notification methods.

Notify-operator attribute

The **notify-operator** attribute is a scaled-down version of a notification profile. It can only be set on InfoPrint Manager queues and actual destinations and can only use the **e-mail**, **message**, **exit** and **wireless** delivery methods. In addition, it only sends messages when one of these events occur.

1. A job is added to a queue.
2. An actual destination starts processing a job.
3. An actual destination finishes processing a job.

The syntax of the **notify-operator** attribute is:

```
delivery-method:delivery-address
```

See InfoPrint notification delivery method characteristics, p. 120 for descriptions of the e-mail and message delivery methods and their corresponding delivery addresses. By default, the notify-operator attribute is set to **message:*your user ID@your node*** when you create an actual destination or a queue. To change the value for an actual destination, you can use the **Notification** tab of the properties notebook. To change the value for a queue, use the **pdset** command.

The messages that the notify-operator attribute delivers are not standard error messages like the ones you receive when you use a full notification profile. Instead, you write the messages that you want to send and specify them as attributes of jobs when you submit them.

Note

You must either specify those attributes in the command when you submit the job or set the text in the job and document defaults for the appropriate logical destination.

To specify your messages, you use these job attributes:

- **job-message-to-operator**
  The message that InfoPrint Manager sends when it adds a job to the queue.
- **job-start-message**
  The message that InfoPrint Manager sends when the job starts processing.
- **job-end-message**
  The message that InfoPrint Manager sends when the job finishes processing.

You can type the text for all three of these messages in the appropriate field on the **Job Notification** tab of the properties notebook for a job or a default job.

In addition, you can use the **job-start-message** attribute in conjunction with the **job-start-wait** attribute. If you set the **job-start-wait** attribute on a job (or in a default job) to **true**, the actual destination receives the job and sends the job-start message. Then, it pauses the actual destination (whether a job is currently being printed) until the operator intervenes. This function is useful when different jobs need to be printed on different types of paper. Operators do not need to manually pause the actual destination to change paper between jobs; InfoPrint Manager does it automatically. Once the printer is ready, the operator must resume the actual destination and the job prints. This function should be set for the first in a series of jobs, once you are certain that there are no jobs printing on the printer. Because it compels InfoPrint Manager to pause the printer for every job with `job-start-wait=true`, you should not specify this value on the default job.

To use the **job-start-wait** attribute, you must set it to **yes** on the job or default job. You can specify the attribute in the command when you submit a print job or set it on the **Job Notification** tab of the properties notebook for default jobs.

1

Managing security in InfoPrint Manager for Linux

InfoPrint Manager provides multiple layers of protection designed to safeguard data, authenticate users, and secure communication channels across the network. This section describes the security mechanisms available within InfoPrint Manager and explains how to configure and manage them effectively.

Managing Access Control Lists security for InfoPrint Manager for Linux

InfoPrint Manager security is a feature that you administer through the InfoPrint Manager Management Interface.

It lets you protect your printing system by associating an Access Control List (ACL) with an InfoPrint Manager object or operation. An ACL is the list of users and groups who have specific types of permission to do something to or with an object.

Note

InfoPrint Manager Security only uses the login ID permissions. Therefore, if a user logs in and does an SU to root or to another user, the permissions associated with the original login ID remain in effect.

Types of permission

In InfoPrint Manager, users can have three levels of permission: **read**, **write**, and **delete**. The levels provide these types of access:

- **Read**
  For operations, the user can do the operation. For servers and queues, the user can view the attributes. If you restrict access to a server or queue, access to all objects contained by that server or queue is automatically restricted, even if the objects are not explicitly protected. For destinations, the user can view attributes and submit jobs to that destination.

  Note

  – To access an object contained by a server or a queue, you must have at least **read** permission for the higher-level object.
- **Write**
  For all objects, the user can view and modify attributes.
- **Delete**
  For all objects, the user can view and modify attributes and can delete the object.

If you place user A on the ACL for the logical destination "print2ld" and give this user **read** permission, user A can send the print jobs to it and can open the “print2ld” object to see its properties. However, user A cannot make changes to those properties. If user A tried to change any of them or tried to delete the destination, user A would receive an error message. If you decide that user A needs to be able to

do more and give them **write** permission as well, user A will be able to change the properties of "print2ld," but still will not be able to delete it.

Important

- If you protect a destination (logical or actual) so that only certain users can modify or delete its properties, you might inadvertently prevent other users from submitting print jobs to it. To be sure that all of your users can still print to the destination, add the wildcard character (*) to the ACL as a user with read permission.

You can also attach ACLs to the operations that you can do on InfoPrint Manager objects. Allowing you to protect both operations and objects means that InfoPrint Manager Security provides different levels of security: you can protect all objects by using ACLs at the operation level or you can protect individual objects with ACLs applied only to them. Or you can do both: protect all objects by using operation-level ACLs for some operations and limit access to sub-sets of objects by using object-level ACLs.

All InfoPrint Manager object names, including security groups and ACL members, are case-sensitive.

For operations, there is only one level of permission: **read**. If a user has **read** permission, they can perform that action; if the user does not, they cannot perform the action. For example, user B is a printer operator and must be able to move jobs to different positions in the print queue because some jobs need to be printed before others. You can give user B **read** permission for the operation **Reorder Job** to allow the user to do their job. On the other hand, user C submits print jobs from their office workstation and does not like to wait for the jobs ahead of theirs in the queue to print. To prevent user C from moving jobs, do not put user C on the ACL for the **Reorder Job** operation. When user C tries to move their job to the top of the queue, the action will be denied.

When you install InfoPrint Manager, many operations are already protected so that only members of the **admin** and **oper** Security groups, p. 131 can do them. You can see the ACLs for operations in the Management Interface by selecting the **Security-ACL-Operations** item in the left pane. If you want users to be able to do those operations, you must either add those users to the individual ACLs or to a group that has permission (either the existing **admin** and **oper** groups or a new group that you create).

Note

- If an object is protected, a user can only do an operation on that object if he has both **read** permission for the operation and the appropriate level of permission for the object (for example, a job or a printer).
- If the object is not protected, only users in the default **admin** or **oper** groups are able to do the action.
- If the object is protected, the permission needed depends on the operation. For example: **List** requires **read** permission on the object, **Set** requires **write** permission, and **Delete** requires **delete** permission.
- If a user is added to a non-standard group, they are not able to do the action unless they are the owner of the job in question.

Security groups

No matter what size organization you work in, manually adding every user to every ACL can be a time-consuming process. To reduce some of the work, you can create *security groups*, groups of users who need to have the same levels of permission for the same objects. You use the name of the security group

like a user ID; instead of adding each user ID to an ACL, you add the group name. For example, if you want all ten of your print operators to be able to perform the same operations, create a group and name it **operators**. Then, add **operators** to the appropriate ACLs.

Important

- All fields, such as user IDs, group names, host names, and DNS suffixes, are case sensitive.

When you install InfoPrint Manager, three security groups are created by default:

- **acl_admin**
  Users who have authority to manage security by changing access control lists and groups. The default members are **administrator** and the user who was logged on when InfoPrint Manager was installed.
- **admin**
  Users who have administrator authority. The default members are **administrator** and the user who was logged on when InfoPrint Manager was installed.
- **oper**
  Users who have operator authority. The default member is **administrator**.

Note

1. You can modify these groups as needed. In the example above, you could have simply added your operators to the default **oper** group and modified any permissions that were not set to the level that you wanted them.
2. The default group members contain the wildcard character (*) for greater flexibility. If you do not want the Administrator user on other systems to be able to administer InfoPrint Manager, replace the * with the explicit address of the system that the InfoPrint Manager server is installed on, for example **Administrator@serversystem.company.com**. For more information about wildcarding, see Identifying FST users and groups: wildcarding, p. 133.
3. Usernames can be defined in two ways: either as **user@host** or simply as **user**. The appropriate format depends on the security model and the specific application being used. This flexibility allows granular security settings to be configured for each security group.

You can add users to multiple groups, but you cannot make one group a member of another group. For example, if you hire five new print operators, you might create a group for them called **trainees**, since you only want them to have limited permissions until they are finished with their training. When they finish their training, you cannot add **trainees** as a member of the **operators** group. You will have to add their user IDs to the operators group one at a time. In addition, you will have to either delete the **trainees** group or delete the members from it; otherwise, those users will have conflicting levels of permission.

When users are members of more than one group and each group has a different level of permission for a particular object, the most restrictive permission applies. In the example above, if you forgot to remove the new employees from the **trainees** group at the end of their training, they would not be able to perform the tasks their job required: they would still be restricted.

Identifying FST users and groups: wildcarding

When you add FST users to ACLs or Security groups, you identify them by their user ID and the computer that they work on in this format: *username@computername*. The permissions you assign will only apply when that person accesses InfoPrint Manager from that workstation.

If, however, you or any of your users want to be able to work with InfoPrint Manager objects from various workstations, you might not want to add multiple user ID/computer name combinations for the same person—you can use wildcarding instead. When you use wildcarding, you replace the computer name or user ID with the wildcard character (*) when you add a member to an ACL or Security Group. The wildcard character stands for "any computer" or "any user ID." So, if you create an ACL member called **administrator@***, no matter what computer you log on to as **administrator**, you have the same permissions. If you use the wildcard character before the computer name, for example ***@computer**, any user who logs on to computer1 can do the actions that the ACL member has permission for.

Important

- Use the wildcard character with caution, as it can lead to security risks. Whenever possible, use the full name of the user.

Identifying Federated Authentication groups

If there is no matching FST group, when you add federated authentication groups to ACLs or Security groups, you must create an FST group with the same name as the federated authentication group you need to identify. These groups must be added to the corresponding ACLs for the InfoPrint Manager objects and operations.

Important

- When you log in using federated authentication, InfoPrint Manager validates the user roles to control access to different resources and functionalities based on the roles specified in the user identification token or in the **User information endpoint** response. To access the Web Management Interface, the system verifies whether the user ID belongs to the **acl_admin** group. To grant permissions to desired users, they must be members of the **acl_admin** group defined on the website of your chosen IdP.

Identifying LDAP/ Active Directory users and groups

When you add LDAP/Active Directory users to ACLs or Security groups, you identify them by their Login in the LDAP/Active Directory system. If these users use the same Login across workstations, they are able to work with InfoPrint Manager objects from other workstations.

Working with ACLs and groups

Use the Linux InfoPrint Manager Management Interface utility (requires X-mode) to manage the security of your print system. Open a Linux term window and enter `startipmmi` on the command line. On Red Hat Enterprise Linux, go to Applications →InfoPrint Manager and select Management Interface to start the Management Interface GUI. On SUSE Linux Enterprise Server, go to Computer → More

Applications → InfoPrint Manager and select Management Interface to start the Management Interface GUI.

Tab #1	Tab #2	Tab #3	Choices
InfoPrint Manager→	Security→	Groups→	Add Group
			Show Group
			Add User to Group
			Remove User from Group

Tab #1	Tab #2	Tab #3	Tab #4	Choices
InfoPrint Manager→	Security→	Access Control→	Operations→	Show Access Control List
			Servers→	Change Access Control List
			Destinations→	Remove Access Control List
			Queues→	

These three groups are created for you by default and are displayed when you select the **Groups** → **Show Group** option: **acl_admin, admin, and oper**. The authorized user is placed in the **acl_admin** group. To have the access necessary to modify the FST security characteristics, the user must be a member of the **acl_admin** group.

For other specific information supporting a menu choice, press **F1** for help.

Managing LDAP security for InfoPrint Manager for Linux

InfoPrint Manager provides an extension to the FST security that allows you to use an LDAP/Active Directory server for user authentication and access rights. When enabling LDAP security, the FST security continues to work as before.

Enable/Disable LDAP Security

Use the Linux InfoPrint Manager Management Interface (IPMMI) utility (it requires GUI) to enable or disable the LDAP security of your print system.

Depending on your needs, choose one of these options:

- Open a Linux term window and enter `startipmmi` on the command line, then press **enter**.
- On the RedHat Enterprise System, go to **Applications** → **InfoPrintManager**, and select Management Interface to start the Management Interface GUI. On the SUSE Linux Enterprise Server, go to **Computer** → **More Application** → **InfoPrint Manager**, and select Management Interface to start the Management Interface GUI.

To enable the LDAP security:

1. Click the **Security** tab.
2. Right-click the **LDAP** tab and choose **Enable LDAP Security** option.

   Note

   You must have at least one LDAP connection with valid authentication settings and search options to enable the LDAP security.

To disable the LDAP security:

1. Click the **Security** tab.
2. Right-click the **LDAP** tab and choose **Disable LDAP Security** option.

Create/Change LDAP Connection

To create an LDAP connection:

1. Click the **Security** tab.
2. Right-click the LDAP tab and choose the **New...** option.

To change an LDAP connection:

1. Click the Security tab.
2. Click the **LDAP** tab.
3. Right-click the **LDAP** connection and choose the **Open...** option.

Connection Name

Enter the connection name of the LDAP server.

IP Address or Host Name

Enter the host name or IP address of the LDAP server.

Port

Enter the port number that is used for communication.

Encryption Method

Enter an encryption method for the LDAP Server. Select the **Use Start TLS Extension** or the **Use SSL Encryption** option if you want to use the Start TLS or the SSL protocols.

Description

Enter an optional description.

Test Connection

If the information you enter is valid, you receive a confirmation message. If you enter incorrect settings, you receive an error message.

LDAP Authentication

Specify how InfoPrint Manager authenticates to the Lightweight Directory Access Protocol Server (LDAP). This information is used as authentication data for all existing LDAP connections. InfoPrint Manager uses the information to authenticate to the LDAP Server to retrieve specific data (for example, group membership and login attributes) about the entries.

To change the LDAP authentication:

1. Click the **Security** tab.
2. Click the **LDAP** tab.
3. Right-click the LDAP connection and choose the **Authentication...** option.

Bind DN or user

Enter the distinguished name (DN) of the account.

Bind Password

Enter your password.

Note

When you use the Anonymous login, it is not necessary to specify a value for: Bind DN/user or Bind Password.

Method

Select the method of authentication: **Simple** or **Digest**.

SASL Realm

Enter the name of the SASL Realm. This option is available when you use the **Digest** method.

Anonymous login

Select to authenticate as an anonymous user when no access permission is required.

Test Authentication

Validates settings. If the information you enter is valid, you receive a confirmation message. If you enter incorrect settings, you receive an error message.

Important

If you have multiple LDAP servers defined, the authentication information is common for all.

LDAP Search options

Users:

Search Base

Specifies the distinguished name (DN) of the branch in the LDAP directory tree where the users are located.

Login Attributes

Specifies the user attributes for the login in the LDAP server.

Object Class Filter

Specifies one or more optional object classes to filter when InfoPrint Manager searches for users.

Custom Filter

Specifies an optional custom filter that InfoPrint Manager uses when it searches users.

Groups:

Search Base

Specifies the distinguished name (DN) of the branch in the LDAP directory tree where the groups are located.

Group Name Attributes

Specifies an attribute that identifies the group name (for example, **cn**).

Group Member Attribute

Specifies the attribute of a user group (for example, **member**).

Object Class Filter

Specifies one or more optional object classes to filter when InfoPrint Manager searches for groups.

Custom Filter

Specifies an optional custom filter that InfoPrint Manager uses when searching for groups.

Performance Search Option

Use "memberOf" capability

Informs InfoPrint Manager that the group membership can be determined directly from the **memberOf** field.

Note

Make sure that this property is supported on your LDAP server.

Traverse Nested Groups

Applies only to the Microsoft Active Directory and it is used to check if a user is an indirect member of a group.

Use "ibm-allGroups" Capability

Applies only to IBM Tivoli Directory Server and it is used to determine the group membership directly from the attribute **ibm-allgroups**.

Note

Make sure that this option is supported on your LDAP server.

Use case-sensitive search

This option is used for case-sensitive matching in LDAP/AD queries. For instance, when this option is enabled, `USER` does not match `user` in the LDAP/AD query. This option must be consistent with the LDAP/AD server case-sensitive settings.

Managing Federated Authentication for InfoPrint Manager for Linux

InfoPrint Manager allows you to use federated authentication as an alternative to the existing FST or LDAP/AD security.

Federated authentication is a method of granting users secure access to InfoPrint Manager Web Management Interface and InfoPrint Manager Web Administration Interface by relying on external

identity providers (IdPs). Instead of managing separate user credentials within our system, federated authentication allows users to log in using their existing accounts from trusted third-party services.

Configuring Federated Authentication

Before enabling federated authentication, the InfoPrint Manager administrator must configure the connection settings between InfoPrint Manager and the federated authentication server.

To configure federated authentication:

1. Start the InfoPrint Manager Web Management Interface.
2. Click the **Security** tab in the left pane.
3. Select the **Federated Authentication** option.
4. In the **Federated Authentication** dialog, specify the required values.

   Important

   - Make sure that the information that you enter in each field is correct. InfoPrint Manager does not validate the data you enter in the **Federated Authentication** dialog.
   - For more information about a field property, click the ? button next to it.
5. Select the name of the identity provider (IdP) you want to use to authenticate from the **Identity provider** list.
6. In the **Authorization endpoint** field, enter the uniform resource identifier (URI) of the identity provider where client applications send the user to get authenticated.
7. In the **Client ID** field, enter the unique string representing the client identifier issued by the identity provider for InfoPrint Manager during the registration process.
8. In the **Client secret** field, enter the string representing the client passkey generated by the identity provider allowing the client to authenticate to the authorization server.

   Note

   - The **Client secret** field is not available when the **Use PKCE** option is checked.
9. In the **Token endpoint** field, enter the uniform resource identifier (URI) of the identity provider where the access token and the ID token are requested.
10. In the **User information endpoint** field, enter the uniform resource identifier (URI) of the identity provider where the user information is requested.

    Note

    - The **User information endpoint** field is displayed only for the Common Approach to Identity Assurance (CAIA) and Keycloak identity providers.
11. In the **Logout endpoint** field, enter the uniform resource identifier (URI) where the user is redirected to end the authentication session.
12. In the **Redirect hostname and port** field, enter the external host name and port for the InfoPrint Manager web server. The host name and port are used to generate the uniform resource identifier (URI) of the application, the location where the authorization server redirects the user once the application has been successfully authorized and granted an authorization code or access token.

Note

- A colon (:) must separate the host name/IP address and the port number. For example, `prod.yourcompany.com:14443` or `123.123.123.123:14443`.
- The web protocol (HTTP or HTTPS) is optional. If not specified, InfoPrint Manager detects the protocol in use and automatically adds it to the URI.

13. Check the **Allow insecure context** box if you want to enable the possibility to communicate with identity providers using self-signed certificates.
14. Check the **Enforce federated authentication** box to make federated authentication compulsory and to bypass the standard login dialog of the application.

Note

- If you entered any incorrect values in the **Federated Authentication** dialog, after checking the **Enforce federated authentication** box you no longer have access to the standard login dialog, making you unable to update the configuration values.

15. Check the **Use PKCE** box if you want to use a Proof Key for Code Exchange (PKCE).

Note

- The **Use PKCE** field is displayed only for the Okta identity provider.

16. In the **User roles paramater** field, enter the parameter name sent by the identity provider, that contains the user roles or group membership information relevant to InfoPrint Manager. You have to configure beforehand the identity provider to include the user roles or group membership information as a claim either in the user identification token for Active Directory Federation Services (AD FS) and Okta, or in the **User information endpoint** response for CAIA and Keycloak identity providers.

Note

- Keep in mind that the actual steps and terminology can vary depending on the identity provider being used. The group claim name might be `role` for CAIA, `groups` for Okta and Keycloak, and `memberof` for AD FS. Any other value you currently use is accepted.
- Consult the documentation of your identity provider for detailed instructions on how to do this customization.

Important

- InfoPrint Manager validates the user roles to control access to different resources and functionalities based on the roles specified in the user identification token or in the **User information endpoint** response. To access the Web Management Interface, the system verifies whether the user ID belongs to the **acl_admin** group. To grant permissions to desired users, they must be members of the **acl_admin** group defined on the website of your chosen IdP.

17. If your company uses proxy servers, ask your IT department for the correct IP address or host name and port number to use. In the **Proxy host and port** field, enter the host name and port of the proxy server used to communicate through.

Note

- A colon (:) must separate the host name/IP address and the port number. For example, `proxy.example.com:3128` or `123.123.123.123:3128`.

18. Click **Save**.

Note

- If you click the **Save** button while the **Federated Authentication** switch is set to ON, the web server restarts automatically to apply the changes.

1

Enable/Disable Federated Authentication

The InfoPrint Manager Web Management Interface allows you to enable or disable federated authentication.

To enable or disable federated authentication:

1. Go to the InfoPrint Manager Web Management Interface.
2. Click the **Security** tab in the left pane.
3. Select the **Federated Authentication** option.
4. Use the switch at the top of the **Federated Authentication** dialog to enable or disable federated authentication.

Note

- When you enable federated authentication, the FST or LDAP/AD security continues to work as before.
- When you enable or disable federated authentication, the web server restarts automatically.

Managing Transport Layer Security encryption for InfoPrint Manager for Linux

The Transport Layer Security (TLS) protocol encrypts internet traffic of all types, making secure communication possible across a network in a way designed to prevent eavesdropping and tampering.

InfoPrint Manager uses OpenSSL (https://www.openssl.org) to enable TLS encryption so that the communication between servers and clients is secure. InfoPrint Manager provides the OpenSSL libraries on all supported OS-es. By default, the latest version of TLS available with InfoPrint Manager is used.

Note

Not all communication channels can be encrypted. The exceptions are: LPD protocol, InfoPrint TCP/IP Port Monitor, pioinfo, pioinfoprt, piorpdm, piorpdf, PSF, transform clients, Transform Feature for InfoPrint Manager, and InfoPrint Select notifications.

For the Java components, InfoPrint Manager uses the Java native SSL/TLS implementation. By default, the latest version of TLS available with Java is used.

The identity of the communicating parties is authenticated using public-key cryptography (similar to HTTPS protocol).

Server and Client certificates

To encrypt the traffic between the InfoPrint Manager server and an InfoPrint Manager client, you need a digital certificate. Each digital certificate has two parts: a key (the private part) and the certificate (the

public part). The key should always be kept private, otherwise the communication is not secure anymore.

Important

- InfoPrint Manager only supports unencrypted certificate keys in PEM format.

You can:

1. Personally generate the certificate
2. Buy a certificate from a globally known third-party Certificate Authority (CA).

In the first case, you need to generate a CA authority certificate. The CA key file will sign the certificates you generate (for servers and clients). The CA certificate will authenticate the certificates you generate. The CA certificate (public part) must be available for InfoPrint Manager servers and clients in order to validate the certificates it receives.

In the second case, the certificates are verified using the Mozilla CA Certificate Store, a collection of CAs maintained by Mozilla organization. The raw file can be found at https://hg.mozilla.org/mozilla-central/raw-file/tip/security/nss/lib/ckfw/builtins/certdata.txt.

Note

- `certdata.txt` file is distributed with InfoPrint Manager in a form that is suitable to be used with OpenSSL.

The certificate included in `certdata.txt` usually expires in 398 days. To update the collection of CAs maintained by Mozilla organization, follow these steps:

1. On a computer with `perl` and `curl` installed and internet access, copy `mk-ca-bundle.pl` from a computer running InfoPrint Manager or from an InfoPrint Manager client computer. The `mk-ca-bundle.pl` script is located in the `<install path>\bin` directory.

   Note

   - You can also download the `mk-ca-bundle.pl` script from https://raw.githubusercontent.com/curl/curl/master/scripts/mk-ca-bundle.pl.
   - `curl` can be downloaded from https://curl.se/download.html.
   - Make sure that `perl` and `curl` are used in PATH.
2. Open a console and change the directory to the location of the `mk-ca-bundle.pl` script.
3. Execute the following command:

```
perl mk-ca-bundle.pl -s SHA256 -t -p \
SERVER_AUTH,CLIENT_AUTH,CRL_SIGN:TRUSTED_DELEGATOR
```

4. Rename `certdata.txt` to `ca-cert.pem`.
5. Copy `ca-cert.pem` to all computers running InfoPrint Manager and also to all InfoPrint Manager client computers. On Windows the location is: `C:\Windows\ipm` .On AIX and Linux the location is: `/etc/ipm`.
6. Restart each computer where the `ca-cert.pem` file has been copied.

A server certificate must have X509v3 Extended Key Usage (EKU) set to Server Authentication (TLS Web Server Authentication). X509v3 Key Usage must be Digital Signature and Key Encipherment.

To use Mutual Authentication (where the client validates the server certificate and the server validates the client certificate), you need both a server certificate and a client certificate on the server machine. Before starting any pdserver, make sure to configure both the server side and the client side (specifically for the user who starts InfoPrint Manager). Every client (whether on the pdserver machine or a different machine) must have their own client certificate and a corresponding `ipmssl.cfg` file, including the user who runs InfoPrint Manager.

A client certificate must have X509v3 Extended Key Usage (EKU) set to Client Authentication (TLS Web Client Authentication). X509v3 Key Usage must be Digital Signature.

Important

- You should not use a globally known third-party CA when Mutual Authentication is enabled. For example, using a Verisign certificate as a "known CA" means that ANYONE who has a certificate signed by them is authenticated. This is most likely not what you want. Generate your own CA certificate, server certificate, client certificates, etc.

The Subject field in the digital certificate must be set to the full computer name from DNS (example: ipmsrv.example.com) or set to a wildcard for your DNS domain (example: *.example.com).

Note

- Partial wildcards are not allowed.

The X509v3 Subject Alternative Name (SAN) must contain any other names or aliases for that computer (if available) and the IP addresses associated with the computer name. The field in the certificate must look like this: `DNS:alt1.example.com, DNS:alt2.example.com, IP:10.0.0.1, IP:fc00::1`, where host names are prefixed with "DNS:" and IP addresses with "IP:".

InfoPrint Manager warns you when your SSL/TLS certificate has less than 31 days left until it expires. The message is issued to error log and it looks like this: "5010-907 The SSL/TLS certificate for InfoPrint Manager expires on May 21 10:33:41 2029 GMT". To avoid interruptions you should consider renewing your certificate.

In order to identify if your InfoPrint Manager server is SSL/TLS enabled, check the error log file for the following message: "5010-908 InfoPrint Manager started with SSL/TLS enabled." after the InfoPrint Manager server start.

When encryption is disabled, InfoPrint Manager server accepts connections from previous versions of clients and servers (which do not understand TLS encryption). Once the encryption is enabled for an InfoPrint Manager server, such clients or servers are no longer able to communicate with the system. Only InfoPrint Manager servers that have encryption enabled or InfoPrint Manager clients that know to use TLS encryption will successfully communicate with the system.

Enabling Transport Layer Security encryption for InfoPrint Manager servers for Linux

To enable TLS encryption for InfoPrint Manager servers, follow these steps:

1. Copy the provided sample configuration file, `ipmssl.cfg`, from the `/usr/lpp/pd/cfg-samples/ssl/server` directory to the `/var/pddir/default_cell/ssl` directory.
2. If you are using a custom CA, copy the CA certificate file (public part) to the InfoPrint Manager server.

3. Copy the server certificate and key to the InfoPrint Manager server. Make sure that the certificate key is stored in a secure location and only available for reading to the user that runs the InfoPrint Manager server.
4. If you have Certificate Revocation List (CRL), copy the CRL file to the InfoPrint Manager server.
5. Edit the `ipmssl.cfg` file using a text editor. These keywords must be configured: EnableTLS, CertFile, and KeyFile. If the server certificate key file and the server certificate are combined in one file, only the `CertFile` keyword must be configured and `KeyFile` must be left commented. If the certificate files and keys are found in the same directory as the configuration file, the full path is optional. Otherwise, the full path should be specified.

   Uncomment and change the following keyword values (where available):

   1) Uncomment and set `EnableTLS` to `1`.
   2) Uncomment the `CertFile` keyword and specify the file name for the server certificate file.
   3) Uncomment the `KeyFile` keyword and specify the file name for the server certificate key.
   4) Optional: uncomment the `CAFile` keyword and specify the file name for the CA certificate if you are using a custom CA.
   5) Optional: uncomment the `CrlFile` keyword and specify the file name for the CRL certificate if you have one available.
   6)) Optional: uncomment and set `MutualAuthenticationRequired` to `1` if you are planning to use Mutual Authentication.

      The last two optional keywords are commonly used if errors occur during the TLS handshake:

      - `CertValidationIgnoreHostName` can help if the client host name from the DNS is different from the value set in the certificate presented by the client during the TLS handshake (Subject field and X509v3 Subject Alternative Name (SAN) field). Setting this keyword to 1 prevents the validation of the host name.

        This table shows how validation is made if the Subject field from the certificate contains a wildcard:

Host name	**Certificate Subject or Certificate Subject Alternative Name**	**Validation**
host.example.com	host.example.com	OK
host.example.com	*.example.com	OK
host.subdomain.example.com	*.subdomain.example.com	OK
host.example.com	host.another-example.com	FAIL
host.subdomain.example.com	host.another-subdomain.example.com	FAIL
host.example.com	host*.example.com	FAIL
host.example.com	*host.example.com	FAIL

host.subdomain.example.com	host*.subdomain.example.com	FAIL
host.subdomain.example.com	*host.subdomain.example.com	FAIL

Note

To be valid, a host name in certificate must have at least 2 (two) dots.

- `IgnoreCertificateErrors` ignores any errors related to the client certificate (invalid value presented in the certificate Subject or SAN fields, certificate expired, client not sending a certificate, and so on.)

6. Reboot the computer or restart all running InfoPrint Manager components (Notification, SAP, LPD, DPF, IPP, LDAP, MVSD Download, Web Server) and all InfoPrint Manager servers.

Sample ipmssl.cfg configuration file:

```
#
# IPM SSL/TLS configuration file (server)
#

##############################################################################
#
# Empty lines and whitespace-only lines are ignored, as are lines whose
# first non-whitespace character is a semicolon (;) or a hash (#).
# This file use hashes to denote commentary and semicolons for options
# you may wish to configure.
# Every comment applies to the following section or option. The defaults
# refer to IPM's built-in values, not anything set in this file.
#
# Uncomment the lines where you want to do a change and enter the
# desired value. Option names are case-sensitive.
#
# If a full path is required but only a file name is provided
# (i.e., no '/' or '\' in file name), the lookup for the file is
# done only in the directory where this configuration file
# is located. Apply to: CAFile, CrlFile, CertFile, KeyFile.
#
# IMPORTANT: If you make changes to this file, make sure that
#                    you restart all the InfoPrint Manager processes.
#                    On Windows operating system, make sure that you
#                    also restart Print Spooler service.
#
##############################################################################

#
# Enable/disable encryption (0 = disabled, 1 = enabled).
# Default: 0
#
;EnableTLS = 0
#
# Path to a file containing one or more Certificate Authority (CA).
# Required if mutual authentication is enabled and _client_ certificate is
# not signed by a globally known CA. See also MutualAuthenticationRequired.
# Default:  <empty>
#
;CAFile =

#
# Path to a file containing one or more Certificate Revocation List (CRL).
# Default:  <empty>
#
```

```
;CertFile =
#
# If the key is not combined with the certificate this directive will
# specify the path to Certificate Private Key (KEY) file.
# Default: <not set>, required.
#
;KeyFile =
#
# Enable/Disable mutual authentication (0 = disabled, 1 = enabled).
# Default: 0
#
# Note that you should NOT use a globally known CA when mutual authentication
# is enabled. i.e., using a Verisign certificate as a "known CA" means that
# ANYONE who has a certificate signed by them will be authenticated.
# This is most likely not what you want. Generate your own CA certificate,
# client keys and certificates, etc.
#
;MutualAuthenticationRequired = 0

#
# When enabled will prevent comparing client name with the name in
# certificate subject. Also apply to Subject Alternative Name (SAN).
# Ignored if MutualAuthenticationRequired is not enabled.
# Ignored if IgnoreCertificateErrors is enabled.
# (0 = disabled, 1 = enabled).
# Default: 0
#
;CertValidationIgnoreHostName = 0

#
# Ignore any errors related to certificate validation (0 = disabled,
# 1 = enabled).
# Default: 0
#
;IgnoreCertificateErrors = 0
```

Note

- If SSL/TLS is enabled and you use IPPGW, you must change the argument to flag **-h** in the `/etc/rc.ippgw.daemons` file from localhost to the FQDN of your InfoPrint Manager computer.

Enabling Transport Layer Security encryption for InfoPrint Manager clients for Linux

Normally, there is no need to configure anything on the client side if you use a server certificate from a globally known third-party CA.

If you are using a custom CA certificate, or if you use Mutual Authentication, you must configure the InfoPrint Manager clients to be able to communicate with the server. There are two places where the client configuration file can reside:

- The user configuration directory:
  For AIX/ Linux: `~/.ipm`

  For Windows: `%APPDATA%\Ricoh\InfoPrint Manager\ssl`
- The administrator-enforced configuration directory:
  For AIX / Linux: `/etc/ipm`

For Windows: `%windir%\ipm`

A configuration file enforced by an administrator must be readable for everyone, but not writable. Any directive present in the administrator version of the configuration file overwrites the same directive in the user version of the configuration file, regardless of its configuration in the latter file. At least one of the two client configuration files must exist in order to modify the default SSL encryption behavior. If the certificate files and keys are found in the same directory as the configuration file, the full path is optional. Otherwise, the full path should be specified.

To enable TLS encryption for InfoPrint Manager clients, follow these steps:

1. Copy the provided sample configuration file `ipmssl.cfg` from `/usr/lpp/pd/cfg-samples/ssl/client` directory to the desired location of the client configuration file (user or administrator).
2. If you are using a custom CA, copy the CA certificate file (the public part) to the InfoPrint Manager client.
3. If the server uses Mutual Authentication, copy the client certificate and key to the machine running the InfoPrint Manager client. Ensure that the certificate key is secured and only available for reading to the user running the InfoPrint Manager client.
4. If you have Certificate Revocation List (CRL), copy the CRL file to the InfoPrint Manager client.
5. Edit the `ipmssl.cfg` file using a text editor.

   If the client certificate key file and the server certificate are combined in one file, only the `CertFile` keyword must be configured and `KeyFile` must be commented. If the certificate files and keys are found in the same directory as the configuration file, the full path is optional. Otherwise, the full path should be specified.

   Uncomment and change the following keywords values (where available):

   1) Optional: uncomment the `CertFile` keyword and specify the file name for the client certificate file if the InfoPrint Manager server is using Mutual Authentication.
   2) Optional: uncomment the `KeyFile` keyword and specify the file name for the client certificate key if the InfoPrint Manager server is using Mutual Authentication.
   3) Optional: uncomment the `CAFile` keyword and specify the file name for the CA certificate if you are using a custom CA.
   4) Optional: uncomment the `CrlFile` keyword and specify the file name for the CRL certificate if you have one available.

   The last two optional keywords are commonly used if errors occur during TLS handshake:

   - `CertValidationIgnoreHostName` can help if the server host name from the DNS is different from the value set in the certificate presented by the server during TLS handshake (Subject field and X509v3 Subject Alternative Name (SAN) field). Setting this keyword to 1 prevents the validation of the host name. This table shows how the validation is made in case the Subject field from the certificate contains a wildcard:

Host name	**Certificate Subject or Certificate Subject Alternative Name**	**Validation**
host.example.com	host.example.com	OK
host.example.com	*.example.com	OK

host.subdomain.example.com	*.subdomain.example.com	OK
host.example.com	host.another-example.com	FAIL
host.subdomain.example.com	host.another-subdomain.example.com	FAIL
host.example.com	host*.example.com	FAIL
host.example.com	*host.example.com	FAIL
host.subdomain.example.com	host*.subdomain.example.com	FAIL
host.subdomain.example.com	*host.subdomain.example.com	FAIL

Note

To be valid, a host name in the certificate must have at least 2 (two) dots.

- `IgnoreCertificateErrors` ignores any errors related to the server certificate (invalid value presented in the certificate Subject or SAN fields, certificate expired, and so on.)

6. Reboot the computer.

Sample `ipmssl.cfg` configuration file:

```
#
# IPM SSL/TLS configuration file (client)
#

##############################################################################
#
# Empty lines and whitespace-only lines are ignored, as are lines whose
# first non-whitespace character is a semicolon (;) or a hash (#). This
# file uses hashes to denote commentary and semicolons for options you
# might want to configure.
# Every comment applies to the following section or option. The defaults
# refer to IPM's built-in values, not anything set in this file.
#
# Uncomment the lines where you want to do a change and enter the desired
# value. Option names are case-sensitive.
#
# Any directive found in administrator version of the configuration file
# overwrites the same directive in user version of the configuration file
# regardless of what is configured in the later file or not.
#
# If a full path is required but only a file name is provided
# (i.e., no '/' or '\' in file name), the lookup for the file is
# done only in the directory where this configuration file
# is located. Apply to: CAFile, CrlFile, CertFile, KeyFile.
#
# IMPORTANT: If you make changes to this file, make sure that
#                    you restart all the InfoPrint Manager processes.
#                    On Windows operating system, make sure that you
#                    also restart Print Spooler service.
#
##############################################################################

#
# Path to a file containing one or more Certificate Authority (CA).
# Required if _server_ certificate is not signed by a globally known CA.
```

```
# Default: <empty>
#
;CAFile =

#
# Path to a file containing one or more Certificate Revocation List (CRL).
# Default: <empty>
#
;CrlFile =

#
# Path to unencrypted PEM Certificate (CRT) file.
# Default: <not set>, required only if the remote server require mutual
# authentication.
#
;CertFile =

#
# If the key is not combined with the certificate, this directive
# specifies the path to Certificate Private Key (KEY) file.
# Default: <not set>, required only if the remote server requires mutual
# authentication.
#
;KeyFile =

#
# When enabled, it prevents comparing _server_ name with the name in
# certificate subject. It also applies to Subject Alternative Name (SAN).
# Ignored if IgnoreCertificateErrors is enabled.
# (0 = disabled, 1 = enabled).
# Default: 0
#
;CertValidationIgnoreHostName = 0

#
# Ignore any errors related to certificate validation (0 = disabled, 1 =
# enabled).
# Default: 0
#
;IgnoreCertificateErrors = 0
```

Enabling Transport Layer Security encryption for sendmemo

The `sendmemo` component is responsible with submitting emails from the InfoPrint Manager server, either as Email Notifications or through an Email actual destination. If you want to enable encrypted communication between the InfoPrint Manager server and your email server, you must configure the sendmemo SSL/TLS configuration file. The configuration file, like `smtp-server-port` and `smtp-server-host` server attributes, is configured per server. Therefore, the location of the configuration file is the server working directory: `/var/pd/<your-server-name>`, where *<your-server-name>* is the name of the server that you want to have an encrypted communication with your email server.

Note

- Your email server must also have SSL/TLS configured and functional. You can choose between `STARTTLS` or `SMTPS`. By default, `STARTTLS` can be used with port 25/tcp (smtp) and 587/tcp (submission). By default, `SMTPS` can be used with port 465/tcp (smtps). However, you can use any other TCP port as long as your email server is configured as such.

By default, sendmemo uses TLS to encrypt the connection to the email server. It negotiates the highest available cipher shared with the email server. By default, SSLv3 encryption is disabled. If your server is

very old and needs Secure Sockets Layer (SSL) version 3 for encrypted connections to it, you can set the environment variable `IPM_ENABLE_SSL_V3` to a non-empty value.

Important

- SSLv2 is disabled and cannot be enabled.

If the certificate files and keys are found in the same directory as the configuration file, the full path is optional. Otherwise, the full path should be specified.

InfoPrint Manager supports `SMTP AUTH PLAIN` for authentication.

To enable TLS encryption for sendmemo, you must follow these steps:

1. Copy the provided sample configuration file `sendmemo-ssl.cfg` from `/usr/lpp/pd/cfg-samples/ssl/sendmemo` directory into the working directory of the server.
2. If you are using your own CA, copy the CA certificate file (the public part) to the InfoPrint Manager server.
3. If you have Certificate Revocation List (CRL), copy the CRL file to the InfoPrint Manager server.
4. Edit the `sendmemo-ssl.cfg` file using a text editor. Uncomment and change the following keywords values (where available):
   1) Uncomment the `EnableTLS` keyword and the desired value as follows:
      - 0: SMTP session encryption disabled
      - 1: use STARTTLS
      - 2: use SMTPS

      Note

      – The correct SMTP port number must be set using the specific InfoPrint Manager server attribute `smtp-server-port`.
   2) Optional: Uncomment the `CAFile` keyword and specify the file name for the CA certificate if you are using a custom CA.
   3) Optional: Uncomment `CrlFile` keyword and specify the file name for the CRL certificate if you have one available.
   4) Optional: Uncomment the `Hostname` keyword and specify the host name that sendmemo uses to identify itself (EHLO) to the email server. This is useful when you are behind NAT and sendmemo needs to send email to the SMTP servers that are outside of your LAN or when autodetect fails. Some email servers might ignore an invalid host name passed to EHLO, others might not.

      The format must be one of the following (RFC 2821):

      FQDN host name:

      example: host.example.com

      Brackets enclosed IPv4 address:

      example: [1.2.3.4]

      Brackets enclosed IPv6 address:

      example: [IPv6:fc00::1]

5) Optional: Uncomment the `User` keyword and specify the SMTP user name for email server authentication.
6) Optional: Uncomment the `Password` keyword and specify the user password for email server authentication.

Note

- The `User` and `Password` parameters are ignored if SMTP session encryption is disabled.

The last two optional keywords are commonly used if errors occur during TLS handshake:

- `CertValidationIgnoreHostName` can help if the server host name from the DNS is different from the value set in the certificate presented by the server during TLS handshake (Subject field and X509v3 Subject Alternative Name (SAN) field). Setting this keyword to 1 prevents validation of the host name. This table shows how validation is made in case the Subject field from the certificate contains a wildcard:

Host name	Certificate Subject or Certificate Subject Alternative Name	Validation
host.example.com	host.example.com	OK
host.example.com	*.example.com	OK
host.subdomain.example.com	*.subdomain.example.com	OK
host.example.com	host.another-example.com	FAIL
host.subdomain.example.com	host.another-subdomain.example.com	FAIL
host.example.com	host*.example.com	FAIL
host.example.com	*host.example.com	FAIL
host.subdomain.example.com	host*.subdomain.example.com	FAIL
host.subdomain.example.com	*host.subdomain.example.com	FAIL

Note

  - To be valid, a host name in the certificate must have at least 2 (two) dots.
- `IgnoreCertificateErrors` ignores any errors related to the server certificate (invalid value presented in the certificate Subject or SAN fields, certificate expired, and so on.)

Sample `sendmemo-ssl.cfg` configuration file:

```
#
# sendmemo SSL/TLS configuration file
#

##############################################################################################
#
# Empty lines and whitespace-only lines are ignored, as are lines whose
# first non-whitespace character is a semicolon (;) or a hash (#). This
# file uses hashes to denote commentary and semicolons for options you
# might want to configure.
# Every comment applies to the following section or option. The defaults
```

```
# refer to IPM's built-in values, not anything set in this file.
#
# Uncomment the lines where you want to do a change and enter the desired
# value. Option names are case-sensitive.
#
# If a full path is required, but only a file name is provided (i.e., no
# '/' or '\' in file name), the lookup for the file is done only in the
# directory where this configuration file is located. Apply to: CAFile,
# CrlFile, CertFile, KeyFile.
#
##############################################################################

#
# Enable/disable SMTP session encryption. Available options:
#   0: SMTP session encryption disabled
#   1: use STARTTLS
#   2: use smtps
# Note that the correct SMTP port number must be set using the specific
# IPM server attribute (smtp-server-port).
#
# Default: 0
#
;EnableTLS = 0

#
# Path to a file containing one or more Certificate Authority (CA).
# Required if _server_ certificate is not signed by a globally known CA.
# Default: <empty>
#
;CAFile =

#
# Path to a file containing one or more Certificate Revocation List (CRL).
# Default: <empty>
#
;CrlFile =

#
# When enabled, it prevents comparing _server_ name with the name in
# certificate subject. It also applies to Subject Alternative Name (SAN).
# Ignored if IgnoreCertificateErrors is enabled.
# (0 = disabled, 1 = enabled).
# Default: 0
#
;CertValidationIgnoreHostName = 0
#

# Ignore any errors related to certificate validation (0 = disabled, 1 =
# enabled).
# Default: 0
#
;IgnoreCertificateErrors = 0

# Specify the host name that sendmemo uses to identify itself (EHLO).
# The format must be one of the following (RFC 2821):
#    * FQDN hostname:
#        host.example.com
#    * Brackets enclosed IPv4 address:
#        [1.2.3.4]
#    * Brackets enclosed IPv6 address:
#        [IPv6:fc00::1]
# Anything else is invalid. Some email servers might ignore an invalid
# host name passed to EHLO, other will not.
#
```

```
# Useful when you are behind NAT and sendmemo needs to send email to SMTP
# servers that are outside of your LAN or when autodetect fails.
# Default: sendmemo will autodetect host name.
;Hostname =

#
# SMTP username for SMTP AUTH PLAIN.
# When set, sendmemo will try to authenticate with remote SMTP server.
# Ignored if EnableTLS is 0 or if password is empty.
# Default: <empty>
#
;User =

#
# SMTP password for SMTP AUTH PLAIN.
# Ignored if EnableTLS is 0. Required if SMTP username is set.
# Default: <empty>
#
;Password =
```

Managing job encryption for InfoPrint Manager for Linux

To prevent data from being accessed, modified, or stolen, InfoPrint Manager can encrypt the files that enter the system.

InfoPrint Manager uses OpenSSL (https://www.openssl.org) to enable job encryption. To encrypt and decrypt large amounts of data, InfoPrint Manager uses symmetric encryption. Only printable job files are encrypted.

Note

- On AIX and Linux, when encryption is enabled, **`pdpr -l`** copies the file instead of creating a symbolic link.

To ensure that all InfoPrint Manager files are encrypted, instead of using job encryption, encrypt all the partitions where InfoPrint Manager stores files: `/var/pd`, `/var/psf`, `/var/psf/segments`, `/tmp`, and `swap`.

Partition encryption is an additional layer of security, not a replacement for InfoPrint Manager file encryption. Encrypted partitions ensure that if the hard drive (HDD) is stolen or disposed of, the data on it cannot be recovered. However, if the HDD remains in the computer where it was encrypted, access to the data (files or jobs) is transparent. On the other hand, InfoPrint Manager file encryption protects data by preventing access if the HDD is stolen or discarded, and does not allow transparent access to file contents.

To achieve complete encryption that covers the OS-specific print systems, encrypt the `/var`, `/tmp`, and `swap` partitions.

Note

- If encrypting swap is not possible or practical, ensure that InfoPrint Manager has ample memory available to prevent decrypted data from being written to disk by the operating system.

Any printable file is encrypted as soon as it enters InfoPrint Manager. The job is processed (sniffed, RIPped, etc) in encrypted form, including all intermediary and temporary files created during

processing. In a few cases, the job might be decrypted to disk due to functionality or system service constraints, or similar limitations.

Decryption to disk takes place during processing when:

- A custom step (transform) is required for the job
- A run script action needs to be executed as part of a Print Rule for the job.

The job will be re-encrypted once the processing is done.

The decryption to disk occurs before the job is sent to a specific print system or to a printer when:

- using a custom command specified by the customers
- using one of the following DSSs: CUPS DSS, BSD DSS, DFE DSS, PSF-Other DSS, or PSF Command DSS.

For Anyplace Print across namespaces, a move job is made. The job files are decrypted and a job is submitted. Files are encrypted again on the destination server.

To identify if job encryption is enabled on your InfoPrint Manager server, check the error log file for the following message: "`5010-909 InfoPrint Manager started with Job Encryption enabled.`", after the InfoPrint Manager server has started.

Enabling job encryption for InfoPrint Manager for Linux

To enable job encryption for InfoPrint Manager servers:

1. Stop all running instances of InfoPrint Manager. In interoperated environments, stop all running instances of InfoPrint Manager on all interoperated machines.
2. Copy the sample configuration file, **`ipm-file-encryption.cfg`**, from `/usr/lpp/pd/cfg-samples/file-encryption` directory to `/var/pddir/default_cell/crypto` directory.
3. Uncomment and set **`EnableFileEncryption`** to `1`.
4. Uncomment and set **`Cipher`** to the desired cipher to use for encryption.

Note

- The list of available ciphers can be obtained by running the following command:

```
openssl-ipm enc -ciphers
```

  Do not include the first character ('-') from the cipher name.

- Using an AES cipher with a processor that supports hardware encryption and the AES instruction set can speed up encryption and decryption times. For the AES instruction set, see https://software.intel.com/content/www/us/en/develop/articles/intel-advanced-encryption-standard-instructions-aes-ni.html.

5. Run **`pdenc_jobs`** as the user that runs InfoPrint Manager or root. In interoperated environments, run **`pdenc_jobs`** sequentially on all interoperated machines.
6. Start all instances of InfoPrint Manager or reboot.

Important

- Do not change the cipher in the **ipm-file-encryption.cfg** configuration file once encryption has been enabled. To change the cipher, follow the procedure in Disabling job encryption for InfoPrint Manager for Linux, p. 154, change the cipher in the **ipm-file-encryption.cfg** configuration file, and then follow the procedure to enable job encryption for InfoPrint Manager.
- Do not remove the **ipm-file-encryption.dat** file once encryption has been enabled. This file is stored in the same directory as the **ipm-file-encryption.cfg** configuration file.
- Before running **pdenc_jobs**, make sure that there is at least `1.5* (size of jobs + rip files)` free space available on the partition where jobs are stored.

Disabling job encryption for InfoPrint Manager for Linux

To disable job encryption for InfoPrint Manager servers:

1. Stop all running instances of InfoPrint Manager. In interoperated environments, stop all running instances of InfoPrint Manager on all interoperated machines.
2. Run **pddec_jobs** as the user that runs InfoPrint Manager or root. In interoperated environments, run **pddec_jobs** sequentially on all interoperated machines.
3. Uncomment and set **EnableFileEncryption** to 0 or remove the **ipm-file-encryption.cfg** file.
4. Start all instances of InfoPrint Manager or reboot.

Note

- Before running **pddec_jobs**, make sure that there is at least `1.5* (size of jobs + rip files)` free space available on the partition where jobs are stored.

Sample `ipm-file-encryption.cfg` file:

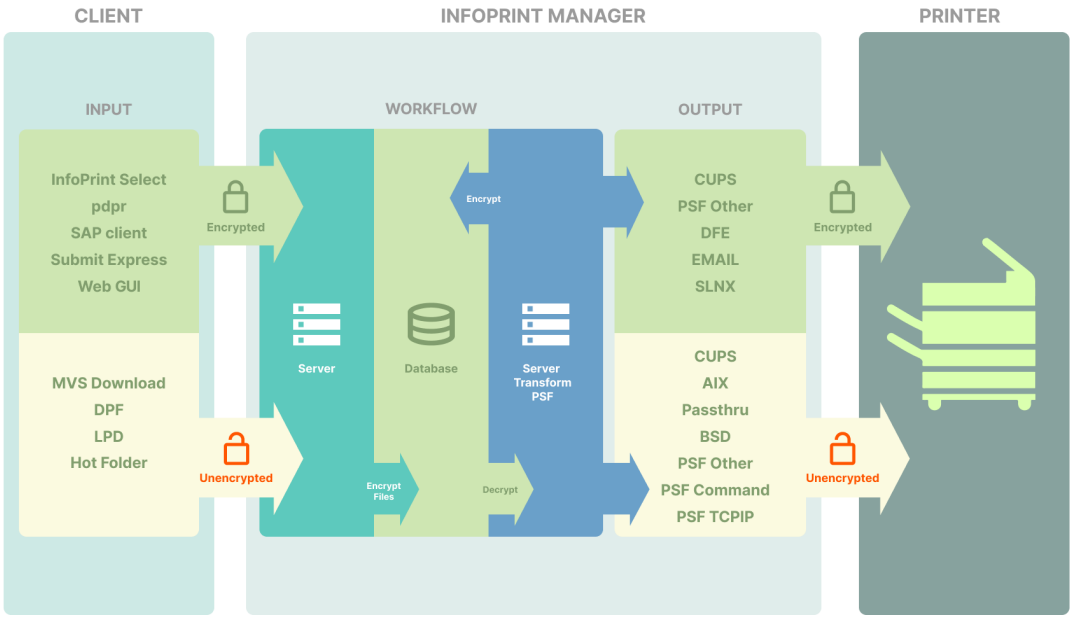
```
#
# IPM File Encryption configuration file
#

##############################################################################
#
# Empty lines and whitespace-only lines are ignored, as are lines whose
# first non-whitespace character is a semicolon (;) or a hash (#).
# This file uses hashes to denote commentary and semicolons for options
# you may wish to configure.
# Every comment applies to the following section or option. The defaults
# refer to IPM's built-in values, not to anything set in this file.
#
# Uncomment the lines where you want to make a change and enter the
# desired value. Option names are case-sensitive.
#
##############################################################################

#
# Enable/disable encryption (0 = disabled, 1 = enabled).
# Default: 0
#
;EnableFileEncryption = 0

#
# Specifies the cipher to be used when encrypting/decrypting files.
```

```
# To obtain the supported list of ciphers run:
#
#   openssl-ipm enc -ciphers
#
# Do not include the first character ('-') from the cipher name.
# Default: <not set>, required.
#
# DO NOT change the cipher in this file after encryption has been enabled!
#
# To change the cipher, stop all pdservers, run the decryption tool, change the
# cipher in this file, run the encryption tool and then start all pdservers.
#
;Cipher =
```


Managing encryption in transit and at rest in InfoPrint Manager for Linux

This section outlines the workflow, limitations, and constraints of InfoPrint Manager encryption for stored and transmitted data.

InfoPrint Manager encryption workflow



InfoPrint Manager Web Administration Interface (Web GUI), the pdpr command, InfoPrint Submit Express, the SAP client, and InfoPrint Select can send jobs to InfoPrint Manager using an encrypted channel. The other clients use plain communication; therefore, IPSec or other channel encryption mechanisms are recommended.

Once a job is received, InfoPrint Manager encrypts any data written to disk. The only exceptions are documents processed by Print Rules and Custom Steps, where the document is decrypted for third-party applications.

When jobs are printed, some output methods do not support encryption. Notable examples include the AIX printing system, CUPS, and the Windows print subsystem. To pass the document to native print systems, the data must be decrypted. PSF TCP/IP keeps the data encrypted on disk; however, the communication channel itself is not encrypted.

InfoPrint Manager encryption limitations and constraints

The following components do not have encryption in transit capabilities:

- MVS Download Receiver
- DPF Receiver
- LPD Gateway
- Hot Folders

The following components do not encrypt temporary data:

- Receiving components:
  - MVS Download Receiver
  - DPF Receiver
  - LPD Gateway
  - Hot Folders
- DSSs that generate unencrypted data:
  - CUPS
  - BSD
  - PSF Other
  - PSF Command

When encryption at rest is enabled, InfoPrint Manager loads files into memory for faster processing. If you print large files, ensure that the data segment allocation settings are sufficiently large; otherwise, some jobs may fail due to memory limitations. For more information, see *RICOH InfoPrint Manager for AIX and Linux: Configuring and Tuning Guide*, the "Linux virtual memory" topic.

Enabling Secure Sockets Layer (SSL) in the Web Server for HTTPS protocol

Secure Socket Layer (SSL) is a protocol that provides security for communications between browsers and the Web Server by implementing encrypted data and certificate-based authentication.

You can activate SSL in the Web Server component and link it to a digital certificate. It has an existing digital certificate, self-sign certificates, which should be replaced with personalized certificates generated by customers.

Before you configure the SSL, you must obtain a digital certificate and store it on the computer that the base product is installed on. Follow the instructions provided by the certificate authority (CA) for installing the certificate. You must also know where the keystore is located on the server and the password for the keystore.

In addition, save copies of these files in a safe location so you can find them easily if you need to restore them in the future.

To generate a key and keystore, use the Java **keytool** command. For details about using **keytool**, consult your CA or Java documentation.

Note

Your private key password and keystore password must be the same. If they are not the same, you receive a `java.io.IOException` error.

To configure SSL:

1. Purchase a digital certificate from a reputable certificate authority (CA).
2. Follow the instructions provided by the CA to install the certificate on the computer where the InfoPrint Manager server is installed.
3. Use the Java **keytool** command to generate a key and keystore. For details about using the **keytool** command, refer to the Java documentation. Write down the location of the keystore and the password for the keystore. You need them later.
4. Log in to the computer as the InfoPrint Manager administrator.
5. Copy the `.keystore` file previously created to the `home` directory for the InfoPrint Manager system user.
6. Change the owner of the `.keystore` file to the appropriate owner and group. Type this command, substituting the InfoPrint Manager user ID for *system_user* and the InfoPrint Manager group for *system_group*:`chown system_user:system_group .keystore`.

   The default system user is **ipm1** and the default group is **sys**.
7. If you cannot copy the `.keystore` file and change the owner, log out and log back in to the primary computer as the root user or use **su** to become the root user and repeat the previous two steps
8. Switch to the InfoPrint Manager user. Type this command, substituting the InfoPrint Manager user ID for *system_user*:`su - system_user`.
9. Go to the `/usr/lpp/pd/ipmws/conf` directory.
10. Open the `server.xml` file and make these changes:

    1) Find this configuration and uncomment it:

```
<Connector
port="14443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150"
maxParameterCount="1000"
maxPartCount="100"
SSLEnabled="true">
<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
<SSLHostConfig  protocols="TLSv1.2+TLSv1.3"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,
TLS_CHACHA20_POLY1305_SHA256"">
```

```
<Certificate
certificateKeystoreFile="etc/InfoPrint.keystore"
certificateKeystorePassword="changeit"
type="RSA"/>
</SSLHostConfig>
</Connector>
```


2) Replace the value for *keystoreFile* with the path to your `.keystore` file.
3) Replace the *keystore_password* by changing the keystorePass value with the password to your keystore.
4) Save and close the file.

11. If you use another port than the default one (14000), run the **pdwsinitports** utility.
12. Restart the InfoPrint Manager Web Server.
13. Verify the secure connection by opening a web browser and typing this address, replacing server_name with the host name or IP address of the computer where InfoPrint Manager is installed: `https://server-name:14443/IPM.`

If you use a self-signed certificate or if the certificate is not specifically tied to the server, the web browser issues a warning that the certificate is not trusted.

If you install service updates or a new version of the InfoPrint Manager, you must activate SSL again because the installation process clears the SSL settings in the Web Server component.

If you need to update or replace your digital certificate, install the new certificate into the keystore and remove the old certificate. You do not have to do this task again because InfoPrint Manager can find the new certificate in the keystore.

Configuring Tomcat to redirect all HTTP requests to HTTPS

You can configure Tomcat to redirect all HTTP requests to HTTPS, ensuring that all communication with the Web Server is encrypted and secure.

To force the Web Server redirect all calls to a secure connection:

1. Use a self-signed certificate or a certificate generated by a Certificate Authority (CA).
   InfoPrint Manager Web Server already provides a sample self-signed certificate. The certificate must be signed by the trusted third party so that it can work in the visitors browsers without warnings. To obtain a signed certificate, choose a CA and follow its instruction.
2. Install and configure the Web Server so that you can use your certificate. For more information, see Enabling Secure Sockets Layer (SSL) in the Web Server for HTTPS protocol, p. 156. To use the sample certificate, enable the SSL connector in `/usr/lpp/pd/ipmws/conf/server.xml` file.
3. Go to `/usr/lpp/pd/ipmws/conf/web.xml` file, scroll to the bottom of the file and add the following just above the `</web-app>` entry:

```
<!-- Force HTTPS, required for HTTP redirect! -->
<security-constraint>
<web-resource-collection>
<web-resource-name>Protected Context</web-resource-name>
<url-pattern>/*</url-pattern>
</web-resource-collection>
```

```
<!-- auth-constraint goes here if you require authentication -->
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

4. Save the web.xml file and restart the Web Server from InfoPrint Manager Management Interface for these changes to take effect.

Note

- If you have multiple Web Servers running, make sure that they all use the same HTTP or HTTPS protocol.

Customizing error logs in the InfoPrint Manager server

InfoPrint Manager provides three configuration files for various InfoPrint Managers servers and daemons:

1. **spl_error.cfg** for the InfoPrint Manager server error log.
2. **notifyd_error.cfg** for the Notification daemon error log.
3. **sap_error.cfg** for all the SAP daemons.

If you want to customize error logs for one of these servers or daemons to control their size, whether they wrap, the severity level of messages written to the logs, and the number of back-up logs that InfoPrint Manager saves, do one of the following procedures.

Customizing an InfoPrint Manager server error log

By default, the InfoPrint Manager server error log starts when the InfoPrint Manager server starts. Use the following procedure to customize the InfoPrint Manager server error log(s). Any InfoPrint Manager server processes that are running on this Linux server are changed by this customization the next time they are started.

1. Find the **spl_error.cfg** file in the **/usr/lpp/pd/bin** directory. Copy it into the **/var/pd/servername** directory if you want to customize the error log only for **servername**, or into the **/var/pd/** or **%PDBASE%** directory if you want to customize the error log for all InfoPrint Manager servers:

```
cp /usr/lpp/pd/bin/spl_error.cfg /var/pd/servername/spl_error.cfg
```

or

```
cp /usr/lpp/pd/bin/spl_error.cfg /var/pd/spl_error.cfg
```

Note

If you already customized your server error logs, remember to change the name or move the current **spl_error.cfg** file in **/var/pd/servername** or **/var/pd/** so that your customization values can be copied over into the newer version of the file.

2. Open the copy of the **spl_error.cfg** file that you created and specify the settings for the log attributes that you want.

For example, you can modify the **log-backup-number** attribute, which allows you to specify the number of backup logs that InfoPrint Manager saves, including the initial **error.log.BAK** file.

Sample spl_error.cfg file provided by InfoPrint Manager


```
#
# ErrorLog Configuration File
#
# Log size (units = KBytes)
log-size = 1024
# Wrap On?
log-wrap = true
# Severity?
log-severity = debug
# Number of backup log files
log-backup-number = 10
```

When you start the InfoPrint Manager server after a shut down, it backs up the first error log by creating an **error.log.BAK** file. With each successive restart, it creates an additional **.BAK** file, adding a timestamp as a fourth qualifier. By default, the **log-backup-number** attribute specifies a value of **10**, meaning that you can have nine back up files with timestamps and the initial **.BAK** file. You can reduce this number as low as `1` or increase it as high as `999`.

For a complete list of the error log attributes that you can customize, see the "Attributes for logs" section in the "InfoPrint Manager object attributes" chapter of *RICOH InfoPrint Manager: Reference*.

3. Once you have finished defining your options, save and close the **/var/pd/servername/spl_error.cfg** or **/var/pd/spl_error.cfg** file.
4. To shut down the InfoPrint Manager server so that these customization changes can take effect, specify from the Linux command line:`pdshutdown -w now` *`servername`*.

   Note

   If you do not need the customization change to take effect immediately, you can skip the final two steps and just wait until the next time the Linux system where the InfoPrint Manager server is installed on is restarted.
5. To restart the server, enter `start_server -F` *`servername`* where *`servername`* is the name of your InfoPrint Manager server.

Customizing a Notification daemon error log

By default, the Notification daemon error log starts when the Notification server starts. Use this procedure to customize the InfoPrint Notification daemon error log.

1. Find the **notifyd_error.cfg** file from the **/usr/lpp/pd/bin** directory where the file is installed and copy it into the **/var/pd** directory:

```
cp /usr/lpp/pd/bin/notifyd_error.cfg /var/pd/notifyd_error.cfg
```

Note

If you already customize your Notification daemon error log, remember to change the name or move the current **notifyd_error.cfg** file in **/var/pd** so that your customization values can be copied over into the newer version of the file.

2. Open the copy of the **notifyd_error.cfg** file that you created and specify the settings for the log attributes that you want.

   For example, you can modify the **log-backup-number** attribute, which lets you specify the number of backup logs that InfoPrint saves, including the initial **error.log.BAK** file.

Sample notifyd_error.cfg file provided by InfoPrint Manager

```
#
#  Error Log Configuration File
#
log-size = 1024
log-wrap = true
log-severity = debug
log-backup-number = 10
```

When you start the Notification daemon after a shut down, it backs up the first error log by creating an **error.log.BAK** file. With each successive restart, it creates an additional **.BAK** file, adding a timestamp as a fourth qualifier. By default, the **log-backup-number** attribute specifies a value of **10**, meaning that you can have nine back up files with timestamps and the initial **.BAK** file. You can reduce this number as low as `1` or increase it as high as `999`.

For a complete list of the error log attributes that you can customize, see the see the "Attributes for logs" section in "InfoPrint object attributes" chapter of *RICOH InfoPrint Manager: Reference*.

3. Once you have finished defining your options, save and close the **/var/pd/notifyd_error.cfg** file.

   Note

   If you do not need the customization change to take effect immediately, you can skip the final two steps and just wait until the next time the Linux system where the InfoPrint Manager server is installed on is restarted.

4. From an Linux command line, specify `stopnotd` to shut down the Notification daemon so that these customization changes can take effect.
5. Enter `startnotd` to restart the Notification daemon.

Setting InfoPrint Manager environment variables

InfoPrint Manager uses environment variables:

- To define its environment in the system
- To define the objects managed by and options available in the InfoPrint Manager Management Interface

Environment variables are strings that have a name and a value in the form of **NAME=**`value`. See *RICOH InfoPrint Manager: Reference* for a complete list of InfoPrint Manager environment variables that affect InfoPrint Manager.

Preparing to work with InfoPrint Manager environment variables

Read through theses topics to become familiar with the environment-variable structure and terms.

Where are environment variables found?

InfoPrint Manager environment variables can be found in these files or directories:

1

/etc/environment

This file contains environment variables that apply to all sessions for all users on a given system. InfoPrint Manager does not set any environment variables using this file, however you can manually set various variables there to be used by InfoPrint Manager.

/etc/profile.d/ipm_environment.sh

The **/etc/profile.d/ipm_environment.sh** contains environment variables that apply to all the user's Linux sessions. The values in this file override the values in `/etc/environment`.

/etc/ipm/profile.d

You can create custom shell scripts in **/etc/ipm/profile.d** to change your InfoPrint Manager server environment. All shell scripts must have the `.sh` extension. The shell scripts must be readable by the **$IPM_USER** user. The environment values set in those scripts override the values in `/etc/environment` and/or **/etc/profile.d/ipm_environment.sh**.

Do not:

- use "exit" statements inside the shell scripts
- invoke programs and/or use shell script statements that might block the script
- override or remove the InfoPrint Manager environment variables
- remove InfoPrint Manager entries from PATH

The shell scripts are sourced alphabetically.

Note

- **/etc/ipm/profile.d** applies to InfoPrint Manager server and MVS Download receivers only.

How are environment variables set?

You can use these methods to set or change values for environment variables:

- If you are an administrator, you can set Linux environment variables and InfoPrint Manager environment variables in the **/etc/environment** file.

  Settings in these files apply to all sessions for all users, unless users override them through their **/etc/profile.d/ipm_environment.sh** files or from the command line.
- You can modify your **/etc/profile.d/ipm_environment.sh** file in your home directory by using a Linux editor. Settings in the **/etc/profile.d/ipm_environment.sh** file override settings in the **/etc/environment** file. These settings apply only to you and are in effect for all Linux sessions, unless you override the settings from the command line.
- You can enter one of these commands on the command line to set environment variables for a single Linux session:
  - In the Bash shell or the Bourne shell, use the **export** command.

Command line settings override settings in the **/etc/environment** file and **/etc/profile.d/ipm_environment.sh** file. These settings apply only to you and are in effect only until you exit the Linux session from which you issued the command.

You cannot change the environment of a running process. If you change the value of an environment variable, you must restart any process that was started before the change if the change is to take effect for that process. For example, an InfoPrint Manager server or an MVS Download receiver is a process. If you change any of the **/etc/environment** or **/etc/profile.d/ipm_environment.sh** files, and/or add, change, or remove scripts from **/etc/ipm/profile.d**, you must stop the server or the MVS Download receiver and start it again for the change to take effect.

What restrictions apply?

These restrictions apply when you create or define values for environment variables:

- Ensure that newly created environment variables do not conflict with standard variables, such as **MAIL**, **PS1**, **PS2**, and **IFS**. See the operating system documentation for further information.
- Ensure that the information in the `/etc/profile.d/ipm_environment.sh` file is in the **NAME=**`value` format. The file is not a shell script and does not accept data in any format other than the **NAME=**`value` format.
- Do not use variables to represent values in the `/etc/profile.d/ipm_environment.sh` file. For example:

```
PATH=$PATH:/path1:/path2
```

  is not a valid entry in the file. Instead of `$PATH`, enter a full path name.
- The format in which you specify the value for any variable that defines directory paths determines whether your value is appended to the current value or overrides it.
  - This format overrides the current value:

```
export PATH=/path1:/path2
```

    If you do not include the InfoPrint Manager directories in the value you specify, you will not have automatic access to InfoPrint Manager.
  - This format appends your value to the current value:

```
export PATH=$PATH:/path1:/path2
```

Querying environment variables

You can use one of these commands to query the value of an environment variable:

- In any shell, use the **echo** command.
- In the Korn shell, use the **print** command.

For example, to determine the default logical printer, enter:

```
echo $PDPRINTER
```

Note

InfoPrint Manager uses default values for some InfoPrint Manager environment variables without setting the environment variable. When you query those environment variables and you have not set a value, the response is a blank line.

To query all environment variables that have values set, enter: `env`

Editing the /etc/profile.d/ipm_environment.sh file

To set an environment variable in the **/etc/profile.d/ipm_environment.sh** file, you need to edit the file with a Linux editor. See How are environment variables set?, p. 162 and What restrictions apply?, p. 163 for additional information.

This example shows how to set the **PDPRINTER** environment variable in the **/etc/profile.d/ipm_environment.sh** file using a Linux editor:

1. Enter this command to change to the **/etc/profile.d/** directory:

```
cd /etc/profile.d/
```

2. Enter this command:

```
vi ipm_environment.sh
```

   A typical **/etc/profile.d/ipm_environment.sh** file contains lines similar to this:

```
 # /etc/profile.d/ipm_environment

#only set PATH if it does not exist(add pd)
if ! echo ${PATH} | /bin/grep -q /usr/lpp/pd/bin ; then
   export PATH=/usr/lpp/pd/bin:${PATH}
fi

#only set PATH if it does not exist(add psf)
if ! echo ${PATH} | /bin/grep -q /usr/lpp/psf/bin ; then
   export PATH=/usr/lpp/psf/bin:${PATH}
fi


if ! echo ${PDBASE} | /bin/grep -q /var/pd ; then
   export PDBASE=/var/pd
fi

if ! echo ${NLSPATH} | /bin/grep -q /usr/lib/locale/%L/LC_MESSAGES/%N ; then
   export NLSPATH=/usr/lib/locale/%L/LC_MESSAGES/%N:$NLSPATH
fi
```

3. Insert or modify this line in the **/etc/profile.d/ipm_environment.sh** file:

```
export PDPRINTER=LogicalDestinationName
```

   where `LogicalDestinationName` is the name of the logical printer you want to set as the default logical printer for all users.

4. Save the change in plain text format and exit from the editor.
5. Make the change effective by entering the command:

```
. ./ipm_environment.sh
```

6. Verify the change by entering the command:

```
echo $PDPRINTER
```

The logical destination name you entered displays.

Setting an environment variable from the command line

To set the value of the environment variable **PDPRINTER** to `Printer3` for a single session, enter one of these commands:

- In the Bash shell or the Bourne shell, enter:

```
export PDPRINTER=Printer3
```

- In the C shell, enter:

```
setenv PDPRINTER Printer3
```

Verifying the setting of environment variables

You can verify the setting of environment variables that InfoPrint set or appended during InfoPrint installation by doing the following procedures.

Verifying the setting of the PATH environment variable

During the installation process, InfoPrint updates the **PATH** environment variable to include the directory location of the InfoPrint executable files; InfoPrint appends **/usr/lpp/pd/bin** and **/usr/lpp/psf/bin** to the paths already specified.

To verify that the **PATH** environment variable is set and contains **/usr/lpp/pd/bin** and **/usr/lpp/psf/bin**, enter the command:

```
echo $PATH
```

Verifying the setting of the PDBASE environment variable

During the installation process, InfoPrint adds the InfoPrint **PDBASE** environment variable to the **/etc/profile.d/ipm_environment.sh** file and specifies **/var/pd** as the working directory where InfoPrint stores its object code.

To verify that the **PDBASE** environment variable is set and contains **/var/pd**, enter the command:

```
echo $PDBASE
```

Monitoring memory and space on your InfoPrint Manager server

InfoPrint Manager provides a method for monitoring memory use on your InfoPrint Manager server. This support also provides you the opportunity to define and establish recovery options through exit programs or shell scripts that you can customize and add to the system.

The system allocates space for the printing and monitoring tasks required in an InfoPrint environment through different file systems. A *file system* is a structure of parent directories, subdirectories, and files. A *directory* is a unique type of file that contains only the information need to access files or other directories. Directories contain files, subdirectories, or a combination of both. The directory containing a subdirectory is the parent directory.

On occasion, you might have to make sure that work files used in printing jobs do not consume too much space on your file systems. For example, PostScript or TIFF job tickets that are ripped for printing might leave work files in the **/var/psf** file system. Over time, these files can reduce system performance and eventually stop the system. Develop a maintenance strategy with your service representative.

Monitoring memory use on your InfoPrint Manager server

By default, the **Memory Usage** tab of the server property notebook is left blank; you must set these values and create the customized shell scripts or programs to activate this support. You can access this support from the InfoPrint Manager Administration GUI:

1. Select **Server** → **Properties**.
2. Click the **Show more** button.
3. Select the **Memory Usage** tab.

From the **Memory Usage** page, you can specify a low memory threshold percentage and a high memory threshold percentage that cause InfoPrint to invoke a customized exit program or shell script that you can write for your print installation. For example, if the server's system memory usage goes above 60% (the value for the **upper memory usage threshold** field), InfoPrint invokes a user-customized shell script (the `/localtools/uplimit` value for the **Upper memory usage exit** field) that contains specific recovery actions for your print installation. If the server's system memory usage drops below 59% (the value for the **lower memory usage threshold** field), InfoPrint calls a user-customized shell script (the `/localtools/lowlimit` value for the **Lower memory usage exit** field) that contains specific recovery actions for your print installation. This exit usually restores the previous state of the server, undoing temporary changes made by the upper threshold exit.

The customized exit programs or shell scripts can do as little or as much as you specify. For example, you might write a script to disable a certain set of printers. The script could also move jobs to other servers. Both the **lower-memory-usage-exit** and **upper-memory-usage-exit** server attributes allow you to substitute values in the commands that are generated to run the exit program. For example, `%s` is equivalent to *server name* and `%t` is equivalent to *threshold value*. If a particular exit attribute contains a value of `memprob %s` in the server **server1**, then when InfoPrint invokes the memory usage shell script it passes `server1` as the first argument.

For example, a shell script named `uplimit` might:

1. Prevent new jobs from arriving at a logical destination by disabling that InfoPrint object.
2. Prevent the listing of jobs so that queries of the current backlog do not add to the memory constraints.
3. Ensure that any actual destinations (in this case, the three pooled actual destinations named `ad`*n*) are re-enabled, so that they can reduce any backlog of jobs that could be causing the problem.

The `uplimit` script might look like this:

```
#!/bin/ksh
```

```
pddisable prt1-ld
pdset -cserv -xdisallow-list-obj-class=job server1
pdenable ad1 ad2 ad3
```

The `lowlimit` script might look like this:

1. Re-enable the logical destination so that new jobs are being accepted.
2. Reset the server to allow the listing of job queries.

The `lowlimit` script might look like this:

```
#!/bin/ksh
pdenable prt1-ld
pdset -cserv -xdisallow-list-obj-class== server1
```

Note

You can use the **Queries to Prevent** section on the **Memory Usage** page to both add and remove the listing of any queries against a specified InfoPrint object, such as **Jobs**. The **Jobs** object is the most useful to restrict, but this should be done cautiously, because once it has been set, you cannot query any jobs from either the InfoPrint Manager Administration Interface or the InfoPrint Manager Operations GUI.

Note that if the shell scripts or programs are not in the system path, you must specify the fully qualified path for the shell scripts that you have customized. If you misspell a shell script name or refer to a file that does not exist, InfoPrint lets you change the setting. For more information about the **lower-memory-usage-threshold**, **lower-memory-usage-exit**, **upper-memory-usage-threshold**, and **upper-memory-usage-exit** server attributes that support this feature, see *RICOH InfoPrint Manager: Reference*.

Once you have selected the appropriate values, click **OK** to apply this change and close the **Server Properties** dialog.

Configuring Tomcat memory allocation

Encrypted files require substantial memory allocation on the system hosting InfoPrint Manager. The exact amount of memory depends on the number and structure of the files.

If problems occur when viewing files exceeding 2 GB in size, update the **setenv** configuration file at `/var/pd/ipmws/bin/setenv.sh`. Adjust this option:

```
set CATALINA_OPTS=-server -Xms2048m -Xmx2048m
```

to a size appropriate for your configuration. The `-Xmxset` maximum Java heap size and the `-Xmsset` initial Java heap size parameters ensure that the Java virtual machine does not run out of memory. Both `-Xmx` and `-Xms` should be set to the same value.

Monitoring file system use (automatically)

InfoPrint Manager includes support to help you monitor file system usage. The InfoPrint Manager server polls these critical file systems at regularly scheduled intervals, and it increases the polling as the usage increases. Initially, the file system is polled for usage every 16 minutes.

Here is the polling schedule:

80% usage

Every 8 minutes

90% usage

Every 4 minutes

95% usage

Every 2 minutes

97.5% usage

Every minute

If the usage for a file system goes above the InfoPrint Manager pre-defined threshold of 80%, InfoPrint Manager logs a message and generates a **disk-space-low** notification event. As usage increases above each threshold, more messages and **disk-space-low** events are logged and generated. See Configuring your notification profile for disk-space-low events, p. 168 for information about how to configure your notification profile so you can be notified of the **disk-space-low** event.

Each time usages goes above the 80, 90, 95, and 97.5 percent usage levels, the **disk-space-low** event is generated and logged. However, InfoPrint Manager only issues one event for each usage level.

Note

The threshold is reset if the file system usage drops below the previous threshold; for example, when the file system usage drops to 70% after an 80% **disk-space-low** event, it is reset.

Configuring your notification profile for disk-space-low events

It is recommended that you include both the **disk-space-low** and **out-of-disk-space** events in your notification profiles. The InfoPrint Manager server sends **out-of-disk-space** notifications by default, so that when your file system usage reaches 100%, this event is generated. You can modify the server notification profile to add the **disk-space-low** event. However, the messages from the file system usage monitoring are always logged; regardless of the events in the server notification profile.

To add a **disk-space-low** event notification, open the Administration GUI and select **Server→Properties→Other→Notification Profile** and click **Add**. Add the **disk-space-low** event to the notification profile. You must also select the notification method to use on the same panel.

InfoPrint Manager monitors these file systems:

- /var/pddir/default_cell (NAMESPACE) PDNAMESPACE
- `/var/pd(PDBASE)`
- `/var/psm`
- `/var/psf`
- `/var/psf/segments`
- `/var`
- `/var/spool/lpd`

- `/tmp`

Monitoring file system use (manually)

The next section describes how to manually monitor file system usage.

Using the disk usage tool

The **Disk Usage Analyzer** icon on the front panel of the Linux allows you to manually display information about total space and available space on a file system.

Using the df command

You can see how your system uses space in a single file system by using the **df** command. To manually display the space on the file system where your current directory is located, enter this command on the command line:

```
df .
```

This system displays this:

```
Filesystem     1K-blocks       Used    Available   Use% Mounted on
/dev/sda1       495844         31142       439102     7% /boot
```

File systems and directories to clean up

Certain file systems and directories tend to fill up. You should check, and if necessary, clean up these file systems and directories regularly. See Customizing error logs in the InfoPrint Manager server, p. 159 for more information that will help reduce the amount of clean up that is necessary.

File systems used by all InfoPrint installations

If either the **/var/psf** or the **/var/pd** file system is more than 85% full, you should contact your InfoPrint service representative to determine whether you need to clean out files or modify any configuration settings. You might consider removing some files, such as large error or accounting logs. You can find these files in the following locations:

- Server accounting logs: `/var/pd/servername/accounting.logs`
  For more information about accounting logs, see How do you manage the contents of the InfoPrint server accounting logs?, p. 99 in Gathering accounting and auditing data about print jobs, p. 94.
- PSF accounting logs: `/var/psf/`. See How do you manage the contents of the PSF DSS accounting data file?, p. 107.
- Error logs: `/var/pd/server.BAK/error.log.BAK.timestamp`

If you want to customize your log settings, see Customizing error logs in the InfoPrint Manager server, p. 159.

Segment Size parameter

The **Default SEGMENT SIZE** parameter affects the disk I/O performance of the **/var/ psf/segments** file system. This parameter determines the maximum size of any temporary file created by the Input Manager component that is stored in **/var/ psf/segments**. This setting determines the number of temporary files that Input Manager requires for all print jobs submitted to InfoPrint Manager: the smaller the setting, the more files required to process the AFP input data. This parameter can be set by using IPMMI: **InfoPrint Manager→Options→Customize the Input Manager**

For Basic Server configurations, the default setting of 5000 KB per segment can be raised to reduce the disk I/O usage associated with reading and writing data to the **/var/psf/segments** file system.

For Advanced Server configurations, the default setting of 1000 KB per segment can be raised to reduce the disk I/O usage associated with reading and writing data to the **/var/psf/segments** file system; however, the default setting usually works well with most AFP production applications.

Using InfoPrint Manager with SELinux set in enforcing mode

This section describes special considerations for working with InfoPrint Manager with SELinux set in enforcing mode.

Running InfoPrint Manager with SELinux set in enforcing mode

When you want to run InfoPrint Manager with SELinux set in enforcing mode on a Red Hat Enterprise Linux, you must create and install a SELinux policy module:

1. Install `checkpolicy` and `policycoreutils` rpm files if not already installed:

```
yum install checkpolicy policycoreutils
```

2. Generate the SELinux policy module:

```
cd /tmp
cp /usr/lpp/pd/selinux/ipm.te /tmp
/usr/bin/checkmodule -M -m -o ipm.mod ipm.te
/usr/bin/semodule_package -o ipm.pp -m ipm.mod
```

3. Install the new SELinux policy module:

```
semodule -i ipm.pp
```

4. Remove temporary files:

```
rm -f ipm.pp ipm.mod ipm.te
```

Note

If you want to configure SELinux at a component level you may use specific policy modules rather then applying `ipm.te`. The InfoPrint Manager installation provides policy modules for specific functions so you can create and install them according to your need.

- `ipmmail.te` - configures SELinux for Email DSS Configuring InfoPrint Email, p. 41
- `ipmcups.te` - configures SELinux for CUPS DSS Using CUPS DSS with SELinux set in enforcing mode, p. 52

Reverting the SELinux changes for InfoPrint Manager

To revert the SELinux changes for infoPrint Manager, run:

```
semodule -r ipm.pp
```


Changing the server host name and pdserver name

Use these procedures to change the host name of the system on which your InfoPrint Manager server runs, or to rename the InfoPrint Manager pdserver.

Changing the server host name

Changing the host name of the server where InfoPrint Manager is running affects both the system and the license fingerprint, which prevents InfoPrint Manager from running. Before proceeding, make sure the following actions have been taken:

- Create a full backup of the InfoPrint Manager system. Then, run a pdcapture and store the resulting file on a separate system. Also, use the ISMU to create a restorable archive of the current InfoPrint Manager system objects.
- Ensure you have a complete list of your License EIDs, which will be needed to generate a new license key once the system fingerprint stabilizes under the new host name.
- Generate a new license key using the new host name. This might require temporarily stopping InfoPrint Manager and changing the host name in order to retrieve the new fingerprint using the License Key Management Application. Go to the self-service Ricoh website https://dl.ricohsoftware.com and use the new fingerprint to generate the new license key for the product. For more information, see the "Revoking the License Key" topic in *RICOH InfoPrint Manager for Linux: Getting Started*.

To change the server host name, follow these instructions:

1. Modify the old host name of the server to the new host name of the server in the `/etc/hostname` file.
2. If necessary, change the old host name with the new server name in the `/etc/hosts` file.
3. For the changes to take effect, restart the machine.

Note

It is not necessary to change the pdserver name.

Changing the pdserver name

To change the pdserver name, follow these steps:

1. Change the old server name with the new server name in the `/etc/rc.pd.servers` file, if available.
2. Stop all running pdservers. For interoperating systems, stop all pdservers from all interoperated machines.
3. From a command line terminal, clear the namespace:
   `# clrfstns /var/pddir/default_cell`

4. Reset the fst.ports:
   # pdinitports
5. Remove the /var/lib/pd/pdserver.conf file:
   # rm /var/lib/pd/pdserver.conf
6. Remove the old_server_name file from the /var/pd/old_server_name/pdb/spooler/ directory:
   # rm /var/pd/old_server_name/pdb/spooler/old_server_name
7. Rename the /var/pd/old_server_name folder to /var/pd/new_server_name:
   # mv /var/pd/old_server_name /var/pd/new_server_name
8. If /var/pd/new_server_name contains one or more files that end with .dev, symlink /var/pd/new_server_name as /var/pd/old_server_name:
   # ln -s /var/pd/new_server_name /var/pd/old_server_name

   Note

   You can skip this step and remove all printers that have .dev files in /var/pd/new_server_name and recreate deleted printers after step 9.
9. Start the new_server_name pdserver or reboot your computer.

Using the InfoPrint Manager System Migration Utility (ISMU)

You can use the InfoPrint System Migration Utility (ISMU) for backing up, restoring, or migrating your InfoPrint Manager environment. This utility is based on the existing server archive functionality, and it backs up your current configuration from a specific level of InfoPrint Manager and restores it on the same InfoPrint Manager environment.

Using this utility, you can back up any level of your InfoPrint Manager environment.

When using this utility for restoring purposes, we recommend that you have the InfoPrint Manager and the backup archive at the same level. If they are not at the same level and you proceed with the restoration or migration process, two warning messages are displayed, stating that you are not on the same level and the results might be unpredictable. This still enables you to run the backup or restore process at slightly different levels (for example x.2.0.255 versus x.2.0.256).

If you need to perform a migration between different levels of InfoPrint Manager, follow these steps:

- Perform the backup and restore required operations, as documented in the appropriate *RICOH InfoPrint Manager: Procedures* for your operating system.
- If necessary, upgrade InfoPrint Manager to the required level.

Note

- The **ISMU** utility was not intended to perform crossover migrations, such as migrating the AIX or Linux environment to the Windows environment and backwards. However, migrating objects that are commonly supported across platforms does work, but it does not successfully migrate platform-specific objects (such as AIX DSS to Linux or Windows).
- You can run the **ISMU** utility directly from the InfoPrint Manager Management Interface.
- Only a person with root user authority or the InfoPrint Manager user can run this command.

Prerequisites

To run the **ISMU** utility, make sure you are aware of these requirements:

- You must have administrative rights on the system where InfoPrint Manager is running.
- InfoPrint Manager must be installed and running at all times.
- These programs must be installed on your machine:
  - **tar, gzip, gunzip**, and **diff**
- **ISMU** also requires Oracle Java Runtime Environment 7.0 or later installed on your computer.

Note

- You can only run **ISMU** locally. This utility does not have network functionality.

Migrating your InfoPrint Manager settings

Before migrating your InfoPrint Manager settings and objects, read the following sections to find out more about which files and settings are automatically backed up and which are not. For the files and settings that are not automatically backed up, you should use the manual backup procedure.

Migrating your InfoPrint Manager settings automatically

This section lists the files and settings that **ISMU** automatically backs up and restores.

- Custom 64xx GRID mapping tables
- Custom ICU tables
- Custom notification and PSF exits
- Custom or modified halftones
- Federated authentication settings
- FST security settings
- IPP Gateway (IPPGW) configuration
- MVS download receivers created with the InfoPrint Manager GUIs and the `mvsprpsm.sh` script
- PSF printer profiles
- Printer configuration files
- Transform configuration files
- /var/pd/linux.models
- Web server configuration

Automatic tasks during migration

This section details all sections of InfoPrint Manager that are automatically backed up during the migration process.

Custom notification and PSF exits

The program automatically backs up and restores any custom exits or PSF exits it finds including these two server exits:

```
lower-memory-usage-exit
```

```
upper-memory-usage-exit
```

FST security settings

ISMU automatically backs up and restores all the files within the `/var/pddir/default_cell/acl` directory.

Custom or modified halftones

ISMU backs up and restores all files from the `/usr/lpp/psf/config/custom` directory.

Transform configuration files

By default, InfoPrint Manager comes with these transforms: gif2afp, img2afp, jpeg2afp, pcl2afp, ps2afp, tiff2afp, and xml2afp. There are two other transforms that are installed separately: afp2pdf and sap2afp. The configuration files for all the transforms are located in `/usr/lpp/psf/<transform_name_directory>`, and if you modified them, then **ISMU** copies the modified files to the new server.

Custom 64xx GRID mapping tables

ISMU copies all the customized ***.grd** files from `/var/psf/<printer_name>`.

Custom ICU tables

ISMU copies all the customized **.cnv** files from `/usr/lpp/psf/unicode/data/mappings`.

By default it contains 6 **.cnv** files:

- icudt24b_ibm-1388-sap2afp.cnv
- icudt24b_ibm-933-sap2afp.cnv
- icudt24b_ibm-937-sap2afp.cnv
- icudt24b_ibm-939-sap2afp.cnv
- icudt24b_ibm-1399-sap2afp.cnv
- icudt24b_IBM-943C.cnv

If these files exist, **ISMU** copies and restores all the files from `/usr/lpp/psf/afp2pdf/cnv`.

Migrating your InfoPrint Manager settings manually

This list contains parts that **ISMU** does not automatically back up and restore.

- Associated installed products and/or InfoPrint Manager features
- Custom ACIF exits
- Custom auxiliary-sheets
- Drivers (backends and ppd files)
- LDAP security settings
- Samba for Linux software and configuration for `/ipdata` and interoperating
- Server Admin/Operator GUI configuration/customization
- Third-party applications
- Web Server databases

Specific tasks during migration

During the migration process, it is very important to make sure that all objects are correctly backed up and your initial configuration is correctly restored. Therefore, these sections list all important areas where you should take extra steps for an accurate migration process.

Samba software and configuration for `/ipdata` and interoperating

If the source **pdserver** that must be migrated is interoperated with another server, you must set up the interoperating environment on the new server.

To find more information about this procedure, follow the instructions from Configuring InfoPrint systems for interoperability, p. 177.

If job ticketing is used, you must configure the `/ipdata` folder. For more information about this task, see the *Using the InfoPrint Submit Express Client* section in *RICOH InfoPrint Manager for Linux: Getting Started*.

Channel configuration

The channel configuration is no longer supported. The printers will be created, but they will not start.

Associated installed products and InfoPrint Manager features

PPFA

If you have this feature installed, you must install it from the InfoPrint PPFA Feature DVD on the new server. If you have customized PPFA source files, copy them to the new server for later processing.

AFP Resource Installer

This is an independent Windows application. Make sure that the resources used by the InfoPrint Manager objects are available on the new system.

Backends and PPD files

ISMU does not automatically back up and restore the Linux drivers. This operation has to be done manually.

Custom ACIF exits

You need to manually back up and restore any customized ACIF exits that you created. These exits are located at `/usr/lpp/psf/bin` (**apk***, **asciinp***, and **dbblank** files) and the source files are located at `/usr/lpp/psf/acif/`.

Custom auxiliary-sheets

If the output shows more than the default sheets that come with InfoPrint Manager (64xx, accounting-log, blank, brief, full, job-ticket, and none), **ISMU** automatically copies the customized sheets to the new server. The customized user-exits from `<install_path>\usr\lpp\psf\exits\psf` are also copied.

Registry data about psfin

You need to check first if the **psfin** data contains the default values. To do it, follow these steps:

1. Open the Management Console.
2. Go to **Options** → **Customize Input Manager**.
3. Check if these values are the default ones:
   - Directory for segment lists: `\var\psf\seglist`
   - Directory for segments and percent usage threshold: `\var\psf\segments 90`
   - Workspace management: automatic
   - Default segment size: 5000
   - Percentage of workspace for retained segments: 5
   - Enable trace: unchecked
4. If the values match the default ones, no work needs to be done; otherwise, modify the defaults on the new server to match the ones from the source server.

Web Server databases

After you stop both source and target Web Servers, copy the following files to the target Web Server:

- `/var/pd/ipwebgui/data/ipmwebgui.mv.db`
- `/var/pd/wmi/data/ipmwebgui.mv.db`

- Both source and target Web Servers must be at the same level.

Example of migration

Example of using **ISMU** to migrate a pdserver (and its objects) from an existing InfoPrint Manager server to a new InfoPrint Manager server:

1. Export command (run on existing server):

```
ismu -pdname=pdserver -pdarchive -pdtemp=/tmp/server_archive -pdsave_cfg_files
```

2. If necessary, change the pdserver name to the new machine host name on the new InfoPrint Manager server (run on new machine):

```
ismu -pdname=pdserver -pdmodify_to=hostname_of_new_machine -pdtemp=/tmp/server_archive
```

3. Import command (run on new Server):

```
ismu -pdrestore -pdtemp=/home/server_archive -pdrestore_cfg_files
```


Note

This series of commands needs to be run for each **pdserver** you want to migrate to a new InfoPrint Manager server.

Configuring InfoPrint systems for interoperability

This chapter contains these sections:

- Understanding interoperating environments, p. 177
- Configuring a primary Linux server with a secondary Linux server, p. 180
- Configuring a primary Linux server with a secondary AIX server, p. 180
- Configuring a primary Linux server with a secondary Windows server, p. 181

Primary Windows servers:

If you are using a Windows operating system for your primary server, see *RICOH InfoPrint Manager for Windows: Procedures* for interoperability instructions.

Understanding interoperating environments

When you install an InfoPrint Manager server on a system, the install process creates a namespace for that server. The *namespace* is a directory structure where InfoPrint Manager stores the location information of all of your InfoPrint objects, including the server itself, queues and destinations, and the Access Control Lists (ACLs) used by InfoPrint Manager Security.

If you install a second InfoPrint Manager server on another system on your network, that server will create its own namespace and the two servers will operate entirely independent of each other. In some cases, however, you might want your servers to be able to work together. To make InfoPrint Manager servers work together, you set up an *interoperating environment* in which two or more InfoPrint Manager servers share the same namespace. The namespace is located in a directory on one of the servers and the other servers are granted permissions that allow them to use that directory.

When configuring InfoPrint Manager servers to support interoperability, make sure that these two rules are followed:

1. Each InfoPrint Manager server must be accessible by all of the other InfoPrint Manager servers using the same host name.

   For example, if **Server1** is an InfoPrint Manager server with a host name of `server.local`, then all other InfoPrint Manager servers must use `server.local` to access **Server1**.

2. The host name for each InfoPrint Manager server must be the host name that is identified for the first (or primary) network card that is defined to that server.

Reasons for setting up an interoperating environment

Not every printing system needs multiple InfoPrint servers. In fact, in many environments one InfoPrint server is adequate. However, in print environments that do very high volume printing or that print in remote locations (for example, a main office with branch offices in different parts of the city), interoperating InfoPrint servers can provide important benefits:

- Interoperating servers are much more scalable than single servers.
  - If your print volume increases and you find that you need more processing power or memory to be able to meet the demand, you can add a second server. However, by making the servers interoperate, you still only have one printing system to administer.
  - If your business expands and you add another office, you can add another server to the namespace. You'll have to define any new printers that you add to the network, but you won't have to redefine existing printers or move resources again to be able to use them from the new location.
- Interoperating servers allow you to distribute print server functions across systems.

  Remote locations can have their own local servers, but you can administer them from a central location. This should result in lower costs.

Multiserver configuration examples

There are various ways to set up multiple InfoPrint Manager servers and have them work together. Below are some example configurations and the benefits they provide.

Install two InfoPrint Manager servers in the same physical location.

One of the servers contains all of the InfoPrint print queues and logical destinations, and the other contains all of the actual destinations. In this setup, you balance processing between the two systems by spooling and scheduling on one system and transforming and printing on the other.

Install one InfoPrint Manager server in the main office and one in each branch office.

The namespace is installed on the InfoPrint Manager server in the main office, and the other two servers are set up to share it. All three servers have logical destinations, queues, and actual destinations defined in them, and the users submit their print jobs to their local server. About 90% of the time, the servers handle local print jobs, so it makes sense for each location to have its own server. However, several times each week, the main office generates reports that are needed at the branches. Instead of printing the reports and sending them across town or faxing several hundred pages to each office, they simply send the print job to printers at the branch offices. If necessary, the branch offices can do the same thing.

In addition, because of the common namespace, the system administrator can monitor all of the servers from a central location; if there's a problem at one of the branches, he can investigate it from the main office.

Important issues in interoperating environments

Before you set up an interoperating environment, consider these issues.

- **Software**

All interoperating systems must be at the same InfoPrint Manager software level. Older versions of InfoPrint Manager will not work with newer versions of InfoPrint when interoperating.

- **Backups**
  You should always backup your systems before making any major configuration changes. Follow the recommendations found in *RICOH InfoPrint Manager for Linux: Getting Started* about which file systems should be backed up.
- **How many servers can be interoperated?**
  In theory, you can interoperate as many servers as you would like, however, the performance of the servers will depend upon the primary server hardware. For best performance, use the most powerful system that you have available for the primary server.
- **Which system should contain the namespace?**
  The namespace does not contain large volumes of data (so it does not have to reside on the system with the most disk space) and it does not move a lot of data across the network. However, because all of the other servers will rely on the namespace server, it must be available and reliable. Put the namespace on the system that has the most availability (always running), most reliability (does not have to be rebooted often), and is the most powerful. **The namespace must be located on an AIX or a Linux system.**
- **How can I share the namespace across servers?**
  There are two methods to share the namespace from the primary server, which depends on what operating system the secondary server is. You can do that using Network File Sharing (NFS) when the secondary server is located on the AIX or Linux operating system, or Samba when the secondary system is a Windows operating system. When NFS is used, the exported namespace filesystem by the primary system is mounted by secondary server using the host name. Samba is used to export namespaces from the primary AIX or Linux server to a Windows secondary server. It is also used for the additional `/ipdata` sharing required by Submit Express job preflight.

Primary vs. Secondary server platform	Namespace Share Method	Namespace Share User	ipdata Share User	/ipdata Share User
AIX (primary) - AIX (secondary)	NFS	ipm1 (UID must match)	NFS	ipm1 (UID must match)
AIX (primary) - Linux (secondary)	NFS	ipm1 (UID must match)	NFS	ipm1 (UID must match)
Linux (primary) - AIX (secondary)	NFS	ipm1 (UID must match)	NFS	ipm1 (UID must match)
Linux (primary) - Linux (secondary)	NFS	ipm1 (UID must match)	NFS	ipm1 (UID must match)
AIX (primary) - Windows (secondary)	Samba	ipm1	Samba	ipm1

Linux (primary) - Windows (secondary)	Samba	ipm1	Samba	ipm1
Windows (primary) - Windows (secondary)	Windows Share	Windows Domain User	Windows Share	ipm1

Important

Interoperating multiple InfoPrint Manager Windows servers is supported only when running under Windows Domain controlled environments.

- **InfoPrint Manager Security**

  All of the information that InfoPrint Manager Security uses to protect your printing system (groups and Access Control Lists (ACLs)) is stored in the namespace. When you share the namespace, the groups and ACLs that you set on the namespace server apply to all of the servers; the groups and ACLs that you set on the other servers before they joined the common namespace disappear. As a result, you should wait until you have the interoperating environment set up before you set up any security on the systems.

  Note

  If the Pull Print Feature is installed on one of the servers that share the namespace, you have to recreate the groups and the ACL used by these features.

Configuring a primary Linux server with a secondary AIX server

Follow the appropriate instructions according to your configuration needs:

New secondary InfoPrint Manager for AIX server

If you are installing InfoPrint Manager for the first time on an AIX system, and you want to make the new system a secondary InfoPrint Manager server, see "Installing secondary InfoPrint servers" in *RICOH InfoPrint Manager for AIX: Getting Started* for information about doing this during the installation.

Creating a secondary InfoPrint Manager server from a primary InfoPrint Manager server

If you have two or more independent InfoPrint Manager primary servers that you want to change into secondary servers, see "Migrating from a primary server to a secondary server" in the *RICOH InfoPrint Manager for AIX: Getting Started*.

Configuring a primary Linux server with a secondary Linux server

Follow the appropriate instructions according to your configuration needs:

New secondary InfoPrint Manager for Linux server

If you are installing InfoPrint Manager for the first time on an Linux system, and you want to make the new system a secondary InfoPrint Manager server, see "Installing secondary InfoPrint servers" in *RICOH InfoPrint Manager for Linux: Getting Started* for information about doing this during the installation.

Creating a secondary InfoPrint Manager server from a primary InfoPrint Manager server

If you have two or more independent InfoPrint Manager primary servers that you want to change into secondary servers, see "Migrating from a primary server to a secondary server" in the *RICOH InfoPrint Manager for Linux: Getting Started*.

Configuring a primary Linux server with a secondary Windows server

Important

In order to interoperate InfoPrint Manager for Linux with InfoPrint Manager for Windows:

- The Linux system must have Samba for Linux installed.

  To create mapped drives of Linux directories exported by Samba on Red Hat Enterprise Linux, when SELinux is in enforcing mode and you need interoperability with AIX, Linux, or Windows, you must run this command as root:

```
setsebool -P samba_export_all_rw=1
```

  To restore the previous behavior, run this command as user root:

```
setsebool -P samba_export_all_rw=0
```

  Note

  For more information on Samba, go to www.samba.org.

Samba Server and Client software to interoperate Linux with Windows machines allows Linux servers and workstations to share files and printers with personal computer clients running Windows operating systems. Because Samba for Linux uses industry-standard Microsoft networking protocols, PC clients can access Linux files using Microsoft networking client software. PC users can use remote Linux file systems directly from their machines like local file systems. The namespace will appear on the network just like any other Windows shared folder. Samba for Linux provides these services by implementing the SMB networking protocol to run on NetBIOS over TCP/host name.

By doing this set of procedures on both an InfoPrint Manager for Linux server and an InfoPrint Manager for Windows workstation, you can configure InfoPrint Manager to support interoperability between InfoPrint Manager for Linux servers and InfoPrint Manager for Windows servers.

When configuring InfoPrint Manager for Linux and InfoPrint Manager for Windows servers to support interoperability, make sure that these rules are followed:

1. Each InfoPrint Manager server must be accessible by all the other InfoPrint Manager servers using the same host name.

   For example, if **Server1** is an InfoPrint Manager server with a host name of `server.local`, then all other InfoPrint Manager servers must use `server.local` to access **Server1**.

2. The host name address for each InfoPrint Manager server must be the host name that is identified for the first or primary network card that is defined to that server.
3. All printer objects, whether logical destination, actual destination, or queue, must have unique names if they are used by interoperating InfoPrint Manager servers.

   If the printer objects do not possess unique names, you might have trouble viewing them through the **InfoPrint Manager Interface** from a non-namespace server.
4. InfoPrint Manager Notifications should only run on the primary server and should be stopped on the secondary servers.

LPD service for Linux can be used however, by submitting jobs with the **lpr** command to actual destinations on the Linux server. The **lpr** command can still be used as before with Windows printer destinations, but not with InfoPrint Manager printers.

Setting up the InfoPrint Linux server for interoperability

Perform these procedures on your InfoPrint Linux server to prepare for interoperability with an InfoPrint Windows server. These procedures presume that you have installed InfoPrint Manager for Linux on your Linux server and have logged onto the server as `authorized user`.

Adding users to and modifying permissions for FST security groups

1. Set up the FST Security groups on your Linux IPMMI server by taking the **InfoPrint Manager** → **Security** → **Groups** → **Show Group** path. From the **Single Select List** pop-up dialog, you should see these three groups: **acl_admin**, **admin**, and **oper**.
2. Once you have verified the existence of these groups and other custom groups, you should make sure that users in the **admin** group can issue all the InfoPrint **pd*** commands. From IPMMI, take the **InfoPrint Manager**→ **Security** → **Access Control** → **Access Control for Operations** → **Show Access Control List** path.
3. From the **Single Select List** pop-up dialog, highlight each command and make sure that`GROUP: admin: r**` displays for each command.

   If a particular pd command does not show **admin** group read authority, use the IPMMI **InfoPrint Manager** → **Security** → **Access Control** → **Access Control for Operations** → **Change Access Control List** path to specify `GROUP:admin` in the **Users or Groups** field of the **Change Access Control List** panel for that pd command. Note that the **admin** group requires only read authority for each **pd** command.

   If you have installed InfoPrint Manager on a "clean" machine, the **admin** group will have read authority for all **pd** commands by default.
4. Once complete, return to the first IPMMI panel displayed in step 2.
5. In IPMMI take the **InfoPrint Manager** → **Security** → **Groups** path. Then select `acl_admin` for the **Name** field. Right-click and select **Open to show the Edit Group** dialog. Click **Add** to add a new member to the group. Enter the name of the Windows administrative user, either local or domain, you normally log in to the InfoPrint Manager for Windows server machine as for the **User or Users to Add** field. Select **Member type** as **FST User**.

   For the remainder of this procedure we will refer to this Windows administrative user as `Administrator*`.

Note

The `Administrator` user must be followed by an asterisk (*) to account for the fully qualified host name. (For example, `Administrator@ipwinservermachine.infoprint.com`.)

From the same panel, specify `admin` for the **Name** field. Enter the name of the Windows administrative user you normally log in to the InfoPrint Manager for Windows server machine as for the **User or Users to Add** field. You do not have to do this step for the `oper` group.

6. Once complete, click OK to make sure the changes take effect and return to the first IPMMI panel displayed in step 2.

Configuring Windows servers for interoperability

Some print environments require more than one InfoPrint Manager server to handle their print management. If you are adding one or more InfoPrint Manager for Windows servers to your print environment, this section will tell you what you need to do to make those servers work with the InfoPrint Manager for Linux server.

Note

1. In an interoperating environment that includes an Linux server and Windows servers, the namespace will reside on the Linux server.
2. You can only install one InfoPrint Manager server on a given Windows workstation or server.
3. If your users will be submitting jobs directly to actual destinations, the actual destination and the associated queue and logical destination must be on the same server.
4. Once you have your multiserver system running, if one of your Windows servers shuts down or is restarted, that InfoPrint Manager server will not restart automatically when the system restarts. You must open the Management Console for that server and start it manually.

Granting appropriate user rights to the domain user on all InfoPrint Manager Windows servers when using a different user than ipm1

To make sure that the Windows InfoPrint Manager servers can run as a Windows service using the domain userid at your choice, you must make sure that this userid has the proper level of authority on each Windows InfoPrint Manager server system. This userid should be included in the Administrators group.

1. Log on to one of the preferred systems as a user who is a member of the Domain Administrators group.
2. Click the Windows **Start** button and select **Settings** → **Control Panel**.
3. In the **Control Panel** window, double-click the **Administrative Tools** icon.
4. In the **Administrative Tools** window, double-click the **Local Security Policy** icon.
5. In the right pane of the **Local Security Settings** Window, double-click **Local policies**.
6. Double-click **User Rights Assignment** in the right pane.
7. In the list of user rights, select **Act as part of the operating system** and double-click it.
8. In the **Local Security Policy Setting** dialog, click **Add**.

9. In the **Select Users or Groups** dialog, click the arrow on the drop-down list box and select the domain that your InfoPrint Manager servers exist in.

   Find and select the domain user you created, and click **Add**.
10. Click OK in the **Select Users or Groups** dialog.
11. Click OK in the **Local Security Policy Setting** dialog.
12. Repeat from step 7 for two other user rights: **Create permanent shared objects** and **Log on as a service**.

For working interoperability, follow these steps as root:

1. Add a disabled `Linux` user using the same name as the user running InfoPrint Manager on the Windows server.
2. Add a Samba user for the user created above.

   For example:

```
smbpasswd -a <above_user>
```

3. Create the `default_cell` Samba share by running this command:

```
/usr/lpp/pd/install/fstconn.sh -n /var/pddir/default_cell
```

4. Edit the Samba configuration file, `smb.conf`. For each available share (`default_cell` and/or `ipdata`) add the above Samba user to the list of valid users:

```
valid users = ipm1 windows_user_1 windows_user_2 windows_user_3 ...
```

Creating the mapped drive to the Linux namespace that allows interoperability on Windows

From each InfoPrint Windows server, you must verify that you have access to the namespace on the InfoPrintLinux server. You can perform this action by mapping a network drive, using the default user **`ipm1`**, from the Windows server to shared namespace on the InfoPrint Linux server.

1. From the main Windows view, right-click the **My Network Places** icon, select **Map Network Drive**. In the dialog, use the **Folder** field to specify:

```
\\servername\default_cell
```

   Where *servername* is the name of the InfoPrint Linux server.
2. Check the **Reconnect at Login** check box.
3. Click on the **Connect Using Different User Name** link.
4. A dialog box will open asking for a username and password associated with `ipm1`. Specify the `ipm1` password from the Samba server.
5. Click **OK** when the window opens.
6. Click **Yes** to accept the current login.

   If the drive mapped correctly, a new window should open to the mapped drive's contents, and **Map Network Drive** windows should be gone.
7. Click **Finish** to map the drive.

Note

Do not follow steps 2 through 6 when logged as `ipm1` user. Map the network drive manually before starting InfoPrint Manager Management Console when logged as `ipm1` user.

Set the drive letter for this network drive as specified in step 4, in the Configuring interoperability on the InfoPrint Windows servers, p. 185 section.

The password for the `ipm1` Windows user must be the same as the password for the `ipm1` SAMBA user.

Configuring interoperability on the InfoPrint Windows servers

Make sure that you complete this procedure on all InfoPrint Windows servers.

On all Windows servers, do these steps:

1. From the **Start** menu, click **Programs** → **InfoPrint Manager** → **Management Console**.
2. Click **Edit** → **Service Configuration**.
3. In the **Settings** group box, clear both the **Run notification server on this system** and **Run InfoPrint Manager LPD service** check boxes.

   Note

   A notification server was installed with each of your InfoPrint Manager servers. However, now that the namespace is shared, you only need to run one of them . The notification server must be run on the namespace server. Disable all but one of the InfoPrint Manager Notification servers.
4. Clear the **Base namespace path** field and type the drive letter that you want to use to link to the namespace followed by a colon (for example, type `x:`). Use the same drive letter that you used when mapping the network drive.
5. In the **Remote namespace folder** field, enter the Universal Naming Convention (UNC) name of the shared directory on the namespace server in this format: *namespaceserver**sharename*. The sharename of the namespace on anLinux server is *default_cell*. For this example, `\\prince\var\pddir\default_cell` .
6. Click **OK**.
7. Restart all of the InfoPrint Manager servers by selecting **File** → **Start Server** on each Management Console.
8. Verify that all of the servers are now sharing the namespace.

   - Log on to any one of the machines that holds an InfoPrint Manager server, open a command prompt.
   - Start the InfoPrint Manager Administration Interface.
   - Click **Server** → **Show All Servers**.

   You should see a list of all of the servers that are using the namespace. If any of them do not appear, stop the servers and check the settings that you changed in step 2 for that server.

Using the LPD Gateway with InfoPrint Manager for Linux

You can use the InfoPrint Manager for Linux line printer daemon (**ipmlpd**) to receive jobs through the **LPD Gateway** to an InfoPrint Linux server. **InfoPrint Manager LPD** is a subsystem that lets you submit

print jobs from other operating systems using a Line Printer (**LPR**) client. **InfoPrint Manager LPD** submits the files it receives directly to an InfoPrint Manager destination and supports the **-o** options that are available with the some **LPR** clients. (**-o** options allow you to specify advanced InfoPrint Manager print options, such as form definitions and page definitions. For more information about using **-o** options with print commands, see *RICOH InfoPrint Manager: Reference*.)

To configure to use the **InfoPrint Manager LPD** protocol you need to verify if any **LPD** daemons or processes that do not belong to InfoPrint Manager (for example, the Common UNIX Printing System [CUPS] **LPD** daemon) are running on the same system, stop them.

Important

Do not uninstall **CUPS**, just stop the **LPD CUPS** daemon on your Linux server.

The InfoPrint Manager for Linux Line Printer Daemon uses **port 515**.

The **LPD** protocol defines a print command called **lpr**. You can use the **lpr** command, or several other commands that use the **LPD** protocol, to submit jobs from AIX, Linux or Windows systems to InfoPrint Manager for Linux **LPD** destinations. You can also send jobs using the **LPD** protocol from other systems, such as z/OS. Syntax and available options vary from one print command to another and from one version of lpr to another. You might have to adapt examples of print commands for your own system.

You might also have to configure the sending system to compensate for options that your print command does not provide. For example, if the print command does not have an option for specifying a print server, you must create a remote print queue on the sending system with the same name as the **LPD** destination.

To start and stop InfoPrint Manager Linux **LPD**, you can use the **startlpd** and **stoplpd**. For details on how to use these commands, see *RICOH InfoPrint Manager: Reference*.

Submitting print jobs through the LPD from an AIX system

To submit jobs from the **LPD** on an AIX system, you can use the **lpr** command, which accepts the **-o** options.

When you submit print jobs using the **LPR** client, you need to set up a remote queue on the local system. To accomplish this task, see your AIX system documentation. Once you have defined a remote queue, it identifies the InfoPrint Manager remote server and the name of the InfoPrint destination (either logical or actual) on that InfoPrint Manager remote server. The **LPD** submits the job to the InfoPrint print spool. We recommend that you specify the **LPD** subsystem in the **/etc/inittab** file to make sure that it starts on a system reboot and is always available.

When you enter the **lpr** command, you must specify the option that designates *binary* or *already-formatted* data. For example, you must use the **-l** (lowercase **L**) option on the **lpr**.

If you are printing from an InfoPrint Manager AIX server or from an AIX server where InfoPrint Manager client is installed, you should use the InfoPrint **pdpr** command to submit jobs directly to the InfoPrint print spool.

Submitting print jobs through the LPD from Linux, and other UNIX systems

You can use any **LPR** client, but to be able to specify advanced InfoPrint options on the **-o** flag, you must make sure that your **LPR** client supports **-o** options. Sample code for an InfoPrint **LPR** client called **lprafp** is provided with InfoPrint Manager for Linux and is available from the **/usr/lpp/psf/utils/lpr** directory. This directory contains sample code of the **lprafp** client for UNIX platforms. This code must be compiled as described in the **README** file, or you can download the binary files from this link: https://dl.ricohsoftware.com/downloads/375bc3fe-3080-401f-91c4-2b8adb607f89.

Regardless of what **LPD** subsystem you decide to use, we recommend that you specify it in the **/etc/inittab** file to make sure that the client starts on a system reboot and is always available.

Submitting print jobs from a non- UNIX system with the InfoPrint (lprafp) LPR client

When you submit print jobs from the InfoPrint **LPR** client on a non-Linux UNIX system, all you need to specify is the InfoPrint Manager server and the name of an InfoPrint destination (either logical or actual). The **LPD** submits the job to the InfoPrint print spool. If you enter the **LPR** shell script with the **-A** flag, it invokes the **lprafp** command program (**lprafp.c**), which is documented in the *RICOH InfoPrint Manager: Reference*. If you enter the **LPR** shell script without the **-A** flag, it invokes the Linux system's LP command, which does not support the InfoPrint **-o** options.

Submitting print jobs through the LPD from a Windows system

You can use any **LPR** client, but to be able to specify advanced InfoPrint options on the **-o** flag, you must make sure that your **LPR** client supports **-o** options. Sample code for an **LPR** client called **lprafp** is provided for download on the Ricoh website. To obtain and install this **LPR** client, see Install and configure lprafp for Windows from the web, p. 187.

Note

If you use another **LPR** client, make sure that you specify the flag for specifying binary data. For example, on the Microsoft TCP/IP Printing **LPD** (the default **LPR** on Windows operating systems), this flag is the `–o1` or the `–o l` (lowercase **L**) option, depending upon your Windows operating system.

Install and configure lprafp for Windows from the web

For Windows operating system clients, the **lprafp** sample code package is available for free download from the Ricoh website at https://dl.ricohsoftware.com/downloads/375bc3fe-3080-401f-91c4-2b8adb607f89.

Using the Internet Printing Protocol (IPP) with InfoPrint Manager for Linux

The Internet Printing Protocol (IPP) is an application protocol that uses the Hypertext Transfer Protocol (HTTP) to allow you to print your documents to any IPP-enabled printer whose web address (or Uniform

Resource Identifier-URI) you know. You can use IPP to send print data over both LANs and the Internet so users who are directly connected to your LAN as well as those who work remotely have access to the same printers. InfoPrint Manager for Linux provides one way to configure your system to print using IPP.

Printing through the IPP Gateway

The **InfoPrint Manager IPP Gateway** is a service in InfoPrint Manager that lets you print using IPP even if you do not have an IPP-enabled printer, as long as you (and your users) have IPP client software installed on your workstation.

Note

- The Windows system comes with IPP client software built in. Users on other platforms can download IPP client software. The Printer Working Group, the organization that is leading the development of IPP, provides a list of IPP client software that you can use to find an appropriate client at http://www.pwg.org/ipp/.

Once you have an IPP client installed, to set up the gateway:

1. Start the **IPP Gateway** using the **InfoPrint Manager Web Management Interface** (WMI). For more information about completing this task, see the WMI help.
2. Verify that the gateway is working. Open your Internet browser and enter this web address, replacing *serverhostname* with the DNS host name of the system that InfoPrint Manager is installed on `http://serverhostname:631/printers`. If the gateway is working correctly, you will see a list of the printers that are available through the gateway. All of your InfoPrint Manager destinations should be listed. Find the address (URI) of the destination you want your users to print to.

   Note

   - If you do not have a DNS name server running on your network, replace *serverhostname* with the dotted-decimal IP address of the system that InfoPrint Manager is installed on instead.
3. Refer your users to http://www.pwg.org/ipp/. This step varies depending on which IPP client you are using; follow the instructions for adding a printer as provided by the IPP client. For the printer address, have your users enter the address that you found in step 2.

Using the Virtual Printer Daemon (VPD) Gateway with InfoPrint Manager for Linux

The InfoPrint Manager Virtual Printer Daemon (VPD) is a utility that uses port 9100 and lets you submit raw print jobs from other operating systems or software using a TCP/IP client. It submits the files that it receives directly to an InfoPrint Manager destination, not to a Linux printer. Only one VPD can run on your InfoPrint Manager server system at a time.

When you submit jobs to the InfoPrint Manager Virtual Printer Daemon (VPD), it uses a mapping file to match the PJL commands within the data stream with InfoPrint Manager attributes. A sample mapping file comes with InfoPrint Manager for Linux; before you submit any print jobs to the InfoPrint Manager Virtual Printer Daemon (VPD), modify this file to meet your needs.

To modify the mapping file:

1. Open the `/var/pd/vpd/vpdmap.txt` file in a text editor and edit the mappings as necessary.
2. Save the file using the same name.

The best way to see what PJL commands you need to add or change is to uncomment the line that reads `* DEBUG` and save the file. (To uncomment the line, delete the *.) Submit a print job through the VPD and check the output. If the job fails or if it prints incorrectly, check the error messages in the server log in the . Use them to help determine which keywords are missing or mapped incorrectly. When you are finished, open the mapping file and comment the DEBUG line out again.

Submitting print jobs

When you submit jobs using a printing client, the VPD identifies a running InfoPrint Manager server and submits that job to it. The VPD uses the name of the target InfoPrint Manager logical or actual destination determined by the PJL mappings or it uses the PDPRINTER environment variable.

Using AFP Upload

AFP Upload consists of a client component PSF Upload TCP/IP DSS (as part of InfoPrint Manager) and an MVS host server component (available as a priced feature of PSF for z/OS).

Note

The z/OS operating system replaced the MVS operating system, but in InfoPrint Manager we continue to refer to the MVS operating system because it is used throughout our program.

The client component is part of the InfoPrint Manager server. The InfoPrint Manager server accepts AFP files (or files it can convert to AFP) and sends them to the MVS server, which places them on the JES spool. The host server program ensures that the attributes passed by the InfoPrint Manager server (such as `class`, `destination`, and `forms`) are recorded with the data sets on the JES spool. For more information on how to pass these values to an MVS system, see Transmitting AFP file attributes to MVS systems, p. 193.

Software prerequisites

The following software products are required to use AFP Upload on TCP/IP networks:

Host System

- z/OS V1R7 or later
- PSF 3.4.0 for z/OS AFP Upload Feature

Linux System

- Supported Linux systems
- Supported InfoPrint Manager versions

General information about the host server program

The host server program is a single load module that can be purchased as a feature of PSF for z/OS. Once started, the host server program runs continuously. It waits for connection requests from one or more InfoPrint Manager servers and creates a subtask to process each request. When a request is completely processed, the subtask ends. The host server program creates a new subtask for each request.

You can run the host server program in a startable address space or as a job managed by a host initiator.

The InfoPrint Manager server sends AFP data to the host server program that places the AFP data in a SYSOUT data set. When the host server program allocates the SYSOUT data set, it accepts the default spool space allocation for the SYSOUT class definition in JES.

The host server program associates attributes passed by the InfoPrint Manager server with the SYSOUT data set. These attributes are similar to JCL OUTPUT statement parameters. They include:

- ADDRESS
- BUILDING
- BURST
- CLASS
- COPIES
- DATACK
- DEPARTMENT
- DESTINATION
- FORMDEF
- FORMS
- NAME
- ROOM
- TITLE
- WRITER

Note

1. The host server program requires a 5A control character to precede each AFP statement in the AFP data sent by the InfoPrint Manager server.
2. The InfoPrint Manager server always passes a value for the DATACK parameter with each uploaded file. This DATACK value overrides the DATACK parameter in the PSF for z/OS PRINTDEV statement.

Specifying the SYSPRINT DD statement

The host server program records its configuration settings and information about errors in the data set referenced by the SYSPRINT DD statement. You must include a SYSPRINT DD statement in the JCL statements that run the host server program.

If the SYSPRINT DD statement refers to a data set other than SYSOUT or DUMMY, the data set must have the following attributes:

directory blocks:

0

record format:

VA

record length:

125

block size:

129

Supported configuration parameters

The Linux client provides most of the information required by the host server program. The host server program has only a few configuration parameters. The default values for these parameters are appropriate in most cases.

CASE

Specifies whether the host server program should convert text OUTPUT parameter values to upper case. The following OUTPUT parameters are affected:

- ADDRESS
- BUILDING
- DEPARTMENT
- NAME
- ROOM
- TITLE

The CASE parameter does not affect some other OUTPUT parameter values, which are always converted to upper case:

- CLASS
- DESTINATION
- FORMDEF
- FORMS
- WRITER

MIXED

Accept the text values as provided by the Linux client.

Note

Some code pages used by fonts printed on header and trailer pages do not include lower case characters.

UPPER

Convert all alphabetic English characters to upper case.

PORTNUM

Specifies the TCP port number at which the host server program accepts connection requests from clients. Valid values are 1 to 65535, but values above 5000 are recommended. The default value is 6001.

TASN

Specifies the name of the z/OS TCP/IP address space. The default value is TCPIP.

TRACE

Specifies whether the host server program should record its internal flow in the data set referenced in the SYSPRINT DD statement.

Note

Using a TRACE value other than NONE degrades performance. You should specify a value other than NONE only if directed to do so by an IBM service representative.

DS

Record transmitted and received data, as well as the internal execution path.

NONE

Do not record trace information. The host server program only writes configuration and error information in the SYSPRINT data set.

NORM

Record the internal execution path.

Specifying the configuration parameters

Specify host server program parameters and values as a program parameter on the JCL EXEC statement.

For example,

```
//STEP    EXEC PGM=AUPTCPS,PARM='/CASE=MIXED'
```

or

```
//STEP    EXEC PGM=AUPTCPS,PARM='/CASE=MIXED,TRACE=NORM'
```

The PARM value has the format:

```
'run-time options/program parameter'
```

Run-time options control aspects of the program's execution environment, such as the amount of storage available to the program.

The program parameter is passed to the program.

Specifying PL/I run-time options

The host server program is written in the PL/I language. Like most PL/I programs, aspects of its execution environment can be controlled by run-time options.

You can specify run-time options in the PARM operand on the JCL EXEC statement.

For example,

```
//STEP    EXEC PGM=AUPTCPS,PARM='SPAE,SPIE/CASE=MIXED'
```

The PARM value has the format:

```
'run-time options/program parameter'
```

The program parameter is passed to the program. The host server program accepts its configuration parameters in this way.

For more information about run-time options for PL/I programs, see *z/OS Language Environment Programming Guide*.

Transmitting AFP file attributes to MVS systems

InfoPrint Manager lets you specify values to MVS through two methods:

1. Using specific InfoPrint Manager attributes, p. 193
2. Using the destination-pass-through attribute, p. 194

The CLASS, FORMS, and DESTINATION values allow you to specify values through either method. If you want to send these three values through AFP Upload and Print, as well as the BURST value, you must use the **destination-pass-through** attribute. If you specify the CLASS, FORMS, and DESTINATION attributes as specific attributes (**mvs-class**, **mvs-destination**, or **mvs-forms**) and then specify a BURST value on the same command or actual destination, InfoPrint Manager ignores the specific document attributes and only passes the BURST value onto the MVS host program. To pass BURST with the other three values, you must use the method described in Using the destination-pass-through attribute, p. 194.

Using specific InfoPrint Manager attributes

You can specify a series of document attributes that map to MVS values by associating them with a job, specifying them on a **pdpr** print command, or setting them on an actual destination.

The following table provides specific attributes that can be used to pass values to an MVS host system through AFP Upload and Print.

Specific InfoPrint Manager attributes to pass values to an MVS host system

MVS Values	InfoPrint Manager Attribute(s)
ADDRESS	• **address1–text=** • **address2–text=** • **address3–text=** • **address4–text=**
BUILDING	**building–text=**
CLASS	• **mvs-class=** • (**destination-pass-through=**)
COPIES	**copy–count=**
DATACK	**data–fidelity-problem-report=**
DESTINATION	• **mvs-destination=** • (**destination-pass-through=**)
DEPARTMENT	**department-text=**
FORMDEF	**form-definition=**
FORMS	• **mvs-forms=** • (**destination-pass-through=**)
NAME	**name-text=**
ROOM	**room-text=**
TITLE	**title-text=**

For example, to pass a BUILDING value of `004` and a ROOM value of `A5`, you could use the **pdpr** print command to pass attributes for a single job with a **test1.tif** file:

```
pdpr -p nome-lp -x"building-text=004 room-text=A5" test1.tif
```

For specific syntax rules, see *RICOH InfoPrint Manager: Reference*, which is available in the RICOH Software Information Center at http://help.ricohsoftware.com/swinfocenter.

Using the destination-pass-through attribute

For a subset of MVS values that do not match specific InfoPrint Manager attributes, InfoPrint Manager lets you specify values to MVS as **destination-pass-through** (also called **printer-pass-through**) attribute values. These values can be associated with a job, specified on a **pdpr** print command, or set on an actual destination. The MVS host server program associates attributes passed by the Linux client with the SYSOUT data set. These attributes are similar to JCL OUTPUT statement parameters.

They include:

- BURST
- CLASS
- DESTINATION
- FORMS

These procedures describe how you can pass these values to an MVS host server program using AFP Upload and Print, using this syntax:

```
-x "destination-pass-through='PASSTHRU=MVS_PARM=value'"
```

where *MVS_PARM* can be BURST, CLASS, DESTINATION, and FORMS, either individually or in any combination. When more than one attribute is being passed, each value must be separated by a comma, as shown in the examples below.

The MVS value WRITER must be specified through the **destination-pass-through** attribute, but it does not conflict with specifying either **mvs-class**, **mvs-destination**, or **mvs-forms**. As you can see in the table below and the examples that follow, the syntax for passing the MVS WRITER value differs from that used for the other options.

This table shows how you can use the **destination-pass-through** attribute to pass values to an MVS host system through AFP Upload and Print.

Using the destination-pass-through attributes to pass values to an MVS host system

MVS Values	InfoPrint Manager Pass Through Syntax
BURST	destination-pass-through='PASSTHRU=BURST=xxx'
CLASS	destination-pass-through='PASSTHRU=CLASS=xxx'
DESTINATION	destination-pass-through='PASSTHRU=DESTINATION=xxx'
FORMS	destination-pass-through='PASSTHRU=FORMS=xxx'
WRITER	destination-pass-through='WRITER=xxx'

Examples: Using the destination-pass-through attribute

The first two-step example creates an initial value document and associates it with the default logical destination so that all jobs sent to this destination will pass the specified attributes.

1. Specify this **pdcreate** command to create an initial value document named **upload1–dd** on the InfoPrint Linux server **nome**:

```
pdcreate -c initial-value-document \
-x"destination-pass-through='WRITER=WTR1 PASSTHRU=BURST=YES,CLASS=A'"\
nome:upload1-dd
```

2. Specify this pdset command to associate the initial value document named **upload1–dd** with the **nome-lp** logical destination:

```
pdset -c destination\
-x"destination-initial-value-document=upload1-dd"\
nome:nome-lp
```

The second example uses the **pdpr** command to pass attributes for a class of `A`, a form of `B`, a job destination of `B205`, and a BURST value of `YES` on a single job with a **test2.tif** input file:

```
pdpr -p nome-lp\
-x"destination-pass-through='PASSTHRU=CLASS=A,FORMS=B,DESTINATION=B205,
```

```
    BURST=YES'"\
test2.tif
```

The third example uses the pdset command to specify a class of A, a form of B, a BURST value of YES, and a WRITER value of WTR1 on all jobs sent to the actual destination **nome-ad**.

```
pdset \
-x"destination-pass-through='WRITER=WTR1 PASSTHRU=CLASS=A,FORMS=B,
    BURST=YES'"\
nome:nome-ad
```

Using InfoPrint Manager commands with AFP Upload

Because the Linux client program for AFP Upload operates as an InfoPrint Manager actual destination (PSF Upload TCP/IP DSS), you can control and monitor it through the commands used with other InfoPrint Manager actual destinations:

- To shut down an PSF Upload TCP/IP actual destination, use the **pdshutdown** command that is documented in *RICOH InfoPrint Manager: Reference*.
- To find status about PSF Upload TCP/IP actual destinations, use the **psfstat** command.

There are limitations to the tasks that you can do with an PSF Upload TCP/IP actual destination. You can neither forward space, back space, nor pause a job to an PSF Upload TCP/IP actual destination.

When the connection to the MVS system host fails

If you experience connection problems with the MVS host after having submitted a job through AFP Upload and Print, how to make sure that the job prints depends upon whether InfoPrint Manager detected the loss of connectivity.

If InfoPrint Manager detected the lost connection, the PSF Upload TCP/IP logical destination becomes red and the job goes into the PENDING state. Once you have re-established connectivity to the MVS host program, re-enable the PSF Upload TCP/IP destination. Then InfoPrint Manager will restart the job and continue printing.

If InfoPrint Manager does not detect the lost connection, the job remains in the PENDING state, while the printer goes into a needs-key-operator state. To resolve this condition and print your job, you must regain connectivity to the MVS host program, cancel the job in the PENDING state, and then resubmit the job to the PSF Upload TCP/IP actual destination.

Troubleshooting problems with AFP Upload on TCP/IP networks

This chapter describes some of the places where information about errors is recorded. It also lists a few common errors and their most likely causes.

Places where error information is recorded

This section lists a few of the places you should look to get information about errors configuring and using AFP Upload on TCP/IP networks:

error.log for an InfoPrint Manager actual destination

The AFP Upload client program operates as an InfoPrint Manager PSF Upload TCP/IP actual destination. Like all InfoPrint Manager actual destinations, it records error messages in a file named **error.log** in the directory for the InfoPrint Manager actual destination. The directory is named

```
/var/psf/printername
```

where `printername` is the name of the InfoPrint Manager actual destination.

You can use the **psfmsg** command to display detailed information about any InfoPrint Manager PSF message. For more information about the **psfmsg** command, see *RICOH InfoPrint Manager: Reference*.

MVS console log

The MVS console log contains information about host server program errors, especially if the host server program runs in a startable address space.

MVS held output

If the host server program runs as a job managed by an MVS initiator, check the job's held output for information about errors. You must use a MSGCLASS on the JCL JOB statement that causes job execution data sets to be held.

SYSPRINT data set

The AFP Upload host server program records its configuration values and information about run-time errors in the data set referenced by the SYSPRINT DD statement. See Specifying the SYSPRINT DD statement, p. 191.

Common errors with AFP Upload using TCP/IP networks

The following contains common errors when using AFP Upload using TCP/IP networks. Each problem displays the **Symptom** and its **Likely Causes and Resolutions**:

Symptom:

You run the host server program as a job managed by an MVS initiator. After running for a period, the host server program ends with system code 522.

Likely Causes and Resolutions:

System code 522 indicates that all of the tasks in a job step were in a wait state for the time specified in the JWT parameter of the SMFPRMxx parmlib member. However, lengthy waits are normal during periods when no AFP Upload Linux clients send files to the host server program.

Specify TIME=NOLIMIT in the JCL EXEC statement for the host server program to bypass job step wait limits.

Symptom:

You enqueue an ASCII file to be uploaded, but InfoPrint Manager for Linux issues the following error message:

```
0423-659, File _________ contains data of a type that cannot be processed by this
PSF/6000 instance. For example, the PSF/6000 instance cannot process ASCII
datatype files.
```

Likely Causes and Resolutions:

You cannot directly enqueue an ASCII file to be uploaded; you must use ACIF to transform the file into a AFP data stream file.

Symptom:

The host server program fails immediately after you start it. The SYSPRINT data set contains the following error:

```
INITAPI ERROR = 10191
```

Likely Causes and Resolutions:

The host server program referenced a TCP/IP address space name that is not correct or the address space name is correct but the address space is not running.

If you do not specify a TCP/IP address space name (TASN) when you start the host server program then `TCPIP` is assumed. If the TCP/IP address space name is not `TCPIP`, the host server program records this INITAPI macro error and ends.

If you specify a TCP/IP address space name (TASN) when you start the host server program but it is incorrect, the host server program also records this INITAPI macro error and ends.

Ask your system programmer for the z/OS TCP/IP address space name. Specify it to the host server program using the TASN configuration parameter. For details about the TASN configuration parameter and how to specify it, see General information about the host server program, p. 190.

Symptom:

InfoPrint Manager for Linux issues the following error message:

```
0423-655 PSF/6000 cannot communicate with the server program
at TCP port number ____ at Internet Protocol (IP) address ____.
```

Likely Causes and Resolutions:

The host server program is not started at the given IP address and TCP port number.

Ensure that the IP address is correct. Start the host server program. Ensure that the **PORT number** configuration parameter for the actual destination equals the PORTNUM value that you specified when you started the host server program. If you do not specify a PORTNUM value when you start the host server program then 6001 is used. For details about specifying the **PORTNUM** configuration parameter, see General information about the host server program, p. 190. For details about creating a PSF TCP/IP printer, see *"Creating and managing actual destinations"* in *RICOH InfoPrint Manager for Linux: Getting Started*.

Specifying JCL Statements to run the host server program in a startable address space

You can run the host server program either in a startable address space or as a job managed by an MVS initiator. This section describes how to run the host server program in a startable address space.

You must place a JCL PROC in a PROCLIB and use the MVS START command to run the PROC. The JCL must execute the AFP Upload server program, named AUPTCPS. This program is written in the PL/I language. It requires that the Language Environment runtime library be included in the STEPLIB definition. The AFP Upload server program also requires that the SYSPRINT DD name be allocated. For more information about the SYSPRINT DD name used by the host server program, see Specifying the SYSPRINT DD statement, p. 191.

You can use the MVS START and CANCEL commands to control execution of the host server program. The host server program does not support or respond to the MVS STOP and DISPLAY commands.

Note

The host server program runs continuously. If you run the host server program in a startable address space, the host server program runs until you cancel the address space.

Specifying JCL Statements to run the host server program as a job

You can run the host server program either in a startable address space or as a job managed by an MVS initiator. This section describes how to run the host server program as a job managed by an MVS initiator.

The JCL must execute the AFP Upload server program, named AUPTCPS. This program is written in the PL/I language. It requires that the Language Environment runtime library be included in the STEPLIB definition. The AFP Upload server program also requires that the SYSPRINT DD name be allocated. For more information about the SYSPRINT DD name used by the host server program, see Specifying the SYSPRINT DD statement, p. 191.

Specify TIME=NOLIMIT in the JCL EXEC statement. This specification causes job step wait limits to be bypassed. If this specification is omitted, the job will end with system code 522 if all of the tasks are in a wait state for the time specified in the JWT parameter of the SMFPRMxx parmlib member. Lengthy waits are normal during periods when no AFP Upload Linux clients send files to the host server program.

Note

The host server program runs continuously. If you run the host server program as a job managed by an MVS initiator, the host server program runs until you cancel the job.

Example JCL Statements to run the host server program in a Startable Address Space

For an explanation of these values, see Specifying JCL Statements to run the host server program in a startable address space, p. 199.

Sample JCL Statements to run the host server program in a Startable Address Space

```
//UPLOAD   PROC
//STEP     EXEC PGM=AUPTCPS
//STEPLIB  DD   DSN=AUP.SAUPLOAD,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD   SYSOUT=*
```

Example JCL Statements to run the host server program as a job

For an explanation of these values, see Specifying JCL Statements to run the host server program as a job, p. 199.

Sample JCL Statements to run the host server program as a job

```
//JOBNAME  JOB  ACCOUNT
//STEP     EXEC PGM=AUPTCPS,TIME=NOLIMIT
//STEPLIB  DD   DSN=AUP.SAUPLOAD,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD   SYSOUT=*
```

Using the SystemD services

You can start, stop, restart, enable, or disable the following InfoPrint Manager services using the SystemD **`systemctl`** command:

- InfoPrint Manager IPPGW (ipm-ippgw)
- InfoPrint Manager Licensing (ipm-licverd)
- InfoPrint Manager LDAP Cache (ipm-lcd)
- InfoPrint Manager LPD (ipm-lpd)
- InfoPrint Manager MVS Download (ipm-mvsd)
- InfoPrint Manager Notification (ipm-notifyd)
- InfoPrint Manager PDServers (ipm-pdservers)
- InfoPrint Manager SAP (ipm-sapcbd)
- InfoPrint Manager VPD (ipm-vpd)
- InfoPrint Manager Web Server (ipm-ws)

To start a stopped InfoPrint Manager service, you can use:

```
systemctl start <servicename>[.service]
```

To stop an InfoPrint Manager service, you can use:

```
systemctl stop <servicename>[.service]
```

To restart an InfoPrint Manager service, you can use:

```
systemctl restart <servicename>[.service]
```

To enable an InfoPrint Manager service, you can use:

```
systemctl enable <servicename>[.service]
```

To disable an InfoPrint Manager service, you can use:

```
systemctl disable <servicename>[.service]
```

Note

- Only root and the user running InfoPrint Manager can start, stop, or restart an InfoPrint Manager service by using **`systemctl`**.
- 2. Only root can enable or disable an InfoPrint Manager service by using **`systemctl`**.

For more information, see the man page for `systemctl (1)`.

Important

- On Red Hat Enterprise Linux 10, to enable the $IPM_USER user to restart the InfoPrint Manager services from both the command line and the Web Management Interface, the startup behavior of the `polkit` service has been modified through an `override.conf` file.

2. Administrative Procedures: Configuring for Host Printing

• **Printing data sets from an MVS system through MVS Download Receiver**

Printing data sets from an MVS system through MVS Download Receiver

This section contains information about setting up and using the MVS Download Receiver daemon supplied with InfoPrint Manager to print MVS jobs on InfoPrint-managed printers. The MVS jobs must come from an MVS system that has the Download for z/OS or AFP Download Plus feature of IBM Print Services Facility for z/OS program product installed.

The InfoPrint MVS Download Receiver daemon allows MVS users to print data sets from the MVS JES spool on InfoPrint-managed printer devices. The MVS data sets contain MO:DCA data. With Download for z/OS, you can also print MVS data sets containing PostScript data or data in any of the other formats supported by InfoPrint.

The PSF for z/OS Download feature uses a routing-control data set residing on the MVS system to map any combination of the **CLASS**, **FORMS**, and **DEST** JCL parameters to a specific MVS Download Receiver daemon on a specific InfoPrint Manager server.

See the *PSF for z/OS: Download for z/OS Guide* for information about the Download for z/OS feature and tasks done on the MVS system.

The PSF for z/OS AFP Download Plus feature uses parameters in the PRINTDEV to identify a single MVS Download Receiver daemon on a specific InfoPrint Manager server. See the *PSF for z/OS: AFP Download Plus Guide* for information about the AFP Download Plus feature and tasks done on the MVS system.

Preparing to work with Download for z/OS

Read through this to become familiar with MVS Download terms and components.

Why use Manager MVS Download Receiver?

When you have data on an MVS system that you want to print on an InfoPrint-managed printer device, you can use the Download for z/OS or AFP Download Plus feature of PSF for z/OS and the InfoPrint Manager MVS Download Receiver daemon. The PSF for z/OS features and the InfoPrint daemon:

- Provide a high-speed data transfer utility for high-volume jobs.
- Restart from a checkpoint if something interrupts the data transmission.
- Do not require separate spool management and queueing on MVS system because of seamless integration with System Display and Search Facility (SDSF).

For a comparison between the Download for z/OS and AFP Download Plus features, see the *Download for z/OS Guide*.

What is the MVS Download Receiver daemon?

InfoPrint Manager provides a daemon, **mvsprsd**, which receives data sets from one or more MVS systems. If you expect a high volume of data, you can configure several **mvsprsd** daemons to receive the data sets from MVS.

To configure an **mvsprsd** daemon, you must do these tasks:

1. Make the Advanced Function Presentation (AFP) resources on the MVS system available to InfoPrint if they are not already available.
2. Use the default version or a customized version of the shell script **mvsprpsm.sh** supplied with InfoPrint Manager. See Modifying the InfoPrint mvsprpsm.sh shell script, p. 206 for more information.
3. Assign a port number to the InfoPrint Manager MVS Download Receiver daemon and start the daemon using the **mvsprsd** command.
   See Using the mvsprsd command to start the daemon, p. 214.

How do you make AFP resources available?

AFP Download Plus can send all needed AFP resources with the print files. When this is used, the AFP resources do not need to be accessible to the Linux system. If AFP Download Plus is configured to not send all needed AFP resources or Download for z/OS is in use, the AFP resources needed for the jobs need to be accessible to the Linux system.

For jobs that require AFP resources located on the MVS system, you can use the Network File System (NFS) **mount** command to make MVS file systems containing resources accessible to the Linux system on which InfoPrint Manager is running. Or you can have the resources located on the Linux system. You use the shell script **mvsprsd.sh**, supplied with InfoPrint Manager in the **/usr/lpp/pd/bin** directory, to specify the location of the resources.

You can associate named media with the bins where they are loaded. By using Page Printer Formatting Aid (PPFA), you can create form definitions for your MVS jobs that contain MO:DCA medium maps that specify media names. For more information about how to create these form definitions, see the *Page Printer Formatting Aid: User's Guide and Reference*.

How are print options passed to InfoPrint?

MVS users submitting MVS data sets for printing on InfoPrint-managed printer devices can pass a number of options that further define the requirements of their jobs. The InfoPrint daemon receives the options with the data set and passes the options to the shell script.

See Passing the parameter list from the mvsprsd daemon to the shell script, p. 209 for information about the options.

What does the mvsprpsm.sh shell script do?

The InfoPrint Manager MVS Download Receiver shell script translates the print options specified with a downloaded MVS file into InfoPrint attributes and values. The shell script then builds an InfoPrint **pdpr**

command using the **-x** flag to submit the attributes and values; it builds the name of the file to print from information received from the MVS system and the date and time the MVS file was received. The shell script then creates a log file and submits the **pdpr** command. A check is made to determine if InfoPrint Manager accepts the command. If InfoPrint accepts the command, the InfoPrint Manager MVS Download Receiver shell script deletes the log file and the data set received from the MVS system. Otherwise, the log and file remain in the defined file system.

What does the log file generated by the mvsprpsm.sh shell script contain?

The filename of the log file has this format:

```
jobname.dataset_name.forms_name.yyddd.log
```

The log file generated by the InfoPrint Manager MVS Download Receiver shell script contains this information:

- The date InfoPrint received the MVS data set and print options
- The print options sent to the shell script by the **mvsprsd** daemon
- A copy of the **pdpr** command issued by the shell script to print the file
- Any error messages issued in response to the **pdpr** command
- The return code from the **pdpr** command

Note

If InfoPrint Manager accepts the command, the InfoPrint Manager MVS Download Receiver shell script deletes the log file and the data set received from the MVS system. If you are having problems, you might want to comment out this behavior while debugging.

If the **pdpr** command fails, you can use the log file located in the file system you defined with the **-d** flag of the `mvsprsd` utility.

- Ensure the shell script received the correct parameters
- Perform InfoPrint Manager troubleshooting procedures, such as using the **pdmsg** command to determine the cause of the problem with the **pdpr** command.

How do I get page count displayed for AFP Download Plus jobs?

If you want to be able to see the page count for your AFP Download Plus jobs listed in the InfoPrint Manager GUI, turn on the page-accounting-support in AFP Download Plus. See the *PSF for z/OS: AFP Download Plus Guide* for more information.

Performance considerations with the Direct Download method

The performance of the download receiver/daemon on the InfoPrint Manager server decreases when using the **Direct download** function. The **Direct download** support can be enabled or disabled from the host (see **AFP Download Plus** guide for further reference).

Note

When enabled, there is an impact upon the throughput of the file from the beginning of file reception across TCP/IP until first page is printed. Also, the CPU and disk usage may increase during the operation.

The memory usage is not expected to be affected.

Using **Direct download** causes additional temporary files in the download working directory. You must ensure that you have sufficient space in the file system where the download working directories reside.

If the download working directories are in the same file system as the **InfoPrint Manager** spool (`/var/pd`), no additional file system space is required.

If the download working directories are in a different file system as the **InfoPrint Manager** spool, ensure that you have sufficient space for twice the amount of data in the working directories.

Modifying the InfoPrint mvsprpsm.sh shell script

You can use the default version of the **mvsprpsm.sh** shell script, located in the **/usr/lpp/pd/bin** directory, or modify the shell script to meet the needs of your production printing environment. For example, you might include the path to AFP resources on the Linux system or change how the shell script generates the file name for the data set received from the MVS system. Or you might want to specify a default page definition.

When modifying the **mvsprpsm.sh script**, note that parameters you comment out do not get dropped but get sent passthrough to the actual destination. For example, if you comment out the value for copies defined in the script:

```
#   cop   )   outputstr="$outputstr results-profile=::${optparm#=}" ;;
```

the script passes the original JCL copy-count value to the printer. Instead of commenting out the line in the shell script, change the name of the parameter to something that will be read as a value, but not passed onto the InfoPrint printer, such as **junk**:

```
cop   )   junk="$outputstr results-profile=::${optparm#=}" ;;
```

Note

- Copy the shell script to another directory such as **/var/pd** and modify the copy. This eliminates the possibility of service overwriting your modifications.
- Only a person with `root` user authority or the InfoPrint Manager user can run this command.

The **mvsprpsm.sh** shell script and associated scripts have been changed to add support for AFP Download Plus and the multiple data set function of AFP Download Plus. If you already have custom scripts that will be used instead of the InfoPrint Manager provided scripts, make sure that they have been updated with the changes in the new scripts before using AFP Download Plus.

Changes to custom scripts/exits are only required if any of these new functions are desired:

- AFP Download Plus multiple data set
- Support for the **pdpr** retry count and retry interval
- Displaying the AFP Download Plus page count on the Admin/Operator GUI

- Printing/discarding AFP Download Plus failed jobs
- Using inline resources with Download for z/OS multiple data set
- Support for separator pages with AFP Download Plus multiple data set
- Using the new **-w** or **-e** option.

If none of these functions are used, the existing custom scripts can be used.

See the current sample script/exit provided and the sample script/exit that your custom script/exit is based on to determine the changes. From this, you will identify the proper approach for updating and testing your custom script/exit.

Specifying AFP resource directories

The **mvsprpsm.sh** shell script contains these lines, which are commented out by default:

```
# NOTE: Add resource-context-xxx search paths to the pdpr options
# if necessary.  For example:
#outputstr="$outputstr resource-context-font=/font/reslib"
#outputstr="$outputstr
#   resource-contect-page-def=/dept123/pagedef:/acme/pagedef"
```

Use the format shown in the third, fourth and fifth lines to specify the paths to the directories containing AFP resources. Do not include the pound sign, **#**, when specifying the directories. You can specify multiple directories by separating them with colons. You can also specify MVS file systems accessed through the NFS **mount** command.

Changing how the shell script generates the file name

The **mvsprpsm.sh** shell script contains the following lines, which cause the MVS Download Receiver shell script to include the MVS job identifier in the name of the file generated by the shell script:

```
# Rename the downloaded file to include the MVS job id (comment out
# if this behavior is not desired)
if [ "$paopt" != "" ; then
   JID=${paopt##*jobid=}
   if [ "$JID" != "$paopt" ] ; then
      JID=${JID%%, *}
      if [ "$JID" != "" ]; then
         mv $filename $JID.$filename
         filename=$JID.$filename
         mv $logfile $JID.$logfile
         logfile=$JID.$logfile
      fi
   fi
fi
```

If you do not want the MVS job identifier included, comment out the lines.

Using multiple printers with MVS Download Receiver

By default, an MVS Download Receiver process submits all jobs to the same logical printer. A parameter that specifies this logical printer starts the **mvsprsd** daemon. This printer is fixed for the

duration of the process. If you want to send MVS jobs to multiple InfoPrint Manager printers, then multiple daemons must be started for each printer.

If you want a single **mvsprsd** daemon process to route MVS jobs to different printers, you can uncomment these lines from the **mvsprpsm.sh** shell script:

```
## ##########################################################################
## The following code-segment should be uncommented if dynamic printer
##        assignment is desired.  Uncomment lines that are singly (#)
##        commented. Lines with double comment (##) should remain
##        commented.
##
## The MVS parameter DEST=xxxx is used to assign the logical printer
##        (thereby making the MVS Download daemon's startup
##         queue irrelevant).
##
## The assumption below is that the DEST passed is the actual name
##        of an IPM logical destination.  Note that both JES2 and JES3
##        will pass the DEST in the NJE header in UPPER CASE but the
##        code below will convert it to lower-case.
## ##########################################################################
##########################################################
## Begin dynamic printer assignment from DEST
##########################################################
#  MVSdest=${paopt##*destination=}
#  if [ "$MVSdest" != "$paopt" ] ; then
#     MVSdest=${MVSdest%%,*}
#     MVSdest=${MVSdest% }
#     if [ "$MVSdest" != "" ] ; then
#        outputstr="$outputstr mvs-destination=$MVSdest"
#        paopt=${paopt##destination=$MVSdest}
         ##############################################################
         ## The following typeset converts to lower-case (-l flag)
         ##############################################################
#        typeset -l lpname=$MVSdest
#     fi
#  fi
##########################################################
## End dynamic printer assignment from DEST
##########################################################
```

Once this code has been uncommented, the MVS operators merely need to specify the target InfoPrint Manager logical destination through the **DEST=** parameter on each job that is submitted through MVS Download Receiver.

If you will be sending jobs that do not specify a value for the **DEST=** parameter in the JCL, the MVS configuration substitutes a default value, which is typically `LOCAL`. If your InfoPrint configuration does not have a destination with the same name, those jobs fail and InfoPrint issues this message:

```
5010-056 Cannot find object 'local' rc=1
```

To make sure that these jobs do not fail, define either an InfoPrint Manager logical destination or an actual destination named **local** (or whatever serves as a lowercase equivalent for the default value on your MVS configuration) that addresses the appropriate printer.

Specifying a default page definition

All line-data jobs downloaded from MVS systems require a page definition before you can print them using InfoPrint. If no page definition is specified, an error occurs. By default, the shell script contains this line:

```
pagedef     )    outputstr="$outputstr page-definition$optparm" ;;
```

This causes the shell script to set the value for the **page-definition** attribute based on a value passed to the shell script by the **mvsprsd** daemon. The value passed by the daemon depends on options downloaded from MVS with the data set. You can add conditional processing to the shell script to specify a page definition if no page definition specification is downloaded with the file.

Passing the parameter list from the mvsprsd daemon to the shell script

The InfoPrint **mvsprsd** daemon passes a parameter list for each data set it receives to the InfoPrint Manager MVS Download Receiver shell script. The parameter list contains this information:

1. The name of the downloaded data set in the format:
   - For Download for z/OS:
     `MVS_system_name.job_name.dataset_name.forms_name.yyddd.hhmmsstABCD.PRD`
     where:

     MVS_system_name

     Specifies the name of the MVS system that downloaded the data set.

     job_name

     Specifies the name of the job containing the data set.

     dataset_name

     Specifies the name of the data set.

     forms_name

     Specifies the name of the form identified with the MVS **FORMS** parameter.

     yyddd

     Specifies the year followed by the number of the day of the year.

     hhmmsstABCD

     Specifies the time the data set was received in hours, minutes, seconds, tenths of a second, hundreds of a second, thousands of a second, and hundred thousands of a second.

     PRD

     Specifies the type of file that contains print data. PRD is the suffix used by PSF for AIX, InfoPrint Manager for AIX, InfoPrint Manager for Linux or InfoPrint Manager for Windows servers to identify the data file received and other information needed to create the final print job.
   - For AFP Download Plus:

```
AFPDP.MVS_system_name.fsa_name.userid.jobid.job_name.step_name.dd_name.
```

```
  forms_name.ccyyddd.hhmmsstABCD.PRD
```

where:

MVS_system_name

Specifies the name of the MVS system that downloaded the data.

fsa_name

Specifies the name of the FSA that processed the data set.

userid

Specifies the userid of the user who submitted the job.

jobid

Specifies the job identifier that was assigned by the z/OS system.

job_name

Specifies the name of the job containing the data set.

step_name

Specifies the name of the step in the job.

dd_name

Specifies the **DD** name in the step.

forms_name

Specifies the name of the form identified with the MVS **FORMS** parameter.

ccyyddd

Specifies the century and year followed by the number of the day of the year.

hhmmsstABCD

Specifies the time the data set was received in hours, minutes, seconds, tenths of a second, hundreds of a second, thousands of a second, and hundred thousands of a second.

PRD

Specifies the type of file that contains print data. PRD is the suffix used by PSF for AIX, InfoPrint Manager for AIX, InfoPrint Manager for Linux or InfoPrint Manager for Windows servers to identify the data file received and other information needed to create the final print job.

2. The print-options string, which is a series of options and values. See Print options passed to InfoPrint Manager MVS Download Receiver shell script, p. 211 for the list of options. When you specify the **-k** flag with the **mvsprsd** command, InfoPrint also stores the options in a file whose name is in this format:

   - For Download for z/OS:

```
MVS_system_name.job_name.dataset_name.forms_name.yyddd.hhmmsstABCD.JCL
```

   - For AFP Download Plus:

```
AFPDP.MVS_system_name.fsa_name.userid.jobid.job_name.step_name.dd_name.
  forms_name.ccyyddd.hhmmsstABCD.JCL
```

where all the qualifiers in the name are the same as those for the name of the data set, except for **JCL**. The **-k** flag specifies that the file contains print options.

3. The logical printer name, which was specified with the **-q** flag of the InfoPrint **mvsprsd** command when the daemon was started.
4. An empty string.
5. An empty string.
6. The port number that was specified with the **-p** flag of the InfoPrint **mvsprsd** command when the daemon was started.
7. The additional script parameters that were specified with the **-X** flag of the InfoPrint **mvsprsd** command when the daemon was started.
8. A non-zero value, if the **-t** flag was specified on the **mvsprsd** command; or a value of zero, if the **-t** flag was not specified.

Translating print options to InfoPrint attributes and values

The next table shows the print options and values passed to the shell script in the print-options string, the corresponding InfoPrint attribute, and the attribute value used for the **pdpr** command for that job.

- MVS sends print option values in either uppercase or lowercase characters. MVS sends all print option values shown in italics in the next table, such as *pagedef*, *formdef*, and *font1*, in uppercase. It sends other values as shown in the table.
- InfoPrint sets most attribute values to exactly the same values received from the print-option values; the next table lists these values in italics in the **InfoPrint Attribute and Value** columns.

Print options passed to InfoPrint Manager MVS Download Receiver shell script

Print Option Passed to Shell Script	*InfoPrint Attribute and Value for Download for z/OS*	*InfoPrint Attribute and Value for AFP Download Plus*
-oac=account	"account-text=account"	"account-text=account"
-oaddress1=address1	"address1-text=address1"	"address1-text=address1"
-oaddress2=address2	"address2-text=address2"	"address2-text=address2"
-oaddress3=address3	"address3-text=address3"	"address3-text=address3"
-oaddress4=address4	"address4-text=address4"	"address4-text=address4"
-obu=building	"building-text=building"	"building-text=building"
-occtype=a -occtype=m	"carriage-control-type=ansi-ebcdic" "carriage-control-type=machine"	None (discarded)

Print Option Passed to Shell Script	*InfoPrint Attribute and Value for Download for z/OS*	*InfoPrint Attribute and Value for AFP Download Plus*
-occtype=z	"carriage-control-type=ansi-ascii" Note The shell script checks if the job uses carriage control characters. If they are used, then InfoPrint sets this attribute and value based on the input parameter.	
-ochars=font1,font2,font3,font4	"chars=font1,font2,font3,font4"	None (discarded)
-ocop=nnn	"results-profile=::nnn"	"results-profile=::nnn"
-odatat=af	Not applicable	"document-format=modca-p"
-odatat=line	"document-format=line-data" Note If the input value is not line, this value is not set and InfoPrint tries to identify the correct format. If InfoPrint cannot identify the format, InfoPrint uses a format of **ascii**.	Not applicable
-odatac=blkchar	"data-fidelity-problem-reported=position"	"data-fidelity-problem-reported=position"
-odatac=blkpos	"data-fidelity-problem-reported=character"	"data-fidelity-problem-reported=character"
-odatac=block	"data-fidelity-problem-reported=none"	"data-fidelity-problem-reported=none"
-odatac=unblock	"data-fidelity-problem-reported=all"	"data-fidelity-problem-reported=all"
-ode=department	"department-text=department"	"department-text=department"
-odu=no	"sides=1 plex=simplex"	"sides=1 plex=simplex"
-odu=normal	"sides=2 plex=simplex"	"sides=2 plex=simplex"
-odu=tumble	"sides=2 plex=tumble"	"sides=2 plex=tumble"
-of=F1formdef	"form-definition=F1formdef"	"form-definition=F1formdef"
-ofileformat=record	"new-line-option=counted-4-octet-aligned"	None (discarded)

Print Option Passed to Shell Script	*InfoPrint Attribute and Value for Download for z/OS*	*InfoPrint Attribute and Value for AFP Download Plus*
-ofileformat=stream	"new-line-option=lf"	
-ofiletype=dshdr -ofiletype=jobhdr -ofiletype=jobtrl -ofiletype=message -ofiletype=userdata	None (discarded)	"auxiliary-sheet-selection=none" "auxiliary-sheet-selection=none" "auxiliary-sheet-selection=none"
-oformlength=nnn.nnn	"form-length=nnn.nnn"	"form-length=nnn.nnn"
-oin=nnn	"printer-pass-through='-obinnnn'"	"printer-pass-through='-obinnnn'"
-ojobn=job_name	"job-name=job_name"	"job-name=job_name"
-ona=name	"name-text=name"	"name-text=name"
-ono=nodeid	"node-id-text=nodeid"	"node-id-text=nodeid"
-ooffxb=nnn.nnn	"x-image-shift-back=nnn.nnn"	"x-image-shift-back=nnn.nnn"
-ooffxf=nnn.nnn	"x-image-shift=nnn.nnn"	"x-image-shift=nnn.nnn"
-ooffyb=nnn.nnn	"y-image-shift-back=nnn.nnn"	"y-image-shift-back=nnn.nnn"
-ooffyf=nnn.nnn	"y-image-shift=nnn.nnn"	"y-image-shift=nnn.nnn"
-oovlyb=overlay	"overlay-back=overlay"	"overlay-back=overlay"
-oovlyf=overlay	"overlay-front=overlay"	"overlay-front=overlay"
-opa= class=output_class destination=destination forms=form_name segmentid=nnn	"printer-pass-through= '-oclass=output_class' '-odestination=destination' '-oforms=form_name' '-osegmentid=nnn'"	"printer-pass-through= '-oclass=output_class' '-odestination=destination' '-oforms=form_name' '-osegmentid=nnn'"
-opagecount=nnnnnnnnn	"job-page-count=nnnnnnnnn"	"job-page-count=nnnnnnnnn"
-opagedef=P1pagedef	"page-definition=P1pagedef"	None (discarded)

Print Option Passed to Shell Script	InfoPrint Attribute and Value for Download for z/OS	InfoPrint Attribute and Value for AFP Download Plus
-opr=programmer_name	"programmer-text=programmer_name"	"programmer-text=programmer_name"
-oprmode=SOSI1 -oprmode=SOSI2	"shift-out-shift-in=one" "shift-out-shift-in=two"	None (discarded)
-ore=P240 -ore=P300	"font-resolution=240" "font-resolution=300"	"font-resolution=240" "font-resolution=300"
-oro=room	"room-text=room"	"room-text=room"
-osheetcount=nnnnnnnnn	None (discarded)	None (discarded)
-oti=title	"title-text=title"	"title-text=title"
-otrc=no -otrc=yes	"table-reference-characters=false" "table-reference-characters=true"	None (discarded)
-ous=userid	"user-id-text=userid"	"user-id-text=userid"

Using the mvsprsd command to start the daemon

To start the **mvsprsd** daemon whenever the InfoPrint Manager system starts, you can include the **mvsprsd** command in the **/etc/rc.mvsd.daemons** file. This file is executed when the Linux operating system is initialized.

Use this procedure to start the **mvsprsd** daemon when the system starts:

1. Use a text editor to open the `/etc/rc.mvsd.daemons` file and find these lines:

```
# List of MVS Download daemons to be started by rc.mvsd
#
# The syntax of mvsprsd follows:
#
# mvsprsd  [-p port] [-d file_system] [-n 1] [-x shell_script]\
          [-X additional_exit_parameters] [-q destination]
#         [-k] [-B] [-H host_code_page] [-L local_code_page] [-t] [-w] [-e]
#
# This file may be edited directly, but the automatic starting and configuring
# of MVS Download daemons is best controlled through
# the InfoPrint Manager Management Interface. Consult documentation
# for more information.
#
```

2. Add lines similar to these to this file:

```
# This is a comment to document that the following command
# starts the mvsprsd daemon automatically at system startup
/usr/lpp/psf/bin/mvsprsd -p 5400 -d /files1 -d /files2 \
-x /usr/lpp/pd/bin/mvsprpsm.sh -q serv2-lp
```

Note that this example uses the default shell script (**mvsprpsm.sh**) and specifies the **serv2–lp** InfoPrint Manager server.

3. Save and close the file.

If the daemon stops for some reason or you want to restart it using different start-up options, you can use the InfoPrint **mvsprsd** command to start the InfoPrint **mvsprsd** daemon. With the command, you specify a file system in which to store downloaded data sets, a shell script name, a logical printer name, and whether print-option information is stored in a file. You start the daemon on the server that has InfoPrint Manager installed. Once the **mvsprsd** daemon is operational, you can use Download for z/OS to submit a job toInfoPrint Manager where the InfoPrint Manager MVS Download Receiver shell script translates the JCL keywords and values to InfoPrint Manager attributes and values and issues a **pdpr** command.

For the syntax and flags of the **mvsprsd** command, and for examples of its use, see *RICOH InfoPrint Manager: Reference* or the **mvsprsd** man page.

Note

Only a person with `root` user authority or the InfoPrint Manager user can run this command.

Using Management Interface to create an MVS Download Receiver for a default configuration

Before you can begin sending jobs from the MVS JES spool to InfoPrint Manager through the MVS Download feature, follow this procedure to set up an MVS Download Receiver:

1. Open the **InfoPrint Manager Management Interface**.
2. Left-click the **MVS Download Receiver** item in the left pane to select it.
3. Right-click the **MVS Download Receiver** item in the left pane to open a pop-up menu.
4. In the menu that pops up, select **New**. The **Add MVS Receiver** dialog opens.
5. Fill in the fields according to these specifications. For more information about the dialog and the fields it contains, see the Management Console online help.
    - **Port Number**: Enter the port number that you want this receiver to use when it communicates with the host system. This number must match the port number that is specified in the Routing Control File used by the MVS Download FSA for this computer's IP address or the **PORTNO** in the **PRINTDEV** for the AFP Download Plus FSA.
    - **Target Destination**: Select the default InfoPrint destination that you want this receiver to submit jobs to from the drop-down list. InfoPrint Manager will only submit a job to this destination if the jobs does not have a **DEST** specified when it comes from the host system.
    - In the **Processing Options** group box, both options are selected by default and can be left that way.

Note

When the **Retain** command files check box is selected, InfoPrint Manager saves the control files for any MVS Download print jobs that failed in print submission. Therefore, when you want to resubmit those jobs, you do not have to download them from the host system again; the files are still on your InfoPrint server. You can deselect this box, but any jobs that fails in processing will need to be resubmitted from the host to be printed.

Operators should evaluate any processing failures, correct the problems that caused the failures, and resubmit the jobs either from the host system (if this box is deselected) or from the receiver (if this box is selected).

For more information, see Resubmitting or deleting MVS Download Receiver orphaned files, p. 220

6. In the **Error Processing Options** group box, neither option is selected by default and can be left that way.
7. In the **Mapping Options** group box, none of the options are selected by default and can be left that way.
8. Click **OK**.

   One or more pop-up informational messages might appear; read them and click **OK** to dismiss them.
9. Select the MVS Download Receivers item in the left pane of the Management Console. You should see the receiver you just created in the right pane. The receiver status might be **STOPPED**, but it should start momentarily. Click **Refresh View** on the toolbar until the status changes to **RUNNING**. Once the receiver is running, it is ready to receive print jobs from the host system.

Using Download for z/OS support for multiple data set jobs

Download for z/OS sends each JES SYSOUT data set separately to the **mvsprsd** daemon on Linux. By default, the **mvsprsd** daemon spools each data set as a separate file. On MVS, a job can contain multiple data sets. By using the following procedure to customize your installation for MVS Download Receiver multiple data set job support, you might be able to cause the data sets in a job sent to InfoPrint Manager through MVS Download Receiver to print in sequence, with no intervening files, no NPRO processing between files, and only one header and trailer page for the set.

InfoPrint Manager now enables multiple data sets to be received, individually transformed by ACIF into MO:DCA-P, and then combined into a single file and spooled. InfoPrint Manager uses the medium maps (copy groups) from the form definition that is associated with each data set to build a comprehensive inline form definition that it inserts at the beginning of the combined file.

The following describes what you must do to install and configure the MVS Download Receiver support for printing multiple data set set jobs received from PSF/MVS on InfoPrint Manager for Linux.

Description of multiple data set support on both PSF/MVS and InfoPrint Manager for Linux

To support the printing of multiple data sets in a job as a single unit with Download for z/OS, you must:

- Install a special Download for z/OS user exit 15 program (**apsux15m**) on MVS
- Start the **mvsprsd** daemon with the *-n 1* option on Linux

To support the printing of multiple data sets in a job as a single unit with AFP Download Plus, you must:

- Install AFP Download Plus with APAR OA15317.
- Set the **dataset-grouping** parameter in the control statement data set to Yes.
- Start the **mvsprsd** daemon with the *-n 1* option on Linux

The sample user exit 15 program (**apsux15m**) provided with PSF for z/OS for Download for z/OS, adds a parameter that identifies data sets that belong to multiple data set jobs and indicates whether the data set is first, next, or last in a job.

The *-n 1* option ensures that the **mvsprsd** daemon receives the data sets in the correct sequence by creating no more than one child process. If more than one child process were created, Download for z/OS might concurrently send more than one data set from a multiple data set job. If the second data set was shorter than the first, it might be received before the first, causing it to print out of sequence or causing the first data set to be left unprinted.

Note

If you enable multiple data set support using Download for z/OS and you need to configure multiple MVS Download FSA's, you must have one **mvsprsd** daemon for each Download FSA that is submitting jobs to the InfoPrint Manager server. If you do not do this and two Download for z/OS FSA systems send jobs at the same time, the job data sets can become intermingled and errors or incorrect output can result. This restriction does not apply to AFP Download Plus. Multiple AFP Download Plus FSAs can be enabled for multiple data set and use one **mvsprsd** daemon.

Procedure for installing multiple data set support with Download for z/OS

Use this procedure to install the Download for z/OS multiple data set support:

1. Install the Download for z/OS user exit 15 program, **apsux15m**, on MVS. The program must be assembled and linked into a data set referenced in the **STEPLIB** of your Download for z/OS startup **PROC**.
2. Run Download for z/OS on MVS using `SEND_REC_LENGTH=YES` either by specifying it explicitly in the Download for z/OS Routing Control data set on MVS or by allowing it to default. If you specify `SEND_REC_LENGTH=NO`, ACIF might be unable to process some variable length line mode data. In these cases ACIF issues a message that it cannot read the input file with a return code of 310.
3. Because the **mvsprpsm.sh** sample shell script has been updated, you must update any customized versions of this shell script. Specify the name of the shell script using the "-x" option when you start the **mvsprsd** daemon:

```
mvsprsd -n 1 -qlp -p6250 -d/mvsdl -x/mvsdl/mvsprpsm.sh
```

4. Make sure that the **PATH** environment variable when the **mvsprsd** daemon is started includes the **/usr/lpp/psf/bin** directory. You can set default values for environment variables like **PATH** using the **/etc/profile.d/ipm_environment.sh** file:

```
PATH=/usr/lpp/psf/bin:${PATH}
```

You must delimit each directory using a colon (:), not a semi-colon (;). Do not include spaces in the assignment. A period designates the current directory.

5. Make sure that ACIF can locate your resources. It looks in both the **/usr/lpp/psf/reslib** directory and the directories specified by the **PSFPATH** environment variable. If you specify resource directories using these attributes, you must specify those same directories in the **PSFPATH** environment variable:

   - **resource-context**
   - **resource-context-font**
   - **resource-context-overlay**
   - **resource-context-page-segment**
   - **resource-context-page-definition**
   - **resource-context-form-definition**

   You can set default values for environment variables like **PSFPATH** using the **/etc/environment** file:

```
PSFPATH=/usr/xyz_corp/reslib:/usr/xyz_corp/fontlib
```

   You must delimit each directory using a colon (:), not a semi-colon (;). Do not include spaces in the assignment.

6. Do not specify these attributes either directly or in Initial Value Job or Initial Value Document objects:

   - **transform-message-file-name**
   - **resource-context-page-definition**

   You can use the **resource-context** attribute instead of the **resource-context-page-definition** attribute.

7. Specify the *-n 1* option when you start the **mvsprsd** daemon:

```
mvsprsd -n 1 -qlp -p6250 -d/mvsdl -x/mvsdl/mvsprpsm.sh
```

   This option directs the daemon to create exactly one child process to receive data sets from MVS. Most customers use a shell script to run the **mvsprsd** daemon and specify its arguments. Using a shell script ensures that the daemon is always started correctly, with the same arguments. You should add the *-n 1* option to the shell script.

Procedure for installing multiple data set support with AFP Download Plus

Use this procedure to install the AFP Download Plus multiple data set set support:

1. Install AFP Download Plus with APAR OA15317. To enable the multiple data set function, set the **dataset-grouping** parameter in the control statement data set to **Yes**. For more information about installing AFP Download Plus, see the "Installing AFP Download Plus" chapter in *Print Service Facility for z/OS: AFP Download Plus*, S550-0433. Using the Download for z/OS exit 15 program to enable multiple data set on AFP Download Plus is not supported.
2. Because the **mvsprpsm.sh** sample shell script has been updated, you must update any customized versions of this shell script. Specify the name of the shell script using the "-x" option when you start the **mvsprsd** daemon:

```
mvsprsd -n 1 -qlp -p6250 -d/mvsdl -x/mvsdl/mvsprpsm.sh
```

3. When the **mvsprsd** daemon is started, make sure that the **PATH** environment variable includes the **/usr/lpp/psf/bin** directory. You can set default values for environment variables like **PATH** using the **/etc/profile.d/ipm_environment.sh** file:

```
PATH=/usr/lpp/psf/bin:${PATH}
```

   You must delimit each directory using a colon (:), not a semi-colon (;). Do not include spaces in the assignment. A period designates the current directory.

4. Do not specify these attributes either directly or in Initial Value Job or Initial Value Document objects:

   - **transform-message-file-name**
   - **resource-context-page-definition**

   You can use the **resource-context** attribute instead of the **resource-context-page-definition** attribute.

5. Specify the *-n 1* option when you start the **mvsprsd** daemon:

```
mvsprsd -n 1 -qlp -p6250 -d/mvsdl -x/mvsdl/mvsprpsm.sh
```

   This option directs the daemon to create exactly one child process to receive data sets from MVS. Most customers use a shell script to run the **mvsprsd** daemon and specify its arguments. Using a shell script ensures that the daemon is always started correctly, with the same arguments. You should add the *-n 1* option to the shell script.

Limitations of multiple data set support

This support has these limitations:

1. This solution applies only to line mode, mixed mode, or MO:DCA-P data sets.
2. You must specify the *-n 1* option when you start the **mvsprsd** daemon on Linux. This support restricts the **mvsprsd** daemon to receiving one data set at a time. If you did not previously specify the *-n 1* argument and received multiple files concurrently, you might wish to start more than one instance of the **mvsprsd** daemon (using different port numbers) and configure new MVS printers on MVS to communicate with them.
3. You must have enough disk space on the Linux system to receive and transform all of the data sets in a multiple data set job. If you do not use this solution, data sets can be received and transformed individually, so less space might be required.
4. This solution does not create three types of form definition (**FORMDEF**) statements in the generation of a comprehensive inline form definition:

   - MSU (Map Suppression)
   - PFC (Presentation Fidelity Control)
   - MFC (Medium Finishing Control)
   - PEC (Presentation Environment Control)
   - MDR (Map Data Resource)

5. You cannot have inline resources with the same name and different content in the same job, except for page definitions and form definitions.

6. The **job-page-count** attribute includes the separator pages when the host separator page is enabled, but the separator pages are not included in the **job-page-count** for any other jobs.
7. When the separator page support is enabled in the host but the data set does not contain the separator page, the receiver prints the job with InfoPrint Manager separators. To suppress the InfoPrint Manager separators, you can perform one of the following options:
   - Change the scripts or exits to set the **auxiliary-sheet-selection** job attribute to **none** permanently.
   - Set the **printer-start-sheet**, **printer-separator-sheet**, and **printer-end-sheet** job attributes to **none** for the actual destination.
   - Set the **auxiliary-sheet-selection** job attribute to **none** for the logical destination.

Technical description of multiple data set support

This section describes the sequence of the multiple data set support.

1. The MVS Download Receiver daemon receives a data set into a file.
2. The MVS Download Receiver daemon invokes the InfoPrint Manager script to spool the file. (You specified the script on the **-x** option when you started **mvsprsd**.)
3. If MVS user exit 15 or AFP Download Plus marked the file as part of a multiple data set job, the sample exit does not immediately spool the file. Instead, the sample exit calls a second program, which invokes ACIF to transform the file into MO:DCA-P if it is from Download for z/OS.
4. After the last file in a multiple data set job is received and transformed to MO:DCA-P for Download for z/OS, the sample exit invokes the **afpconcat** program to create one file with a comprehensive inline form definition and all the MO:DCA-P documents. The **afpconcat** program can combine the multiple MO:DCA-P files received from MVS Download into a single MO:DCA-P file. The comprehensive inline form definition contains medium maps (copy groups) from all of the form definitions associated with the individual data sets. The **formdef**s requested for each input file are merged into a single combined **formdef** that is placed inline in the output MO:DCA-P document.
5. If medium map names collide, the **afpconcat** program assigns unique names and updates references in the MO:DCA-P documents.
6. The **afpconcat** program merges the inline resource group, discarding page definitions and the original form definitions. Resources with matching resource names and types are compared to verify that they are identical.
7. The sample exit spools the combined MO:DCA-P file.

Resubmitting or deleting MVS Download Receiver orphaned files

Orphaned files are old files left in the Linux file system because you specified the **-k** or the **Retain command files** option in the InfoPrint Manager Management Interface when you started the MVS Download Receiver to submit the jobs, and errors prevented InfoPrint Manager from processing them. Their file names follow the MVS Download file name format, such as:

```
MVS_system_name.job_name.dataset_name.forms_name.yyddd.hhmmsstABCD.PRD.
```

You should either correct the problem that prevented the job from processing and then resubmit the job or delete the files.

Also, when enabling the MVS Download Receiver tracing using the **-t** flag or the Control MVS Download Receiver Tracing dialog in the InfoPrint Manager Management Interface, the MVS Download daemon saves the processing files for debugging purposes.

Resubmitting an orphaned file

If you started the **mvsprsd** daemon with the **-k** flag, InfoPrint saves print options downloaded with MVS data sets. By default, InfoPrint also stores the downloaded data set. InfoPrint saves both the data set and the print-options file in the directory you specify with the **-d** flag when you start the daemon. See Passing the parameter list from the mvsprsd daemon to the shell script, p. 209 for the format of the file names for the data set and print-options file. If errors occur that prevent InfoPrint from processing the job, you can correct the errors and then resubmit the job.

1. Go to the working directory where the failed job's JCL exists.
2. Run the `/usr/lpp/pd/bin/mvsprpsm.sh` command with these parameters (or create a shell script to run the command):

```
1st parameter : PRD file (line data from host)
2nd parameter: Copy the content of JCL file
3rd paramete: Logical Dest name
4th parameter: none (blank)
5th parameter: none (blank)
6th parameter: MVSPRSD port number
7th parameter: none (blank)
8th parameter: 0 (zero)
```

   For example:

```
#!/bin/bash
       JOB00838.SRV1.YAMASAK1.STEP1.PERFAIX.2005105.23235906438.PRD
       '"-odatat=line -ofileformat=record -occ=yes -occtype=a
         -ochars=GT10 -opagedef=P1V04863 -of=F10101LA -ocop=007
         -odatac=unblock -ojobn=MYJOB -ous=MYNAME -oprmode=SOSI2
         -otrc=yes -ono=BLDPSRV1
         -opa=forms=PERFAIX,class=K,destination=LOCAL,jobid=JOBID,
          OUTGRP=ONLY "' local '' '' 6001 '' 0 >>$PPID 2>>$PPID
```

Note

Multiple data set jobs cannot be resubmitted and must be resent from the host.

3. Administrative Procedures: Customizing for Special Jobs

- **Working with transforms**
- **Understanding the transform objects and the custom steps (transforms) subsystem**
- **Using print rules**
- **Printing PCL or PostScript to PSF destinations when using the CPSI transform engine: Specifying which paper bin to use**
- **Submitting PSF printer input to a PCL or PPDS printer**
- **Color and Grayscale Printing**
- **Working with fonts when using the CPSI processing engine**
- **Using global resource identifiers**
- **Configuring and using cut-sheet emulation**
- **Generating and submitting color mapping table source and output files**
- **Setting up high speed printers for spacing**
- **Using InfoPrint Manager with the InfoPrint 5000 to print AFP**
- **Using the InfoPrint 5000 and RICOH Pro VC series models with hot folders from InfoPrint Manager to print PostScript and PDF**
- **Transferring InfoPrint Manager objects across servers**

Working with transforms

InfoPrint provides transform programs for these data streams:

- Printer control language (PCL) data stream
- PostScript data stream
- Portable document format (PDF) data stream
- Tag image file form (TIFF) data stream
- Joint photographic experts group (JPEG) data stream
- Graphic interchange format (GIF) data stream
- ditroff data stream
- Double-byte ASCII and EUC data streams
- line-data data stream
- SAP data
- XML data

These transforms also support color management:

- Portable Document Format (PDF)
- Joint photographic experts group (JPEG)
- Graphics Interchange Format (GIF)
- Tag image file format (TIFF)

The transforms convert the incoming data stream into the Advanced Function Presentation (AFP) format.

When a job in any of these formats is submitted to print on a PSF actual destination, InfoPrint automatically invokes the appropriate transform. You can also run the transforms from the command line without printing the generated output. You might find this useful if you intend to print the job later. Jobs print more quickly if they have already been transformed.

Most transforms have flags or option keywords that you can use to specify processing information. For a list of the flags or options and values for each transform, see *RICOH InfoPrint Manager: Reference* or to the transforms' man pages. Configuration files, environment variables, and user-exit programs allow you to specify processing information for InfoPrint to use when it invokes the transform automatically. You can also use configuration files and environment variables instead of typing flags on the command line whenever you run a transform.

Customizing the PCL, PostScript, and PDF transforms

You can customize the transforms for PCL, PostScript, and PDF data by specifying values in the transform configuration files. Default PCL, PostScript, and PDF transform configuration files, p. 224 shows the default configuration files that InfoPrint provides. You can either modify these files or create your own.

Default PCL, PostScript, and PDF transform configuration files

Data Stream	*Transform*	*Configuration File*	*Daemon Configuration File*
PCL	**pcl2afp**	**usr/lpp/psf/pcl2afp/pcl2afp.cfg**	**usr/lpp/psf/pcl2afp/pcl2afpd.cfg**
PostScript	**ps2afp**	**usr/lpp/psf/ps2afp/ps2afp.cfg**	**usr/lpp/psf/ps2afp/ps2afpd.cfg**
PostScript	**ps2afp**	**/usr/lpp/psf/ps2afp/ps2afp.cfg**	**/usr/lpp/psf/ps2afp/agspdld.cfg**
PDF	**pdf2afp**	**usr/lpp/psf/ps2afp/ps2afp.cfg**	**usr/lpp/psf/ps2afp/ps2afpd.cfg**
PDF	**pdf2afp**	**/usr/lpp/psf/ps2afp/ps2afp.cfg**	**/usr/lpp/psf/ps2afp/agspdld.cfg**
PDF	**pdf2afp**	**usr/lpp/psf/ps2afp/ps2afp.cfg**	**usr/lpp/psf/ps2afp/apped.cfg**

Notice that the PostScript and PDF transforms use the same default configuration files.

Client configuration files

Sample PostScript or PDF client configuration file (ps2afp.cfg), p. 224 shows an example of a configuration file for the PostScript and PDF transforms. The configuration file for the PCL transform is similar, although it does not contain a **pcl_server_timeout** keyword. As a result, the PCL transform does not support the **-s** flag from the command line for the **pcl2afp** command described in Hierarchy of transform options, p. 236.

Sample PostScript or PDF client configuration file (ps2afp.cfg)

```
# ps2afp configuration file

##############################################################################
#                  CPSI, APPE and AGSPDL GENERAL SETTINGS
```

```
# KEYWORD             EQUIVALENT ps2afp FLAG                        PURPOSE
##############################################################################

port = 8251              # -P                  which TCP/IP port to use
appe_port = 8250         # -P        which TCP/IP port to use for appe
agspdl_port = 8254       # -P      which TCP/IP port to use for agspdl
server = localhost       # -S              which server to connect with

ps_max_memory = 96000K   # -M            maximum amount of memory that
                         #         PostScript uses; the 'K' is ignored

ps_job_timeout = 9999    # -j                maximum number of minutes
                         #                   to process PostScript job
ps_server_timeout = 360  # -s                maximum number of minutes
                         #                   server waits between jobs
ps_document_processor = cpsi  # -psproc              which ps document
                              #                       processor to use
pdf_document_processor = cpsi # -pdfproc            which pdf document
                              #                       processor to use

##############################################################################
#                     CPSI SPECIFIC SETTINGS
# KEYWORD             EQUIVALENT ps2afp FLAG                        PURPOSE
##############################################################################

ps_width = 8.5i          # -w                 width of generated image
ps_length = 11i          # -l                length of generated image
ps_x_offset = 0i         # -x                   left and right margins
ps_y_offset = 0i         # -y                   top and bottom margins
ps_resolution = 300      # -r             resolution of target printer
ps_output_type = IO1_G4  # -a            type of AFP image to generate
ps_imgsmall = 0.5i       #                  default value for imgsmall
ps_linesmall = 0.5i      #                 default value for linesmall
ps_txtsmall = 0.5i       #                  default value for txtsmall
#ps_is = 1                #                  the is afp compliance flag
#ps_compress_type = jpeg-sub
                         # -cmp             JPEG subsampled compression
                         #                    other options: jpeg, lzw

##############################################################################
#                     APPE SPECIFIC SETTINGS
# KEYWORD             EQUIVALENT ps2afp FLAG                        PURPOSE
##############################################################################

appe_width = 8.5i          # -w               width of generated image
appe_length = 11i          # -l              length of generated image
appe_x_offset = 0i         # -x                 left and right margins
appe_y_offset = 0i         # -y                 top and bottom margins
appe_resolution = 300      # -r           resolution of target printer
appe_output_type = IO1_G4  # -a          type of AFP image to generate
#appe_is = 1                #                the is afp compliance flag
#appe_compress_type = jpeg-sub
                           # -cmp           JPEG subsampled compression
                           #                  other options: jpeg, lzw
#appe_pagetype = DOCUMENT
                           #              type of afp structured field
                           #                  for each individual page
```

```
##############################################################################
#                     AGSPDL SPECIFIC SETTINGS
# KEYWORD             EQUIVALENT ps2afp FLAG                        PURPOSE
##############################################################################
agspdl_output_type = IO1_G4  # -a        type of AFP image to generate
agspdl_resolution = 300      # -r         resolution of target printer
agspdl_width = 8.5i          # -w             width of generated image
agspdl_length = 11i          # -l            length of generated image
agspdl_is     = 1            #              the is afp compliance flag
#agspdl_compress_type = lzw
                             # -cmp        JPEG subsampled compression
                             #           other options: jpeg, jpeg-sub
#agspdl_page_size_control
                             # -psc          page size control options
                             #      ArtBox, BleedBox, CropBox, TrimBox

##############################################################################
#                            APPE and CPSI

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##############################################################################
#                            APPE and CPSI

# ENABLING THE FINISHING FLAG ONLY
# If you want your users to be able to allow finishing operations,
# such as staple operations or punch operations, uncomment the
# following line.

# device_controls = finishing

##############################################################################
#                            APPE and CPSI

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##############################################################################
#                            APPE and CPSI

# ENABLING BOTH PLEX AND FINISHING
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND allow finishing
# operations, such as staple operations or punch operations,
# uncomment the following line.

# device_controls = plex,finishing

##############################################################################
#                            APPE and CPSI

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
```

```
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
# tray to use, uncomment the line below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.
# Replace size with one of the supported paper sizes \
(LETTER, LEGAL, A3, A4, B4, B5, LEDGER)
# device_controls = plex,inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ), \
inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)

##############################################################################
#                            APPE and CPSI

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output, AND allow finishing
# operations, AND specify which paper tray to use, uncomment the line
# below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.
# Replace size with one of the supported paper sizes \
(LETTER, LEGAL, A3, A4, B4, B5, LEDGER)

# device_controls = plex,finishing, \
inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ), \
inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)
```

Sample pcl2afp client configuration file (pcl2afp.cfg)

```
# pcl2afp configuration file


# KEYWORD             EQUIVALENT pcl2afp FLAG                    PURPOSE
##############################################################################


port = 8253                  # -P              which TCP/IP port to use
server = localhost           # -S          which server to connect with

pcl_max_memory = 6000K       # -M         maximum amount of memory that
                             #             PCL uses; the 'K' is ignored

pcl_job_timeout = 20         # -j             maximum number of minutes
                             #                  to process a PCL job

pcl_width = 8.5i             # -w              width of generated image
pcl_length = 11i             # -l             length of generated image
pcl_x_offset = 0i            # -x                left and right margins
pcl_y_offset = 0i            # -y                top and bottom margins
pcl_resolution = 300         # -r          resolution of target printer
pcl_output_type = IO1_G4     # -a         type of AFP image to generate

#pcl_is = 1                  # -is           the is afp compliance flag
#pcl_compress_type = lzw     # -cmp                     lzw compression
                             #            other options: jpeg, jpeg-sub
```

```
##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##############################################################################

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
# tray to use, uncomment the line below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the PCL bin number (between 1 and 59).

# You can list up to 20 mappings.

# device_controls = plex,inputX=(pcl_bin=Z),inputX=(pcl_bin=Z)
```

The configuration file uses keyword equivalents of the transform flags. See *RICOH InfoPrint Manager: Reference* for the keywords and values that you can specify in the configuration file.

Daemon configuration files

The PCL, PostScript, and PDF transforms all require a daemon to be running before you can use them. The `pcl2afpd` daemon manages the PCL interpreter portion of the `pcl2afp` transform. The `ps2afpd` daemon manages the PostScript or PDF CPSI interpreter portion of the `ps2afp` and `pdf2afp` transforms. The **apped** daemon manages the PDF APPE processing portion of the **ps2afp** and **pdf2afp** transforms. The **agspdld** daemon manages the PDF and PostScript Artifex GhostPDL (AGSPDL) processing portion of the **ps2afp** and **pdf2afp** transforms. For more information about these daemons, see *RICOH InfoPrint Manager: Reference*.

Each daemon, like the corresponding transform, uses a configuration file. The daemon configuration file can contain the same keywords as the transform configuration file, except `server`, plus some keywords that control the operation of the daemon. See *RICOH InfoPrint Manager: Reference* for the daemon keywords.

Sample AGSPDL daemon configuration file (agspdld.cfg)

```
# agspdld configuration file

##############################################################################
# KEYWORD                                                          PURPOSE
##############################################################################
```

```
port = 8254               #                    which TCP/IP port to use

work_directory = /var/psf/agspdl
                          #             directory to put work files into
log_file = /var/psf/agspdl/agspdl.log
                          #                  where to write log messages
log_size = 1024           #                       maximum log size in KB

ps_font_directory = /var/psf/agspdl/Resource
                          #                       list of resource files

ps_output_type = IO1_G4   #               type of AFP image to generate
ps_width = 8.5i           #                    width of generated image
ps_length = 11i           #                   length of generated image
ps_resolution = 300       #                 resolution of target printer
agspdl_is = 1             #                   the is afp compliance flag
agspdl_max_memory = 2097151
                          #                      max interpreter memory
#agspdl_compress_type = lzw
                          #                                 lzw compression
                          #               other options: jpeg, jpeg-sub
#agspdl_page_size_control
                          # -psc               page size control options
                          #          ArtBox, BleedBox, CropBox, TrimBox
```

Sample APPE daemon configuration file (apped.cfg)

```
# apped configuration file

##############################################################################
# KEYWORD                                                            PURPOSE
##############################################################################

port = 8250                     #              which TCP/IP port to use

appe_work_directory = /var/psf/appe
                                # directory to put APPE work files into

log_file = /var/psf/appe/apped.log
                                #    where to write daemon log messages
log_size = 100                  #                  maximum log size in KB

#transform_trace_file = /var/psf/appe/appeTraceFile.log
                                #  where to write APPE transform traces
                                #        comment out to disable tracing

#transform_trace_size = 200     #              maximum trace size in KB
                                #  use 0 to disable trace size checking

#transform_trace_level = MDFL@-1
                #                                       APPE traces level
                # MMST@-1                   output most trace statements
                # MINT@-1          output intermediate level of tracing
                # MDFL@-1               output default trace statements

#appe_fonts_path =
                #list of directories where extra APPE fonts are located
```

```
appe_font_map_files = /usr/lpp/psf/appe/fontmaps/FontMapSample.cfg

appe_width = 8.5i               #              width of generated image
appe_length = 11i               #             length of generated image
appe_x_offset = 0i              #                left and right margins
appe_y_offset = 0i              #                top and bottom margins
appe_resolution = 300           #          resolution of target printer
appe_output_type = IO1_G4       #         type of AFP image to generate
#appe_is = 1                    #            the is afp compliance flag
#appe_compress_type = jpeg-sub
                # JPEG subsampled compression. Other options: jpeg, lzw
#appe_pagetype = DOCUMENT
                # type of afp structured field for each individual page
```

Sample pcl2afp daemon configuration file (pcl2afpd.cfg)

```
# pcl2afpd configuration file

###############################################################################
# KEYWORD                                                             PURPOSE
###############################################################################

port = 8253                     #                  which TCP/IP port to use

work_directory = /var/psf/pcl2afp
                                #          directory to put work files into
log_file = /var/psf/pcl2afp/pcl2afpd.log
                                #               where to write log messages
log_size = 100                  #                    maximum log size in KB

notify = root                   #    who to notify if problems encountered
mail_command = /bin/mail
                                #       program used to send notifications

pcl_program = /usr/lpp/psf/bin/pcl6
                                #          name of PCL interpreter program
pcl_fonts_path = /usr/lpp/psf/pcl2afp
                                #          location of .FCO files for PCL6

pcl_max_memory = 6000K          #            maximum amount of memory that
                                #             PCL uses; the 'K' is ignored

pcl_job_timeout = 20            #                maximum number of minutes
                                #                     to process a PCL job

pcl_width = 8.5i                #                  width of generated image
pcl_length = 11i                #                 length of generated image
pcl_x_offset = 0i               #                    left and right margins
pcl_y_offset = 0i               #                    top and bottom margins
pcl_resolution = 300            #             resolution of target printer
pcl_output_type = IO1_G4        #            type of AFP image to generate

#pcl_is = 1                     #               the is afp compliance flag
#pcl_compress_type = lzw        #                           lzw compression
                                #            other options: jpeg, jpeg-sub
```

```
##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##############################################################################

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS

# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
# tray to use, uncomment the line below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the PCL bin number (between 1 and 59).
# You can list up to 20 mappings.

# device_controls = plex,inputX=(pcl_bin=Z),inputX=(pcl_bin=Z)
```

Sample ps2afp daemon configuration file for InfoPrint 4000 and InfoPrint 4100 printers (ps2afpd.cfg)

```
# ps2afpd configuration file

##############################################################################
# KEYWORD                                                          PURPOSE
##############################################################################

port = 8251              #                    which TCP/IP port to use

work_directory = /var/psf/ps2afp
                         #            directory to put work files into
log_file = /var/psf/ps2afp/ps2afpd.log
                         #                 where to write log messages
log_size = 100           #                      maximum log size in KB

notify = root            #       who to notify if problems encountered
mail_command = /bin/mail
                         #          program used to send notifications
ps_program = /usr/lpp/psf/bin/ps2afpi
                         #      name of PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                         #               name of PostScript interpreter
                         #                          initialization file
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
                /usr/lpp/psf/reslib
                         #            path to search for jobInit files
```

```
ps_max_memory = 96000K   #               maximum amount of memory that
                         #         PostScript uses; the 'K' is ignored

ps_job_timeout = 9999    #                   maximum number of minutes
                         #                   to process PostScript job
ps_server_timeout = 360  #                   maximum number of minutes
                         #                   server waits between jobs

ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
                         #                  list of font mapping files

ps_width = 8.5i          #                    width of generated image
ps_length = 11i          #                   length of generated image
ps_x_offset = 0i         #                      left and right margins
ps_y_offset = 0i         #                      top and bottom margins
ps_resolution = 300      #                 resolution of target printer
ps_output_type = IO1_G4  #                type of AFP image to generate
#following three parameters will apply only if threshsmall specified
ps_imgsmall = 0.5i       #                  default value for imgsmall
ps_linesmall = 0.5i      #                 default value for linesmall
ps_txtsmall = 0.5i       #                  default value for txtsmall
#ps_is = 1                #                   the is afp compliance flag
ps_compress_type = jpeg-sub  # JPEG subsampled compression. \
Other options: jpeg, lzw

##############################################################################

# Customize if desired
#pragma = jobInit 4000.ibm85lpi.tf.dt.ps; \
#            jobInit 4000.ibm85lpi.ta.ps;

#

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##############################################################################

# ENABLING THE FINISHING FLAG ONLY
# If you want your users to be able to allow finishing operations,
# such as staple operations or punch operations, uncomment the
# following line.

# device_controls = finishing

##############################################################################

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##############################################################################
```

```
# ENABLING BOTH PLEX AND FINISHING
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND allow finishing
# operations, such as staple operations or punch operations,
# uncomment the following line.

# device_controls = plex,finishing

##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
# tray to use, uncomment the line below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.
# Replace size with one of the supported paper sizes \
(LETTER, LEGAL, A3, A4, B4, B5, LEDGER)

# device_controls = plex,inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ), \
inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)

##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output, AND allow finishing
# operations, AND specify which paper tray to use, uncomment the line
# below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.
# Replace size with one of the supported paper sizes \
(LETTER, LEGAL, A3, A4, B4, B5, LEDGER)

# device_controls = plex,finishing,inputX=(size,type=ZZZ,weight=ZZZ, \
color=ZZZ), inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)
```

Sample ps2afp daemon configuration file for cut sheet IPDS printers (3160d.cfg)

```
# ps2afpd configuration file

##############################################################################
# KEYWORD                                                          PURPOSE
##############################################################################

port = 8252              #                    which TCP/IP port to use

work_directory = /var/psf/ps2afp2
                         #            directory to put work files into
log_file = /var/psf/ps2afp2/ps2afpd.log
                         #                  where to write log messages
log_size = 100           #                       maximum log size in KB

notify = root            #       who to notify if problems encountered
mail_command = /bin/mail
```

```
                          #          program used to send notifications

ps_program = /usr/lpp/psf/bin/ps2afpi
                          #      name of PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                          #               name of PostScript interpreter
                          #                          initialization file
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
                /usr/lpp/psf/reslib
                          #            path to search for jobInit files

ps_max_memory = 96000K    #               maximum amount of memory that
                          #         PostScript uses; the 'K' is ignored

ps_job_timeout = 9999     #                   maximum number of minutes
                          #                   to process PostScript job
ps_server_timeout = 360   #                   maximum number of minutes
                          #                   server waits between jobs

ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
                          #                  list of font mapping files

ps_width = 8.5i           #                    width of generated image
ps_length = 11i           #                   length of generated image
ps_x_offset = 0i          #                      left and right margins
ps_y_offset = 0i          #                      top and bottom margins
ps_resolution = 300       #                 resolution of target printer
ps_output_type = IO1_G4   #                type of AFP image to generate
ps_compress_type = jpeg-sub  # JPEG subsampled compression.\
Other options: jpeg, lzw

# Customize if desired
pragma = jobInit 2785.ibm85lpi.tf.dt.ps; \
            jobInit 2785.ibm85lpi.ta.ps; \
                        wglRotate yes;

##########################################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##########################################################################################

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##########################################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
# tray to use, uncomment the line below and fill in the correct values:
```

```
# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.
# Replace size with one of the supported paper sizes \
(LETTER, LEGAL, A3, A4, B4, B5, LEDGER)

# device_controls = plex,inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ), \
inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)
```

Improving performance in the PDF RIP

You can improve the performance of the PDF RIP by configuring it to run parts of itself in parallel. These changes improve the performance of the core RIP; for improving performance by running multiple PDF RIPs on a multi-processor, see Using the PDF Parallel RIP, p. 253. If you have a multi-processor server, make the changes described below to the PDF RIP daemon configuration file (**ps2afpd.cfg**) that are shown below in **boldface** text in Sample High Performance PDF Daemon configuration file using CPSI processing engine, p. 235.

Note

1. PDF RIP is only available with Adobe CPSI.
2. If your InfoPrint server has only a single processor, you should not make these configuration changes because they would result in reduced performance.
3. Both **-pragma bwModel=banded** and **-pragma beThread=yes** can also improve performance of the PostScript RIP.
4. If you are processing double-byte fonts (for example, with Kanji printing), you must set `pdfParallel=no`.

Sample High Performance PDF Daemon configuration file using CPSI processing engine

```
# ps2afpd configuration file

##############################################################################
# KEYWORD                                                          PURPOSE
##############################################################################

port = 8251                 #                    which TCP/IP port to use

work_directory = /var/psf/ps2afp
                            #            directory to put work files into
log_file = /var/psf/ps2afp/ps2afpd.log
                            #                  where to write log messages
log_size = 100              #                       maximum log size in KB

notify = root               #       who to notify if problems encountered
mail_command = /bin/mail
                            #          program used to send notifications
ps_program = /usr/lpp/psf/bin/ps2afpi
                            #      name of PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                            #              name of PostScript interpreter
                            #                          initialization file
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
```

```
                /usr/lpp/psf/reslib
                        #            path to search for jobInit files

ps_max_memory = 96000K  #               maximum amount of memory that
                        #         PostScript uses; the 'K' is ignored

ps_job_timeout = 9999   #                   maximum number of minutes
                        #                   to process PostScript job
ps_server_timeout = 360 #                   maximum number of minutes
                        #                   server waits between jobs

ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
                        #                  list of font mapping files

ps_width = 8.5i         #                    width of generated image
ps_length = 11i         #                   length of generated image
ps_x_offset = 0i        #                      left and right margins
ps_y_offset = 0i        #                      top and bottom margins
ps_resolution = 300     #                 resolution of target printer
ps_output_type = I01_G4 #                type of AFP image to generate
#following three parameters will apply only if threshsmall specified
ps_imgsmall = 0.5i      #                   default value for imgsmall
ps_linesmall = 0.5i     #                  default value for linesmall
ps_txtsmall = 0.5i      #                   default value for txtsmall
#ps_is = 1               #                  the is afp compliance flag
ps_compress_type = jpeg-sub  # JPEG subsampled compression.\
Other options: jpeg, lzw
pragma = pdfParallel=yes ;\
         beThread=yes    ;\
         bwModel=banded
```

Hierarchy of transform options

InfoPrint assigns a hierarchy to its use of flags and configuration-file information when it runs the PCL, PostScript, or PDF transform program. The following lists the order in which InfoPrint uses flags and information specified in configuration files:

1. Any values you specify on the command line, including values in the configuration file you specify with the **-C** flag.

   InfoPrint commands process flags from left to right. If you enter the same flag more than once, InfoPrint uses the last occurrence of the flag to determine the value to use. For example, if you specify this command:

   ```
   ps2afp -Cconfig.file -r240 -r300 myfile.ps
   ```

   The **ps2afp** command transforms the file using 300-pel resolution. InfoPrint ignores the resolution value specified in the configuration file `config.file` and the first **-r240** flag and value.

2. Values specified in the **pcl2afp.cfg** or **ps2afp.cfg** configuration file in the current directory, if one exists and if you enter the transform command directly. InfoPrint ignores values in the current directory if you invoke the transform command indirectly by submitting a job to print.
3. Values specified in the **pcl2afp.cfg** or **ps2afp.cfg** configuration file in your home directory, if one exists.

4. Values specified in the default transform command configuration file named in Default PCL, PostScript, and PDF transform configuration files, p. 224.
5. Values specified in a customized configuration file identified with the **-C** flag of the **apped**, **pcl2afpd**, or **ps2afpd** command when the transform daemon started.
6. Values specified in the default transform daemon configuration file named in Default PCL, PostScript, and PDF transform configuration files, p. 224.
7. Default values that are built into InfoPrint. These are the same as the transform defaults except that the default PostScript or PDF output type is an IM1 uncompressed image.

Stapling and punch options available from the ps2afp transform when using the CPSI transform engine

You can use the **ps2afp** transform to specify both staple and punch finishing options. However, you must:

1. Ensure that the IPDS (Intelligent Printer Data Stream) capable printer supports either staple or punch finishing options.
2. Ensure that the PostScript input file contains either staple or punch information.
3. Ensure that the keyword `finishing` is assigned to the **device_controls** option in the **ps2afpd.cfg** configuration file, as shown in the bold-faced example in Sample PostScript Daemon configuration file sections with staple and punch options, p. 237.
4. Ensure that the application or driver that generated the PostScript supports the subset of staple or punch operations enabled in the **ps2afp** transform. See Staple and punch PostScript operations supported by the ps2afp transform, p. 240 for a list of these staple and punch options.

Specifying finishing in the ps2afpd.cfg file

This shows how you can specify finishing values in the **ps2afp** data stream transform:

1. From a command line, access the **ps2afpd.cfg** file that is located in the **/usr/lpp/psf/ps2afp/** directory.
2. Using your editor of choice, edit the file.
3. Find the heading **#ENABLING STAPLING OR PUNCH FINISHING OPTIONS THROUGH THE TRANSFORM** and remove the hash mark (#) that comments out this line:

```
device_controls=finishing
```

4. Save and close the file.

Note

1. The **device_controls** option also accepts bin numbers and **plex** key words.
2. If you omit **plex**, the **ps2afp** transform creates duplex output.
3. If you omit bin numbers, the **ps2afp** transform creates output that indicates that the first input tray should be used.
4. Not all printers support using the staple and punch options that the **ps2afp** transform generates. If you send a document with staple or punch operations to an IPDS printer that does not support the request, InfoPrint Manager issues a message and completes the job, without either staples or punch holes. See your printer documentation to determine which (if any) staple or punch operations are available.

For further examples, see the **ps2afpd.cfg** file shipped with the latest version of InfoPrint Manager, which includes the **device_controls** option.

Sample PostScript Daemon configuration file sections with staple and punch options

```
# ps2afpd configuration file

# KEYWORD                                                                    PURPOSE
##############################################################################################

port = 8251                    #               which TCP/IP port to use

work_directory = /var/psf/ps2afp
                               #            directory to put work files into
log_file = /var/psf/ps2afp/ps2afpd.log
                               #                  where to write log messages

notify = root                  #       who to notify if problems encountered
mail_command = /usr/bin/mail
                               #          program used to send notifications
ps_program = /usr/lpp/psf/bin/ps2afpi
                               #      name of PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                               #              name of PostScript interpreter
                               #                           initialization file
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
                /usr/lpp/psf/reslib
 #            path to search for jobInit files

ps_max_memory = 96000K   #                maximum amount of memory that
 #         PostScript uses; the 'K' is ignored

ps_job_timeout = 9999    #                    maximum number of minutes
 #                    to process PostScript job
ps_server_timeout = 360  #                    maximum number of minutes
 #                    server waits between jobs

ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
 #                  list of font mapping files

ps_width = 8.5i              #   width of generated image
```

```
ps_length = 11i                #   length of generated image
ps_x_offset = 0i               #   left and right margins
ps_y_offset = 0i               #   top and bottom margins
ps_resolution = 300            #   resolution of target printer
ps_output_type = IO1_G4        #   type of AFP image to generate
#following three parameters will apply only if threshsmall specified
ps_imgsmall = 0.5i             #   default value for imgsmall
ps_linesmall = 0.5i            #   default value for linesmall
ps_txtsmall = 0.5i             #   default value for txtsmall
#ps_is = 1                     #   the is afp compliance flag
ps_compress_type = jpeg-sub  # JPEG subsampled compression.
 Other options: jpeg, lzw

##############################################################################

# Customize if desired
#pragma = jobInit 4000.ibm85lpi.tf.dt.ps; \
#            jobInit 4000.ibm85lpi.ta.ps;

#


# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# NOTE: Uncomment only one of the options below.

##############################################################################

# ENABLING THE FINISHING FLAG ONLY
# If you want your users to be able to allow finishing operations,
# such as staple operations or punch operations, uncomment the
# following line.

# device_controls = finishing

##############################################################################

# ENABLING THE PLEX COMMAND ONLY
# If you want your users to be able to choose between single-sided and
# double-sided (duplexed) output, uncomment the following line.

# device_controls = plex

##############################################################################

# ENABLING BOTH PLEX AND FINISHING
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND allow finishing
# operations, such as staple operations or punch operations,
# uncomment the following line.

# device_controls = plex,finishing

##############################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output AND specify which paper
```

```
# tray to use, uncomment the line below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.

# device_controls = plex,inputX=(size,type=ZZZ,weight=ZZZ,
color=ZZZ),inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)

##########################################################################

# ENABLING THE PLEX COMMAND AND MAPPING BINS TO TRAYS
# If you want your users to be able to both choose between single-
# sided and double-sided (duplexed) output, AND allow finishing
# operations, AND specify which paper tray to use, uncomment the line
# below and fill in the correct values:

# Replace the Xs with the number of the AFP tray (between 1 and 255).
# Replace the Zs with the appropriate values.

# device_controls = plex,finishing,inputX=(size,type=ZZZ,weight=ZZZ,
color=ZZZ),inputX=(size,type=ZZZ,weight=ZZZ,color=ZZZ)
```

Staple and punch PostScript operations supported by the ps2afp transform

The following tables list the subset of PostScript information transformed to Advanced Function Presentation (AFP) MO:DCA-P structured fields and triplets to support staple and punch operations. The changes described below should only be done by someone with extensive knowledge of the PostScript language, such as someone writing their own PostScript driver.

The `<< /Staple 3` information below is a particular example drawn from Default staple locations, p. 241 and Additional PostScript staple details, p. 241. Both tables contain pertinent information for adding stapling operations to your PostScript data stream.

```
{ << /Staple 3  /StapleDetails << /Type 17 /Position 17
        >> >> setpagedevice }
{ << /Staple 3  /StapleDetails << /Type 17 /Position 18
        >> >> setpagedevice }
```

This example indicates that the printer will place two staples at the top of a page, whether the page has either a Portrait or a Landscape orientation.

/Staple 3

The key that introduces the StapleDetails Dictionary.

/Type 17

A required key reserved for future use.

/Position

A key that indicates the placement of the staples, using the settings displayed in Additional PostScript staple details, p. 241.

The `<< /Punch 4` information below is a particular example drawn from Default punch locations, p. 243, Additional PostScript punch details, p. 243, and Punch Setting for the /Pattern value, p. 244. All three tables contain pertinent information for adding stapling operations to your PostScript data stream.

```
{ << /Punch 4 /PunchDetails << /Type 7 /HoleType 0 /Pattern 0 /Position 1
        >> >> setpagedevice }
{ << /Punch 4 /PunchDetails << /Type 7 /HoleType 0 /Pattern 0 /Position 5
        >> >> setpagedevice }
```

This example indicates that the printer will place a punch on the left side of a page, whether the page has either a Portrait or a Landscape orientation.

/Punch 4

The key that introduces the PunchDetails Dictionary.

/Type 7

A required key reserved for future use.

/Pattern 0

A key that indicates the default punch setting, using the settings displayed in Punch Setting for the /Pattern value, p. 244.

Note

Most PostScript drivers use `/Pattern 0` because Finishers tend to have one particular setting. `/Pattern 0` works with all Finishers. If you specify `/Pattern 6` in the PostScript driver and your Finisher only supports two-hole punch, InfoPrint Manager will print the job and generate an error message without performing the finishing.

/Position

A key that indicates the placement of the staples, using the settings displayed in Additional PostScript punch details, p. 243.

Default staple locations

Staple	Key name	PostScript Staple
Do not staple	None	/Staple 0
Staple	Not supported	/Staple 3

Additional PostScript staple details

Staple details	Key name	PostScript staple details
Do not staple	None	**Portrait:** /Position 0 **Landscape:** /Position 0
Single left diagonal	LeftDiagonal	**Portrait:** /Position 1 **Landscape:** /Position 5

Single left horizontal	LeftHorizontal	**Portrait:** /Position 3 **Landscape:** /Position 7
Single left vertical	LeftVertical	**Portrait:** /Position 2 **Landscape:** /Position 6
Single right diagonal	RightDiagonal	**Portrait:** /Position 9 **Landscape:** /Position 13
Single right horizontal	RightHorizontal	**Portrait:** /Position 11 **Landscape:** /Position 15
Single right vertical	RightVertical	**Portrait:** /Position 10 **Landscape:** /Position 14
Double left	DoubleLeft	**Portrait:** /Position 4 **Landscape:** /Position 8
Double right	DoubleRight	**Portrait:** /Position 12 **Landscape:** /Position 16

Double top	DoubleTop	**Portrait:** /Position 317 **Landscape:** /Position 18
Booklet	Booklet	**Portrait:** /Position 20 **Landscape:** /Position 20

Default punch locations

Punch	Key name	PostScript Punch
Do not punch	None	/Punch 0
Punch	Not supported	/Punch 4

Additional PostScript punch details

Punch details	Key name	PostScript punch details
Do not punch	None	**Portrait:** /Position 0 **Landscape:** /Position 0
Left	PunchLeft	**Portrait:** /Position 1 **Landscape:** /Position 5
Top	PunchTop	**Portrait:** /Position 3 **Landscape:** /Position 7

Right	PunchRight	**Portrait:** /Position 2 **Landscape:** /Position 6
Bottom	PunchButtom	**Portrait:** /Position 4 **Landscape:** /Position 8

Punch Setting for the /Pattern value

Punch	Key Name	PostScript Punch Details
Default	Punch*	/Pattern 0
2–holes	Punch*2	/Pattern 6
3–holes	Punch*3	/Pattern 5
4–holes	Punch*4	/Pattern 7

Collate PostScript options available from the ps2afp transform when using the CPSI transform engine

When a PostScript file contains the collate key, the **ps2afp** transform recognizes that key, and the resulting AFP output has the copies collated. Not all PostScript printers natively support the PostScript collate key, but if you used a PSF other-driver DSS, InfoPrint Manager exploits this new transform feature and creates PCL that you can print on that same printer and get collated copies.

Note

- The collation function requires more disk space, so increase the space in `/var/psf` to store the temporary file for collation.

Customizing the TIFF, JPEG, and GIF transforms

You can customize the transforms for TIFF, JPEG, and GIF data by:

- Specifying values in the transform configuration files
- Setting the transform environment variables

Default TIFF, JPEG, and GIF transform configuration files, p. 245 shows the default configuration files that InfoPrint provides. You can either modify these files or create your own.

Default TIFF, JPEG, and GIF transform configuration files

Data Stream	*Transform*	*Configuration Files*
TIFF	**tiff2afp**	**/usr/lpp/psf/tiff2afp/tiff2afp.cfg**
JPEG	**jpeg2afp**	**/usr/lpp/psf/jpeg2afp/jpeg2afp.cfg**
GIF	**gif2afp**	**/usr/lpp/psf/gif2afp/gif2afp.cfg**

Sample configuration file

Sample TIFF configuration file, p. 245 shows an example of a configuration file for the TIFF transform. The configuration files for the JPEG and GIF transforms are similar.

Sample TIFF configuration file

```
a=ioca10
choice=full
cmp=g4
ms=5120
msf=0.01
nosniff=yes
noterm=yes
v=yes
pagetype=page
r=600
x=0
y=0
```

The configuration file uses options in the same format as the transform options that you specify on the command line, except that command line options begin with a dash (–). See *RICOH InfoPrint Manager: Reference* for the options and values that you can specify in the configuration file.

Transform environment variables

Many of the TIFF, JPEG, and GIF transform options have equivalent environment variables. Environment variables have this format:

transform_option

where*transform* is the name of the transform command in uppercase and *option* is the command line option without the initial dash (–). For example, the environment variable that corresponds to the **–scale** option of the **jpeg2afp** transform is **JPEG2AFP_scale**. See *RICOH InfoPrint Manager: Reference* for the options and values that you can specify as environment variables.

Hierarchy of transform options

InfoPrint assigns a hierarchy to its use of options and configuration-file information when it runs the TIFF, JPEG, or GIF transform program. The following lists the order in which InfoPrint uses command–line options and information specified in configuration files:

1. Any values you specify on the command line, including values in the configuration files you specify with the **-C** option.

   InfoPrint commands process options from left to right. If you enter the same option more than once, InfoPrint uses the last occurrence of the flag to determine the value to use. For example, if you specify this command:

```
tiff2afp -Cconfig.file -r240 -r300 myfile.tif
```

   the **tiff2afp** command transforms the file using 300-pel resolution. InfoPrint ignores the resolution value specified in the configuration file `config.file` and the first **-r240** flag and value.

2. Values specified in the default transform command configuration file named in Default TIFF, JPEG, and GIF transform configuration files, p. 245.

3. Transform environment variables that you set with the **export** command, if you enter the transform command directly. InfoPrint ignores these environment variables if you invoke the transform command indirectly by submitting a job to print.

4. Transform environment variables in the **/etc/profile.d/ipm_environment.sh** file in your home directory.

5. Transform environment variables in the **/etc/environment** file.

6. Default values that are built into InfoPrint. These are the same as the transform defaults.

Customizing the img2afp transform

The **img2afp** transform can be used to transform color (FS45) images into monochrome bilevel (FS10) images for printing. This transform is valuable for transforming color images so that they print well in black-and-white. You can customize the **img2afp** transform only by specifying options on the command line. Unlike other transforms shipped with the InfoPrint Manager, this transform is not invoked automatically. Instead, you must specify it at the command line to preprocess the data before printing. For specific information about the syntax and use of this command, see *RICOH InfoPrint Manager: Reference*.

While you can create an **img2afp.cfg** file, InfoPrint Manager supplies a series of **img2afp.cfg** configuration files that are based on the printer model. For example, the **img2afp.4000–708cfg** file on an InfoPrint Manager for AIX server contains these settings:

```
#img2afp Infoprint 4000-708 configuration file

a=fs10
respath=.:/usr/lpp/psf/config
thresh=4000-708.ibm106lpi.ta.ps
gcorr=4000-708.ibm106lpi.tf.dt.ps
pagetype=page
v=yes
r=600
```

img2afp.cfg files observe these basic rules:

Note

1. Any item specified in a configuration file can be overridden on the command line.
2. Configuration file syntax is *attribute* = *value*.
3. A hash mark (#) indicates a comment to the end of the line.
4. All attribute names are the same as the command-line names.
5. For options that do not take values, you must type `yes` in the configuration file (as shown below).

```
#sample img2afp configuration file

v = yes  #equivalent to -v flag
a = fs10 #bilevel output
```


To invoke the transform from the command line, you must specify at least:

- **-a fs10** to indicate bilevel output .
- **-thresh** {*name*} to specify the PostScript level 3 halftone dictionary that you are using.
  When you install some printers with InfoPrint Manager, the product makes PostScript level 3 halftone dictionaries available on your server. For example, an InfoPrint HD1-HD2, HD3-HD4, HD5-HD6, HS1, HS3, MD1-MD2, MS1, PS-1, or PD1-PD2 4100 printer models provides these:
  - **/usr/lpp/psf/config/4100-*model*.ibm71lpi.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm85lpi.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm106lpi.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm141lpi.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm71lpiRot90.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm85lpiRot90.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm106lpiRot90.ta.ps**
  - **/usr/lpp/psf/config/4100-*model*.ibm141lpiRot90.ta.ps**
- **-gcorr** *filename* to specify grayscale mapping tables to halftone the grayscale and color images to bilevel for output.
  When you install particular printers with InfoPrint Manager, you also install the following grayscale mapping tables for each type of PostScript level 3 halftone dictionary option.

and **-gcorr** and **-thresh** must be used to select an appropriate halftone.

Limitations of the img2afp transform

When using the **img2afp** transform, be aware of these limitations:

- Only jobs that contain either pure IOCA data or IOCA overlaid by other bilevel-compatible OCAs print correctly.
- **img2afp** does not support one bit per spot CMYK data (FS42) data conversions.

Examples

For each of these examples, the user has an AFP input file of an FS45 image that they are converting to an FS10 file for printing on a black-and-white printer. To obtain their FS45 input file, they might have used either the **tiff2afp**, **gif2afp**, **jpeg2afp**, **pdf2afp**, or **ps2afp** transform to convert an image file from one of those formats

To convert an FS45 image (`sampleFS45.afp`) to an FS10 image (`outFS10.afp`) from an InfoPrint command line for printing on a Ricoh printer, type this on the command line:

```
img2afp -a fs10 -thresh 4322.ibm106lpi.ta.ps \
-gcorr 4322.ibm106lpi.tf.dt.ps sampleFS45.afp -o outFS10.afp
```


To print the resulting FS10 image (`outFS10.afp`) on a Ricoh printer, type this on the command line:

```
pdpr -p ip21-ld outFS10.afp
```

To convert an FS45 image (`DSC011FS45.afp`) to an FS10 image (`DSC011FS10.afp`) with standard halftones from an InfoPrint command line for printing on an InfoPrint 4100–HD3/HD4 Advanced Function Printing System printer, type this on the command line:

```
img2afp -a fs10 -thresh 4100-HD3-HD4.ibm106lpi.ta.ps -gcorr \
4100-HD3-HD4.ibm106lpi.tf.lr.ps DSC011FS45.afp -o DSC011FS10.afp
```

To print the resulting FS10 image (`DSC011FS10.afp`) on an InfoPrint 4100–HD3/HD4 Advanced Function Printing System printer, type this on the command line:

```
pdpr -p silver-ld DSC011FS10.afp
```

To convert an FS45 image (`DSC015FS45.afp`) to an FS10 image (`DSC011FS10.afp`) with dark halftones from an InfoPrint command line for printing on an InfoPrint 4000 Advanced Function Printing System printer, type this on the command line:

```
img2afp -a fs10 -thresh 4000.ibm106lpi.ta.ps -gcorr \
4000.ibm106lpi.tf.ap.ps DSC015FS45.afp -o DSC011FS10.afp
```

To print the resulting FS10 image (`DSC011FS10.afp`) on an InfoPrint 4000 Advanced Function Printing System printer, type this on the command line:

```
pdpr -p ip40001-ld DSC011FS10.afp
```

Working with the transform for line data

InfoPrint Manager includes a line-data transform program that transforms line data into the Advanced Function Presentation (AFP) data stream. Converting the line data lets you print it on high-speed AFP printer devices.

About line data

Read through the following sections to become familiar with line-data terms.

Note

Throughout this chapter references are made to ACIF because the **line2afp** command is linked to the **acif** command. When you use this transform and if errors occur, you might receive some error messages that make reference to ACIF. See the *ACIF User's Guide* for more information about the ACIF command.

What Is line data?

Line data is simply unformatted data — that is, data with few or no embedded control characters to create font changes, page breaks, and so on. The line data is typically created by an application program. For example, an application might produce a bank statement that has line data indicating an account's daily balance.

The InfoPrint line-data transform lets you format these kinds of data with a page definition and form definition for printing on InfoPrint-managed printer devices:

- Single-byte ASCII data with no embedded control characters except for newlines.
- Single-byte ASCII data that contains ANSI carriage-control characters, table-reference characters, or both.
- Single-byte ASCII data that contains carriage returns and form feed controls if you specify the **asciinp** or **asciinpe** user-exit program.

When you transform the line-data job, you must specify a page definition and a form definition.

Note

1. You cannot use the line-data transform when printing files containing Proprinter ASCII.
2. When the **line2afp** command processes any unrecognized or unsupported keyword, **line2afp** issues a message, ignores the keyword, and continues processing any remaining keywords. The **line2afp** command then stops processing.
3. InfoPrint also lets you transform and print DBCS ASCII data without a page definition. See *RICOH InfoPrint Manager: Reference* or the `db2afp` man page.

ANSI and machine carriage controls

In many environments (including IBM mainframes and most minicomputers), printable data normally contains a carriage control character. The carriage control character acts as a vertical tab command to position the paper at the start of a new page, at a specified line on the page, or to control skipping to the next line. The characters can be one of two types: ANSI carriage control or machine carriage control.

ANSI carriage control characters

The most universal carriage control is ANSI, which consists of a single character that is a prefix for the print line. The standard ANSI characters are:

ANSI
Action

space

Single space the line and print

0

Double space the line and print

-

Triple space the line and print

+

Do not space the line and print

1

Skip to channel 1 (the top of the form, by convention)

2-9

Skip to a hardware-defined position on the page

A,B,C

Defined by a vertical tab record or forms control buffer (FCB)

All ANSI control characters perform the required spacing before the line prints. Using **line2afp** command keywords and values, ANSI controls can be encoded in EBCDIC (`cctype`=a) or in ASCII (`cctype`=z).

Machine carriage control characters

Machine carriage controls were originally the actual hardware control commands for InfoPrint printers and are often used on non-InfoPrint systems. Machine controls are literal values, not symbols. They are not represented as characters in any encoding, and therefore, machine controls cannot be translated. Typical machine controls are:

Machine
Action

X'09'

Print the line and single space

X'11'

Print the line and double space

X'19'

Print the line and triple space

X'01'

Print the line (do not space)

X'0B'

Space one line immediately (do not print)

X'89'

Print the line, then skip to channel 1 (top of form, by convention)

X'8B'

Skip to channel 1 immediately (do not print)

Machine controls print before doing any required spacing. There are many more machine control commands than ANSI. Carriage controls might be present in a file, but every record in the file must contain a carriage control if the controls are to be used. If the file contains carriage controls, but the

line2afp command keyword and value `cc=no` is specified, the carriage controls are treated as printing characters. If no carriage controls are specified, the file prints as though it were single spaced.

Variable-length and fixed-length files

The line-data transform program needs to know two things about a file in order to transform it:

- The length of each print record
- The kind of carriage control used

Some files contain information in each record that describes the record length; these are called variable-length files. Other files require an external definition of length; these are called fixed-length files.

For variable- and fixed-length files using length prefixes, MO:DCA–P structured fields are treated as a special case. All such structured fields are self-identifying and contain their own length. They need not contain a length prefix to be correctly interpreted but are processed correctly if there is a length prefix that matches the structured field length.

Variable-length files

Variable-length files can use a length prefix, which means they contain a prefix that identifies the length of the record in the file. Each record contains a two-byte field that gives the length of the record. If the record contains a length, that length must be a prefix for each record and it must be a 16-bit binary number that includes the length of the 2-byte length prefix. Use the `fileformat=record` keyword and value to identify files with length prefixes.

Variable-length files may use a separator or delimiter to indicate the end of a record, instead of using a length prefix. All of the bytes up to, but not including, the delimiter are considered part of the record. The default delimiter is X'0A'. If the file uses EBCDIC encoding, the default newline character is X'25'. Use the `fileformat=stream` keyword and value to designate files that use newlines to indicate record boundaries.

You can also specify the explicit newline characters that InfoPrint should use to delimit records. This is specified as a subparameter of the **fileformat** keyword. For example, **fileformat=stream, (newline= CRLF,ASCII)**. For details, see the description of the **line2afp** command in the *RICOH InfoPrint Manager: Reference*.

If you do not explicitly specify the newline characters, InfoPrint reads the first six bytes and tests for all ASCII characters (code points from X'00' to X'7F') to determine if a file is encoded in ASCII or EBCDIC. If no non-ASCII characters are found, line-data transform program assumes the file uses the ASCII newline character, X'0A'. Otherwise, the transform assumes the file uses the EBCDIC newline character, X'25'. Because an input file can misguide the line-data transform, you should specify newline on the file format keyword to make sure that InfoPrint will use the correct characters.

Fixed-length files

Fixed-length files contain records that are all the same length. No other separators, prefixes, or self-identifying information exists that indicates the record length. You must know the record length and use the `fileformat=record`,nnn keyword and value, where nnn represents the length of each record.

How the imageout keyword affects processing

The line-data transform converts image formats (IM1) in the input file, in overlays, and in page segments to uncompressed IOCA format, if **line2afp** `imageout=ioca` keyword and value is specified (the default). An uncompressed IOCA image might use a significantly higher number of bytes than an IM1 image and might take more processing time to convert, especially for shaded or patterned areas. Although IOCA is the MO:DCA-P standard for image data, and some data stream receivers might require it, all products might not accept IOCA data. All software products from the Ricoh do, however, accept IOCA data as well as IM1 image data.

We recommend that you specify `imageout=asis`, unless you have a specific requirement for IOCA images.

Customizing the line data transform

InfoPrint provides several sample programming exits to assist you in customizing the line-data transform. Use of the programming exits is optional. You specify the names of the exit programs with the `inpexit`, `outexit`, and `resexit` keywords. Each of these keywords is described in *RICOH InfoPrint Manager: Reference* or the `line2afp` man page. If you are using InfoPrint's sample user exits or your own user exit to convert incoming data streams from one code page to another code page, then the **INPCCSID** parameter indicates the code page of the incoming data stream. The **OUTCCSID** parameter indicates the code page of the output from UCONV.

Note

If you have written any user-exit programs that use ICONV, you must alter them to use UCONV instead. See the *RICOH InfoPrint Manager: Reference* for more information.

InfoPrint provides these sample exits:

/usr/lpp/psf/acif/apkinp.c

Input record exit

/usr/lpp/psf/acif/apkout.c

Output record exit

/usr/lpp/psf/acif/apkres.c

Resource exit

In addition, InfoPrint provides these user input record exits to translate line-data data streams:

/usr/lpp/psf/acif/apka2e.c

Converts the ASCII stream data into the EBCDIC stream data.

Note

The default conversion is from code page 805 to code page 037. You can override the defaults with the **inpccsid** and **outccsid** parameters.

/usr/lpp/psf/acif/asciinp.c

Converts unformatted ASCII data that contains carriage returns and form feeds into a record format that contains ANSI carriage control characters. This exit encodes the ANSI carriage control character in byte 0 of every record.

/usr/lpp/psf/acif/asciinpe.c

Converts unformatted ASCII data into a record format as does **asciinp.c** and then converts the ASCII stream data into EBCDIC[1] stream data.

/usr/lpp/psf/acif/dbblank

Processes EBCDIC double-byte line data downloaded from the z/OS spool by adding an extra blank to the end of the input record if:

1. The last byte in the record is a blank (x'40' in EBCIDIC).
2. The second to the last byte is **not** a blank.
3. The input record is line data; that is, it is not a structured field.

This exit also updates the record length by 1 if the blank is added. The exit has no way of determining if the data is DBCS or not, so this will happen to every input record that ends in a single blank. The exit assumes that the input data is EBCIDIC, and that the output data is also EBCIDIC. In other words, the exit does **not** do any code page translation, but it simply adds an extra blank if needed.

Note

1. This exit is only intended to be used if the spool file consists of double-byte or mixed single- or double-byte data that has blank truncation. Adding a blank to other types of data files could cause formatting errors, depending on how the page definition is coded.
2. This exit only checks for a **single** trailing blank, so if a different number of odd blanks is present in the data, it is up to the user to make sure the data is formatted with the correct font.

The C language header file for all line-data transform exit programs is also provided:

- **/usr/lpp/psf/acif/apkexits.h**

along with the build rules for the line-data transform user exits:

- **/usr/lpp/psf/acif/Makefile**

The source code for the sample exits is provided in `/usr/lpp/psf/acif`. You can modify these source files or create your own. In order to compile and build your exit, you must have an C/C++ compiler for the Linux operating system. You can use the `/usr/lpp/psf/bin/Makefile` as an example of how to compile the exits.

For more information about compiling user exit programs, see Compiling and installing the user-exit program, p. 55.

Using the PDF Parallel RIP

InfoPrint Manager provides the PDF Parallel RIP only when using the CPSI processing engine to improve the speed of PDF transforms for most color jobs. InfoPrint Manager converts a PDF input file into a PS file that is divided into separate pages, and distributes these pages to several RIPs running in parallel. Then, it combines the resulting AFP data stream into a printable document.

If you define two RIPs on the same InfoPrint Manager server for a PDF input file with pages of a uniform complexity, the Parallel PDF RIP client can render this file into AFP in half the time of the standard PDF transform. While the greatest performance improvements occur for jobs printing on the InfoPrint 5000 printer, you can use the Parallel PDF RIP client on any printers that produce AFP data.

Note

You cannot use the PDF parallel RIP with the PDF APPE processing engine.

Installing the PDF Parallel RIP client

The Parallel PDF RIP client (**pdfprs**) replaces the standard **pdf2afp** client on your system.

The **pdfprs** component consists of these:

- The **pdfprs** executable in the **/usr/lpp/psf/bin** directory
- The **pdf2afp** configuration file in the **/usr/lpp/psf/ps2afp** directory.

The Parallel PDF RIP client is best used on a server or servers, which can have from four to 16 processors. We recommend you set up a configuration to allow for system memory use like that displayed in Processors required to use the Parallel PDF RIP client, p. 254.

Processors required to use the Parallel PDF RIP client

Processors Allocated	Function Performed
1	network traffic, I/O processes, and general system overhead.
1	**pdfprs** client as defined in Sample configuration file for the pdfprs client, p. 256.
1	This first RIP transforms PDF into PostScript data. Each RIP is likely to use an entire processor to RIP the PDF files.
1 to *n*	For each additional daemon RIP as defined in RIP Daemon configuration files, p. 257.

The most basic configuration (the **pdfprs** client and two additional RIPs) requires a minimum of four processors. Make sure that your configuration can support this feature before installing it and submitting jobs through the InfoPrint Manager **pdpr** print command.

You must create the first RIP to transform PDF input data to PostScript data, then use at least a two-RIP configuration on your InfoPrint Manager server to transform the PDF into AFP data. To demonstrate how this support can be expanded by adding RIPs, this example creates a three-RIP configuration, with the first RIP transforming PDF input data into PostScript data. These RIPs still use the PostScript daemon (**ps2afpd**), just like the standard PDF transform.

From an InfoPrint Manager window, complete this procedure:

1. Create a directory for the PDF Parallel RIP client to distinguish it from the standard PDF data stream transform:

```
mkdir /var/psf/pdf2afp;chmod 777 /var/psf/pdf2afp
```

2. Create working directories for each of the new PDF RIPS:

```
mkdir /var/psf/pdf2afp/1;chmod 777 /var/psf/pdf2afp/1
```

```
mkdir /var/psf/pdf2afp/2;chmod 777 /var/psf/pdf2afp/2
```

```
mkdir /var/psf/pdf2afp/3;chmod 777 /var/psf/pdf2afp/3
```

```
mkdir /var/psf/pdf2afp/4;chmod 777 /var/psf/pdf2afp/4
```

3. Install new configuration files for each of these RIPS:
   - **/var/psf/pdf2afp/8261.cfg**
   - **/var/psf/pdf2afp/8262.cfg**
   - **/var/psf/pdf2afp/8263.cfg**
   - **/var/psf/pdf2afp/8264.cfg**

   Note that for each new configuration file, you must change the **port**, **work_directory**, and **log_file** values. These three parameters are highlighted in Sample PDF parallel transform (8264.cfg) configuration file for color, p. 257 so you can see the directory structure recommended for this configuration.

4. From either the command line or the **ps2afpd.cfg** file, start daemons for each configuration file:

```
ps2afpd -C /var/psf/pdf2afp/8261.cfg
```

```
ps2afpd -C /var/psf/pdf2afp/8262.cfg
```

```
ps2afpd -C /var/psf/pdf2afp/8263.cfg
```

```
ps2afpd -C /var/psf/pdf2afp/8264.cfg
```

Note

In the path where the working directories reside, you can create a **startcfg** script similar to this example to do this task:

```
#!/bin/sh
cd /var/psf/ps2afp
if [$# -gt 0 ]
then
for i in $*
do
/usr/lpp/psf/bin/ps2afpd -C 826$i.cfg
done
else
for i in 1 2 3 4
do
/usr/lpp/psf/bin/ps2afpd -C 826$i.cfg
done
fi
```

5. To make sure that the pdfprs command is linked to the pdf2afp command, you must enter these commands:
   1) Access the correct directory by specifying: `cd /usr/lpp/psf/bin`
   2) Create the necessary logical link to submit jobs to the PDF parallel RIP by specifying: `ln -sf pdfprs pdf2afp`

Note

If you need to restore the old **pdf2afp** client, you can specify this from a command line: `ln -sf ps2afp pdf2afp`.

6. To make sure that the **pdf2afp** configuration file is used, you must edit the PDF mapping in the `/usr/lpp/psf/config/transform.cfg`:

```
map pdf         ps
to
map pdf         pdf
```

Note

If you need to restore the standard **ps2afp** configuration, you can change the mapping to ps:

```
map pdf         ps
```

This configuration will automatically invoke four RIPs and generally run faster than the standard PDF transform.

3

Customizing the PDF Parallel transform

You can customize the transforms for PDF data by specifying values in the transform configuration files. Default parallel PDF transform configuration files, p. 256 shows the default configuration files that InfoPrint provides. You can either modify these files or create your own.

Default parallel PDF transform configuration files

Data Stream	*Transform*	*Configuration File*	*Daemon Configuration Files*
PDF	**pdf2afp**	**/usr/lpp/psf/ps2afp/pdf2afp.cfg** The **pdfprs** configuration file.	**/var/psf/pdf2afp/8261.cfg** **/var/psf/pdf2afp/8262.cfg** **/var/psf/pdf2afp/8263.cfg** **/var/psf/pdf2afp/8264.cfg** Note This represents the 3–RIP configuration of our example, with the first RIP serving as the PDF to PostScript RIP.

Sample configuration file for the pdfprs client

The following figure shows an example of a configuration file for the PDF parallel transform client. Note the three parameters that have been highlighted to show values you want to specify exactly as shown to enhance the performance of the parallel PDF RIP.

Note

The PDF parallel transform cannot be used together with the APPE transform.

Sample PDF parallel transform (pdf2afp.cfg) configuration file

```
# pdf2afp.cfg configuration file
# KEYWORD          EQUIVALENT ps2afp FLAG                    PURPOSE
#############################################################################
width =  8.5i             # -w                 width of generated image
length =  11i             # -l                length of generated image
```

```
x_offset = 0i              # -x                    left and right margins
y_offset = 0i              # -y                    top and bottom margins
resolution = 600           # -r                     resolution of printer
output_type = FS45         #               type of AFP image to generate
job_timeout = 120          # maximum number of minutes to process a job
number_of_rips = 3         #                         how many rips to use
[pdf]                      #       location of PDF -> PS conversion rip
 port = 8261
 server = server1
[rip1]                     #                         location of this rip
 port = 8262
 server = server1
[rip2]                     #                         location of this rip
 port = 8263
 server = server1
[rip3]                     #                         location of this rip
 port = 8264
 server = server1
```

This sample configuration file uses keyword equivalents of the transform flags. See *RICOH InfoPrint Manager: Reference* for the keywords and values that you can specify in the configuration file.

Note

For this sample **pdf2afp.cfg** configuration to work correctly, you must have a configuration with a minimum of four processors dedicated to this task.

RIP Daemon configuration files

The PDF parallel transform requires that at least three daemons be running to manage the PostScript interpreter portion of the `pdf2afp` transform. To simplify your configuration, InfoPrint recommends naming these daemons after the ports that they are running on. For this document, the configuration example begun in Sample PDF parallel transform (pdf2afp.cfg) configuration file, p. 256 will use the names **8261.cfg**, **8262.cfg**, **8263.cfg**, and **8264.cfg**. For more information about this daemon, see *RICOH InfoPrint Manager: Reference*.

The daemon, like the corresponding transform, uses configuration files. The daemon configuration files reside in the **/var/psf/** directory. It is recommended that you create a **/pdf2afp** subdirectory where all support related to the Parallel PDF RIP client (**pdfprs**) can reside.

The following figure shows an example of a configuration file for the PDF parallel transform client that RIPs a color (**FS45**) file.

Sample PDF parallel transform (8264.cfg) configuration file for color

```
# 8264.cfg configuration file
# KEYWORD          EQUIVALENT ps2afp FLAG                    PURPOSE
#############################################################################
port =  8264       #                          which TCP/IP port to use
work_directory =  /var/psf/pdf2afp/4
                   #              directory to put work files in
log_file = /var/psf/pdf2afp/4/ps2afpd.log
                   #                 where to write log messages
notify = root      #         notify if problems are encountered
mail_command = /usr/bin/mail
                   #         program used to send notifications
ps_program = /usr/lpp/psf/bin/ps2afpi
                   # name of the PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                   #         name of the PostScript interpreter
                   #                         initialization file
```

```
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
                /usr/lpp/psf
                        #           path to search for jobInit files
ps_max_memory = 96000K  #              maximum amount of memory that
                        #        PostScript uses; the 'K' is ignored
ps_job_timeout = 9999   #                  maximum number of minutes
                        #                  to process PostScript job
ps_server_timeout = 180 #                  maximum number of minutes
                        #                  server waits between jobs
ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
                        #                 list of font mapping files
ps_width = 8.5i         #                   width of generated image
ps_length = 11i         #                   length of generated image
ps_x_offset = 0i        #                      left and right margins
ps_y_offset = 0i        #                      top and bottom margins
ps_resolution = 600     #                resolution of target printer
ps_output_type = FS45   #                  type of output to generate
pragma = resources=none        ;\
         printerSpeed=130      ;\
         linkBandWidth=6000000 ;\
         resourceNameFilename=../.resNames;
```

If you are defining and configuring daemon RIPs across several servers, you must define the resources in the daemon RIP configuration file as shown in the highlighted section of Sample PDF parallel transform (8264.cfg) configuration file for color, p. 257. This step ensures that NFS can find all the resources and concatenate the result into a single file. You only need to specify the **resourceNameFilename=** value on the **pragma** parameter this way if you are using inline resources and the **FS45** IOCA color format.

These daemon configuration files can contain many of the same keywords as the transform configuration file, except `server`, plus some keywords that control the operation of the daemon. See *RICOH InfoPrint Manager: Reference* or man pages for the daemon keywords.

Sample PDF parallel transform (8264.cfg) configuration file for black and white

```
# 8264.cfg configuration file
# KEYWORD          EQUIVALENT ps2afp FLAG                      PURPOSE
#########################################################################
port =  8264            #                    which TCP/IP port to use
work_directory =  /var/psf/pdf2afp/4
                        #              directory to put work files in
log_file = /var/psf/pdf2afp/4/ps2afpd.log
                        #                 where to write log messages
notify = root           #          notify if problems are encountered
mail_command = /usr/bin/mail
                        #          program used to send notifications
ps_program = /usr/lpp/psf/bin/ps2afpi
                        # name of the PostScript interpreter program
ps_init_file = /usr/lpp/psf/ps2afp/ps2afpe.ps
                        #          name of the PostScript interpreter
                        #                          initialization file
ps_files_path = /usr/lpp/psf/ps2afp :\
                /usr/lpp/psf/config :\
                /usr/lpp/psf
                        #           path to search for jobInit files
ps_max_memory = 96000K  #              maximum amount of memory that
                        #        PostScript uses; the 'K' is ignored
ps_job_timeout = 9999   #                  maximum number of minutes
                        #                  to process PostScript job
ps_server_timeout = 180 #                  maximum number of minutes
                        #                  server waits between jobs
ps_font_map_files = /usr/lpp/psf/ps/psfonts.map :\
                    /var/psf/psfonts/user.map
                        #                 list of font mapping files
ps_width = 8.5i         #                   width of generated image
ps_length = 11i         #                   length of generated image
ps_x_offset = 0i        #                      left and right margins
```

```
ps_y_offset = 0i          #                    top and bottom margins
ps_resolution = 600       #               resolution of target printer
ps_output_type = IO1_G4   #          type of output to generate
```

Configuring for a dual engine halftone printer

If you have either an InfoPrint 4000 Advanced Function Duplex Printing System – Models ID5/ID6 or an InfoPrint 4000 High Resolution Duplex Printing System, you can use the **pdfprs** client to drive a dual-engine configuration with customized halftone support for each engine. The **pdfprs** client can RIP both the color and the black-and-white (monochrome) PDF files, these printers only print black-and-white output. . In addition, these printers do not provide as extensive a performance benefit as that you will see when printing color PDF files. If you do print color PDF files and then want to switch to black-and-white PDF files, you must stop the RIPs and change the configuration files.

For this support, you must configure and start an even number of RIPs. The first RIP must be customized to use the first print engine's halftone configuration, and the second RIP must be customized to use the second print engine's halftone configuration. This pattern must be repeated for each RIP that is configured.

In addition, each RIP must be configured to use the desired halftone configuration at the daemon configuration level. For example, if you are using two InfoPrint 4000 printers, all the even-numbered RIPs should match the halftone configuration specified on printer1, while all the odd-numbered pages should match the halftone configuration specified on printer2. Sample PDF parallel transform (8264.cfg) configuration file for black and white, p. 258 provides an example of a monochrome RIP daemon configuration file, with the variables changed from the Color daemon configuration file in **bold–face** type.

Hierarchy of transform options

InfoPrint assigns a hierarchy to its use of flags and configuration-file information when it runs the PDF transform program. The following lists the order in which InfoPrint uses flags and information specified in configuration files:

1. Any values you specify on the command line, including values in the configuration file you specify with the **-C** flag.

   InfoPrint commands process flags from left to right. If you enter the same flag more than once, InfoPrint uses the last occurrence of the flag to determine the value to use. For example, if you specify this command:

```
pdfprs -C8263.cfg -r240 -r300 myfile.ps
```

   The **pdfprs** command transforms the file using 300-pel resolution. InfoPrint ignores the resolution value specified in the configuration file `8263.cfg` and the first **-r240** flag and value.

2. Values specified in this default transform daemon configuration file: Default parallel PDF transform configuration files, p. 256.
3. Values specified in a customized configuration file identified with the **-C** flag of the **ps2afpd** command when the transform daemon started.
4. Values specified in this default transform daemon configuration file: **/usr/lpp/psf/ps2afp/ps2afpd.cfg**.
5. Default values that are built into InfoPrint. These are the same as the transform defaults, except that the default PDF output type is an IM1 uncompressed image.

Troubleshooting the parallel PDF RIP client

There are certain factors that can enhance the performance of your **pdfprs** client. This section provides some recommendations.

Defining RIPs on an InfoPrint server

The more RIPs you define, the greater the performance improvement. However, because a single RIP uses an entire CPU, you should never start more RIPs than your InfoPrint Server can support. The **pdfprs** client generally uses one processor, so be sure that your configuration can support the number of RIPs that you define and start. For guidelines see Processors required to use the Parallel PDF RIP client, p. 254. For more information about defining RIPs across servers, see RIP Daemon configuration files, p. 257.

Specifying an uneven number of RIPs

If you have PDF files that have simple text data on the front page and more complex image data on the back page consistently throughout the job, a two-RIP configuration will not result in enhanced performance. Since the second RIP always gets the more complex pages, the job will take the same time to RIP as if a single RIP were used, while the first RIP remains idle. Use an uneven number of RIPs to distribute the pages in the job more evenly.

Using –p or –g options from the command line

While you can specify a page subset for a particular job by using the **–p** or the **–g** flags on the **pdfprs** command, we recommend that you avoid these flags. If InfoPrint Manager has to determine a subset of the pages in the job, it slows down processing and reduces the benefit of having parallel RIPs.

Note

- Remove the `/var/psf/ps2afp/PSscreen` and `/var/psf/ps2afp2/PSscreen` folders during product upgrade.

Working with the XML transforms

XML provides a standard method for the exchange and processing of data across different computing platforms. XML data is structured according to standards from the World Wide Web Consortium (W3C). Specifically designed for web applications, XML differs from HTML in that, while HTML contains information about how data is formatted for presentation on webpages, XML contains data with no presentation information. A W3C standard stylesheet language, called Extensible Stylesheet Language (XSL), uses formatting objects to describe how XML data is presented. XML data containing XSL formatting objects is called XSL-FO. XSL-FO describes document presentation details, such as pagination, layout, and styling information.

Using the XML transforms with InfoPrint Manager

XML has become the key standard in the deployment of new technologies for web services, data interchange, messaging, and application integration. Support for XML is found in several printing products, including XML Extender and Print Services Facility (PSF) for z/OS.

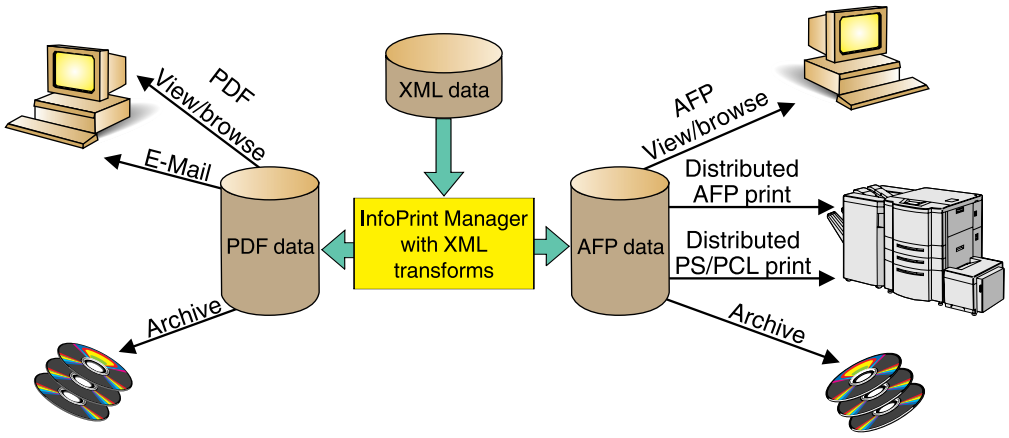
The XML transforms are the preferred method for printing highly-dynamic XML data on AFP printers because the transforms can efficiently process data that changes from job to job. The XML transforms are also an efficient, cost-effective alternative to paper-based communication because it can transform XML data to PDF documents for electronic delivery to customers.

Other benefits of using **xml2afp** and **xml2pdf** to transform XML data to AFP or PDF include:

- You can use your existing AFP output system to print or view your XML data without worrying about compatibility issues.
- The XML transforms provide you with significant formatting flexibility for your XML documents.
- Print delivery is guaranteed when you transform your XML data to AFP and then print to an AFP printer.
- Once XML data is transformed, you can use different tools to:
  - Print to an AFP printer or transform AFP data to PostScript (PS) or printer control language (PCL) for printing on a PS or PCL printer.
  - Send PDF data to an electronic destination.
  - View or archive AFP or PDF data.

The XML transforms let you take advantage of XML for data interchange and content management while using AFP's print and presentation technology and PDF's electronic document exchange. The following figure shows what you can do with XML data when you use the XML transforms.

Scenarios for using XML data that has been transformed to AFP or PDF



The Ricoh products you can use with the AFP or PDF output include:

AFP view/browse

AFP Viewer plug-in or Content Manager OnDemand

PDF view/browse

Content Manager OnDemand

Distributed AFP print

InfoPrint Manager and PSF for z/OS

Distributed PS/PCL print

InfoPrint Manager, InfoPrint Server, and InfoPrint Server Transforms

E-mail

IP PrintWay component of InfoPrint Server

Archive

Content Manager OnDemand

How do the XML transforms work?

You use the **xml2afp** and **xml2pdf** to transform XML data to an AFP or a PDF data stream. To do this, you must supply one of these to the XML transforms::

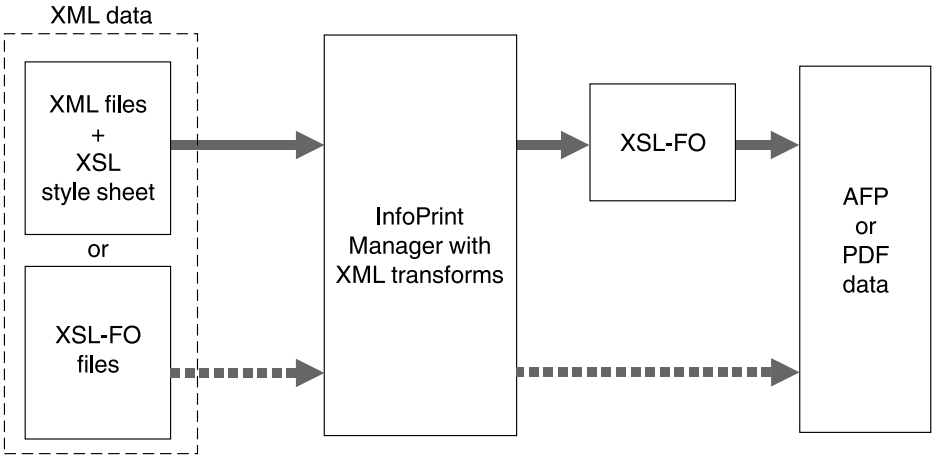
- XML files (or data sets), which do not contain data placement and presentation information, and an accompanying XSL style sheet
- XSL-FO files (or data sets), which contain XML data with XSL formatting objects

To transform XML data to AFP or PDF, the XML transforms use the data you supply and do one of these:

- Uses the XSL style sheet to transform the XML files into XSL-FO and then completes the transformation to AFP or PDF
- Directly transforms your supplied XSL-FO files to AFP or PDF

The following figure shows how the XML transforms process data from XML to AFP or PDF data.

Process for transforming XML data to AFP or PDF



Running the XML transforms

You can run the XML transforms as:

- A standalone transform
- A transform managed by InfoPrint Manager

Capabilities and limitations

The XML transforms have these capabilities and limitations, depending on how data is being transformed:

- Capabilities and limitations for transforming all data
- Capabilities and limitations for transforming data to AFP
- Capabilities for transforming data to PDF

General capabilities and limitations for transforming all data

Output document size

Output document size is dependent on the amount of memory available; however, no checking is done between document size and available memory. If the output document is too big for the available memory, you receive an out of memory error and the transform stops processing.

XSL support

The transforms support most, but not all, formatting objects, properties, and features of the XSL Version 1.0 standards as documented by the W3C (Extensible Stylesheet Language).

Style sheets embedded and referenced in XML files are not supported.

Capabilities and limitations for transforming data to AFP

Bar code support

The transform does not support bar codes.

Font support

The first font in the font index with properties that most closely match the desired characteristics is used. Condensed and expanded fonts are usually only used when the font-stretch property is selected. Italic and oblique fonts are considered the same and are used interchangeably.

The XML transforms support converting double-byte character set (DBCS) fonts from one code page to another.

The XML transforms work with the OpenType fonts from the command line. For more information about OpenType Fonts, see *Using OpenType Fonts in an AFP System*, G544-5876.

Graphic support

You can specify graphic data in one of these AFP resources:

Overlay

An overlay contains a collection of constant image data (such as lines, shading, boxes, or logos) that can be merged with variable data during printing. The AFP output file references the name of an overlay, which is located in a resource directory. Because the overlay is not included but is only referenced in the output file, you must specify the image size in the XSL external-graphic formatting object. For example:

```
<fo:external-graphic src="overlay:corplogo" content-height="4in"
content-width="2in"/>
```

Page segment

A page segment contains image data (such as bar codes, signatures, logos, or graphics in image format) that can be merged with variable data during printing. The AFP output file references the name of a page segment, which is located in a resource directory. Because the page segment is not included but is only referenced in the output file, you must specify the image size in the XSL external-graphic formatting object. For example:

```
<fo:external-graphic src="pageseg:corpsign" content-height="4in"
content-width="2in"/>
```

IOCA object

An Image Object Content Architecture (IOCA) object contains a series of picture elements (pels) arranged in rows and columns, and specifies where the image is placed on a page. The XML transforms access the first IOCA object in a specified MO:DCA-P file and place it inline in the AFP output file (only MO:DCA-P image data is supported). The transforms obtain the actual size from the image; therefore, you do not need to specify the size in the XSL external-graphic formatting object. If you do specify the size with the external-graphic formatting object, the transforms crop the image to make it the specified size or add whitespace to the bottom-right corner, so the image appears in the upper-left portion of the allotted space. Here is an example of the formatting object you specify for an IOCA object:

```
<fo:external-graphic src="myfile/image.afp"/>
```

MO:DCA-P output

Output from the transform conforms to the MO:DCA Interchange Set 1 (IS/1) architecture (see *Mixed Object Document Content Architecture Reference*, SC31-6802).

Capabilities for transforming data to PDF

PDF output

The transform converts XML data to PDF 1.1 data streams.

Customization tasks for the XML transforms

The system administrator can customize the XML transforms as standalone transforms or as transforms managed by InfoPrint Manager. If you have InfoPrint Manager installed, InfoPrint recommends you use it to manage the XML transforms.

This section details the common, standalone, and InfoPrint Manager customization tasks the administrator must do before the job submitter can run the XML transforms. The administrator does the common customization tasks and either the standalone or InfoPrint Manager customization tasks, depending on whether the XML transforms are run as a standalone transform or managed by InfoPrint Manager.

Common customization tasks

The common customization tasks you do depend on whether you are using the XML to AFP transform or the XML to PDF transform.

Tasks for using the XML to AFP transform

The sample **xml2afp** configuration file is provided in `/usr/lpp/psf/xml2afp/xml2afp.cfg`. These items are set in the file and can be modified:

1. The default location of the AFP fonts.
2. The default paper size.

Additionally, defaults for all the **xml2afp** options can be set in the configuration file.

The syntax of the configuration file is exactly the same as the command line syntax with these additional rules:

1. Blank lines are ignored.
2. All data on a line after a pound sign is ignored. The pound sign can occur anywhere on a line.
3. To include a pound sign in parameter data, it must be escaped; that is, it must be preceded by a backwards slash.
4. White space before and after parameter data is removed.
5. Parameter data on successive lines is concatenated with a single space between them.
6. Parameters can span lines as long as they follow rules 1 through 5.

Customizing the xml2afp.cfg file

You can either create a new configuration file or modify the sample that is provided:

1. Create a new **xml2afp** configuration file, **xml2afp.cfg**, which lets you configure the **xmp2afp** transform.

   To create the file on AIX, copy the sample configuration file, `/usr/lpp/xml2afp/samples/xml2afp.cfg` to `/etc/xml2afp.cfg`:

```
cp /usr/lpp/xml2afp/samples/xml2afp.cfg /etc/xml2afp.cfg
```

   Note

   If you want to modify the sample configuration file, you still need to copy it.

2. Edit the file using your preferred editor, such as **vi** or **notepad**.

   1) On the FONTLIB keyword, specify the location of the AFP font libraries to be used by the **xml2afp** transform. For example,

```
FONTLIB="//SYS1.FONTOLN://SYS1.FONT300"
```

      You can use MVS partitioned data set names, which are fully qualified and prefixed with two slashes. Use colons to separate entries in the list.

   2) If necessary, customize these items:

      - Font index files

        If you want to customize the way the **xml2afp** transform maps character data from your documents to AFP fonts, create or modify a font index file and then specify, under the XML2AFP_CUSTOM keyword, the directory where your font index file is located. Font index files are located in the same directory as the device types file (for example, **/etc/xml2afp**).

        You can create your own font index file or use the ones supplied with the **xml2afp** transform.. The font index files describe the file syntax and include one file that specifies outline fonts (`/usr/lpp/psf/xmltransform/outline_font_index` and one file that specifies raster fonts (`/usr/lpp/psf/xmltrasnform//raster_font_index`. See either of those font index files for information to help you create a font index file.

      - XML Toolkit location with the TOOLKIT and CLASSPATH keywords

- Java executable location with the JAVA keyword
- The **xml2afp** transform location with the XML2AFP_BASE keyword, if it is installed in a location other than **/usr/lpp/psf/xml2afp**.
- Memory the Java virtual machine uses to run the transform with the HEAP_SIZE keyword

Important

The configuration file is a shell script. You can use any commands that are valid according to **/bin/sh**. However, because the script runs for each invocation of **xml2afp** transform when it is not managed by InfoPrint Manager, avoid using any commands that take a long time to complete.

See "xml2afp Command: transforms XML data to AFP" in the "InfoPrint Manager Tools" chapter in RICOH InfoPrint Manager: Reference for the sample XML transforms configuration file, `/usr/lpp/psf/xml2afp/samples/xml2afp.cfg`, that is shipped with the **xml2afp** transform.

Customizing the font index file for xml2afp

The sample font index file for **xml2afp** is provided in `/usr/lpp/psf/xmltransform`. The samples include a `outline_font_index` and a `raster_font_index` that you use to map XML font names to both outline FOCA fonts and raster fonts in the font path.

Note

If you are using OpenType fonts, the font index files are not necessary.

A font is referenced in XSL-FLO using these attributes:

family
size
style
weight

Font index files describe AFP fonts using these attributes. In addition, a map is defined that maps unicode code points into AFP code points.

The font index file consists of a sequence of key and value pairs, separated by whitespace (such as a space, tab, or a newline). Comments can be inserted anywhere between the delimiters /* and */. You can also use // to indicate that the remainder of the current line is a comment.

The keys and their meanings are:

map

Select the named character map, creating it if it does not exist. A character map is used to map unicode code points to AFP code page names and code points. The default is `default`.

codepage

Set the code page to use for subsequent character mappings.

to

End a range of mapped unicode code points. The value must be an integer representing a valid unicode code point, and it must be greater than the beginning value of the range given by the closest preceding value for the **char** keyword. The value must be a valid unicode code point.

at

Creates a mapping using the closed range given by a preceding **char** value and an optional **to** value. The unicode characters from begin to end are mapped to the AFP code point range from the **at** value to the **at** value + end - begin, inclusive, using the closest preceding **codepage** value. This value must be an integer from 0 to 65535.

family

Establish the current font family name. The value must be a string containing no whitespace characters.

size

A font size, expressed as a floating point value giving the vertical size of the font, in points (1/72nds inches). Size must be omitted when defining outline technology fonts, since they can take any size.

style

A font style. The value must be one of these keywords:

normal
italic
oblique
backslant

The default is **normal**.

weight

A font weight. The value can be one of these keywords:

normal
bold

The value can also be one of the values between 100 and 900 inclusive, in increments of 100. The keyword **normal** is equivalent to 400, and the keyword **bold** is equivalent to 700. The default value is **normal** (400).

charset

Define a font with the preceding values given for the font properties (family, size, style, and weight), or default values if no preceding property values have been established. The currently selected character map is associated with the font, and it is used for mapping unicode code points when the font is selected. If modifications are made to the map later in the file, those changes do not affect the defined font.

alias

Establish a font family alias for the current family.

Of the keywords listed, only two are active: **at** and **charset**. The other keywords collect values used when an active keyword is found. This allows for a variety of possible arrangements for a font index. For example:

```
family courier
alias monospace
size 7
weight normal
style normal
charset C04200070
style italic
charset C0430070
```

```
weight bold
style normal
charset C0440070
style italic
charset C0450070
```

Defines the same set of fonts as:

```
family courier size 7 weight normal style normal charset C0420070
family courier size 7 weight normal style italic charset C0430070
family courier size 7 weight normal style italic charset C0440070
family courier size 7 weight normal style italic charset C0450070
alias monospace
```

But might be considered more difficult to read.

Tasks for using the XML to PDF transform

The sample **xml2pdf** configuration file is provided in `/usr/lpp/psf/xml2pdf/xml2pdf.cfg`. The default paper size can be modified. Additionally, defaults for all the **xml2pdf** options can be set in the configuration file.

The syntax of the configuration file is exactly the same as the command line syntax with these additional rules:

1. Blank lines are ignored.
2. All data on a line after a pound sign is ignored. The pound sign can occur anywhere on a line.
3. To include a pound sign in parameter data, it must be escaped; that is, it must be preceded by a backwards slash.
4. White space before and after parameter data is removed.
5. Parameter data on successive lines is concatenated with a single space between them.
6. Parameters can span lines as long as they follow rules 1 through 5.

These are the common tasks you must do when using the XML to PDF transform:

1. Create the XML Extender configuration file, **xml2pdf.cfg**, which lets you configure XML Extender. To create the file:
   - Copy the sample configuration file, **/usr/lpp/xml2afp/samples/xml2pdf.cfg**, to **/etc/xml2pdf.cfg**:

```
cp /usr/lpp/xml2afp/samples/xml2pdf.cfg /etc/xml2pdf.cfg
```

   Note

   If you want to modify the sample configuration file, you still need to copy it.

2. Edit the file using your preferred editor.
   If necessary, customize these items:
   - Java executable location with the JAVA keyword
   - The **xml2pdf** transform location with the XML2PDF_BASE keyword, if it is installed in a location other than **install_path\xml2pdf**
   - Memory the Java virtual machine uses to run the transform with the HEAP_SIZE keyword

Important

The configuration file is a shell script. You can use any commands that are valid according to **/bin/sh**. However, because the script runs for each invocation of the **xml2pdf** transform when it is not managed by InfoPrint Manager, avoid using any commands that take a long time to complete.

See "xml2pdf Command: transforms XML data to PDF" in the "InfoPrint Manager Tools" chapter in *RICOH InfoPrint Manager: Reference* for the sample XML transforms configuration file that is shipped with the **xml2pdf** transform.

Color management resource transform support

InfoPrint Manager enables these transforms to do color management:

- ps2afp
- jpeg2afp
- tiff2afp
- gif2afp

Note

The PCL and SAP transform do not support color management.

Transforms capable of color management allows the specification of color resource information at job submission time and the implementation of color management at the transform level within InfoPrint Manager. This support will provide color fidelity with better printer performance.

Submitting job using transforms for color management

Before you use transforms to do color management, make sure that you have installed the ICC Profiles using the Resource Installer into the Resource library as Color Management Resources (CMRs). For more information, see Color Management Resources, p. 311 and RICOH AFP Resource Installer, p. 323.

Note

- You can specify the ICC Profile directly in a pure PS/PDF transform.
- Only audit color conversion, instruction color conversion, and link color conversion CMRs are supported.

To submit a job:

1. Select your destination printer.
2. Right-click and select **Job and Document Defaults**.
3. Click **Modify**.
4. Click **Document AFP Color**. If this tab is not visible, click **Show more** and then select **Document AFP Color**.
5. Enter one or more input color management resources in **Input color management resources**.

6. Next, enter an output color management resource in **Output color management resource**.
7. From **Color rendering indent**, select one option.
8. Then, select the output color resource tagging from **Tag output CMR**.
9. In the **Place CMR inline** option, click **Yes** to inline the color resource information in the FS45. Default is **No**.
10. Click **Submit**.

For more information about the values that you can specify, see *RICOH InfoPrint Manager: Reference*.

Understanding the transform objects and the custom steps (transforms) subsystem

Transforms are InfoPrint Manager objects (like queues and destinations) that you create and configure in InfoPrint Manager. Transforms can be viewed as custom steps that receive data in a particular format or formats, process the data in some way, and output the changed data. When you create a transform, you have complete control over what data format or formats it operates on and what type of processing it does. The processing that the transform performs can be anything from simply copying the data to a different place on disk, to filtering the data stream that the transform receives (for example, to remove a particular PostScript order), to transforming the data to a completely different format (for example, converting PCL to AFP).

When you create a transform, you specify several things, including:

- **Input data format**

  The data stream that is sent to the transform. The input data format is what the transform uses to determine whether it has any work to do. If you send a PostScript job through a transform that has its input data stream set only to **PCL**, the job will bypass the transform without being changed at all. You can set the transform to accept multiple input data formats.

- **Transform options**

  The program or command that actually does the work of the transform; what happens during the *processing* stage. The transform options can be system commands (such as **copy**), programs provided with InfoPrint Manager (such as **ps2afp**), or your own custom transform programs (for example, **mypclfixup**).

  When you create a transform, you specify the complete command line needed to invoke this program or command. In this command line, you can also specify various *substitution variables*, place holders which InfoPrint Manager replaces with a value when InfoPrint Manager runs the transform. For example, if you include `%i` in the line, InfoPrint Manager will replace it with the name of the file that the transform receives; if you include `%o`, it will be replaced with the name of the file that the transform needs to put the changed data into.

- **Output data format**

  The data stream that results from the transform. You can only specify one output data format.

In addition, you can specify whether you want the file that the transform produces to be printed. If the transform does not return the file to be printed (for example, if the transform completes its processing and saves the file to a different location), it is called a *terminating transform*. In addition to the *terminating transform*, you can also create a *conditionally-terminating transform*. To create this, define a unique return code to indicate the transform was successful but does not return data to print.

Transforms do not do anything by themselves. They remain inactive until you associate them with an actual destination as a transform sequence. A *transform sequence* is an attribute of an actual destination that contains an ordered list of transforms. Every job that is submitted to that actual destination is sent through that list of transforms in the order specified before it is printed.

InfoPrint Manager lets you create a sequence for transforming print data before the system sends it to a printer. This provides users with a feature similar to the configurable transform sequences that you could create in PSF/2 using the **XFMFLTR** utility. For more information about creating transform sequences, see Defining a custom step (transform) sequence, p. 285.

Note

1. If you include a terminating transform in a transform sequence, it must be the last transform in the sequence.
2. If you invoke a terminating transform for a document in a job, the entire job ends after all the documents in that job finish processing. It does not matter whether any other documents in the job use the terminating transform.

Configuring custom steps (transforms)

To configure a custom step (transform), you must do these two tasks:

1. Creating custom steps (transforms), p. 271.
2. Associating a custom step (transform) with an actual destination, p. 275.

Creating custom steps (transforms)

You can create a transform sequence through the **InfoPrint Manager Web Administration Interface**.

Note

Due to terminology changes, the equivalent term used for **Configurable Transforms** in the InfoPrint Manager Web Administration Interface is **Custom Steps**. For more information about Custom Steps, go to the InfoPrint Manager Web Administration Interface help system.

To create a transform, an InfoPrint Manager system administrator must use this procedure:

1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click , the Menu icon, and select **Custom Steps (Transforms)...**.
4. Click the **New** button in the upper right corner.
5. In the **Create New Custom Step (Transform)** dialog, fill in these values:

   Name

   Type a meaningful name for the transform.

InfoPrint Manager references the transform name in transform sequences that are associated with actual destinations.

Server

If not filled in, either type or select the InfoPrint Manager server in which you want to create the transform.

Transforms can only be associated with actual destinations in the same server. If the InfoPrint Manager Web Administration Interface is only monitoring one server, this field contains that server name.

Output File Format

Select a format for the output file produced from this transform.

Description

Type a description of this transform that will appear whenever you show this transform through the InfoPrint Manager Web Administration Interface.

Transform Options

Type the command-line transform and its options, including any InfoPrint Manager substitution variables.

Final Step

Specify if the transform step is the last one.

Force Sequence Stop Return Code

Specify a return code that indicates the transform step was successful, but the transform step is not returning data for further processing by InfoPrint Manager.

Document Formats Allowed

Add a format for the input file or files that are being transformed.

Note

You can specify either one or multiple document formats. The format of the input file determines whether this transform is called for a job sent to this actual destination.

Use Server Locale

Specify whether to use the locale language of the server (**On** - default) or English (**Off**) for input and output of information defined in the **%%** substitution variables in the **Transform Options** field.

The transform examples that follow demonstrate common uses of transform objects, using the substitution values shown in the “Transform substitution variables” table. Some of these examples presume that an application programmer has created a program with the name specified at the start of the **Transform Options** field.

Note

1. If you are specifying any substitutions for which the passed values might contain a blank space, such as `job name`, be sure to put quotes around the substitution so it is passed correctly. Note that this only applies to single `%` substitution variables. Do not put quotes around any `%%` substitution variables.
2. The examples that follow presume that the **PATH** environment variable has been set to the appropriate path so you do not have to specify the fully qualified path name in the **Transform Options** field.

Transform substitution variables

Transform options substitution variables	Meaning
%i	The name of the input file to transform.
%o	The name of the file in which to store the output of the transform. Note If this is a non-terminating transform, data must be stored in the file named by %o.
%e	The name of the file in which to store any transform information or error messages. If anything is written to this file, it is logged in the InfoPrint Manager server log. Note 1. If the transform exits with an exit code of 0 or the value set in the **Force Sequence Stop Return Code** field, the error messages from the transform are logged as **debug** severity. 2. If the transform exits with a non-zero return code, but not with the value set in the **Force Sequence Stop Return Code** field transform attribute, the error messages from the transform are logged as **error** severity. 3. InfoPrint Manager writes transform error messages to either the file specified through this substitution variable, or to standard error (**stderr**). You should not mix output to **%e** and **stderr** in a single run of the exit program. You should use either one or the other in any given run.
%u	The name of the file for updating the **pages-completed** and **job-page-count** job attributes.

Transform options substitution variables	Meaning
	Note This file should only be accessed through the **transform_update** utility.
%j	The name of the original print file, minus the path (from the **document-file-name** attribute). Note This should only be used to construct another name, such as a temporary file.
%n	The name of the original print file, minus the path and minus the extension (from the **document-file-name** attribute). Note This should only be used to construct another name, such as a temporary file.
%d	A text string that represents the document format of the input file. This value might be: • **ASCII** for ASCII line data. • **AFPDS** for Advanced Function Presentation data • **PCL** for Printer Control Language data • **PS** for PostScript data Note You can also acquire document format through the **%%Ddocument-format** substitution value.
%p	The value of the document's **destination-pass-through** attribute. Note This value is passed to all subsequent transforms and the actual destination.
%q	The actual destination that is processing the job.
%s	The **document-sequence-number** for the document.
%t	The value of the document's **other-transform-options** attribute. Note To write one transform, but use it differently for different actual destinations or jobs, you can pass values into the transform with this parameter.

Transform options substitution variables	Meaning
%g	The global job identifier for the job.
%#	The job-identifier (10 digits maximum).
%%A*ad_attribute***%%**	The value of the InfoPrint Manager actual destination attribute specified. For example, if %% `Aresource-context%%`, InfoPrint Manager passes the actual destination's **resource-context** attribute value. **Note** See the "Considerations when using the %% *Nattributename* substitution control sequences" topic under the "Usage guidelines" for the **other-transform-options** attribute for transforms in the *RICOH InfoPrint Manager: Reference* before using this substitution variable.
%%D*doc_attribute***%%**	The value of the InfoPrint Manager document attribute specified. For example, if you use %% `Ddepartment-text%%`, InfoPrint Manager passes the document's **department-text** attribute value. **Note** See the "Considerations when using the %% *Nattributename* substitution control sequences" topic under the "Usage guidelines" for the **other-transform-options** attribute for transforms in the *RICOH InfoPrint Manager: Reference* before using this substitution variable.
%%J*job_attribute***%%**	The value of the InfoPrint Manager job attribute specified. For example, if you use `%%Jjob-owner` %%, InfoPrint Manager passes the **job-owner** attribute value of the job that owns the document being transformed. **Note** See the "Considerations when using the %% *Nattributename* substitution control sequences" topic under the "Usage guidelines" for the **other-transform-options** attribute for transforms in the *RICOH InfoPrint Manager: Reference* before using this substitution variable.

Associating a custom step (transform) with an actual destination

Now that you have created a transform object, you must associate it with an actual destination before you can use it.

To associate a data stream transform with an actual destination from the InfoPrint Manager Web Administration Interface, use this procedure:

1. In the left-hand **Objects** panel, click the **Printers** tab and select the printer that you want to send the job to.
2. Click , the Menu icon, and select the **Properties** option.
3. In the **Properties for...** dialog, click **Configuration**.

   The data stream transforms that this printer can use are listed in the **Steps to Use** field.
4. Select the transforms that you want to use (the equivalent of setting the **transforms-sequence** actual destination attribute from the command line).

   Note

   You can string multiple transforms together to do a sequence of tasks. For more information about this feature, see Defining a custom step (transform) sequence, p. 285.
5. Click **OK** to associate this transform sequence with the selected printer.

If you prefer to use the command line, you can use these InfoPrint Manager commands from either an AIX or Linux command line or an MS/DOS window with **transform** objects:

- **pdcreate**
- **pddelete**
- **pdset**
- **pdls**

For more information about the attributes in creating transform sequences or custom steps, see the "Attributes for Transforms" topic in the "InfoPrint Object Attributes" chapter of the *RICOH InfoPrint Manager: Reference*.

Examples of creating custom steps (transform) objects

The following shows four specific examples (from simple to more advanced) for creating transform objects. The third and fourth examples require additional programming to create the programs that are called from the **Transform Options** field. This topic should suggest how you can use custom steps to customize your InfoPrint Manager installation.

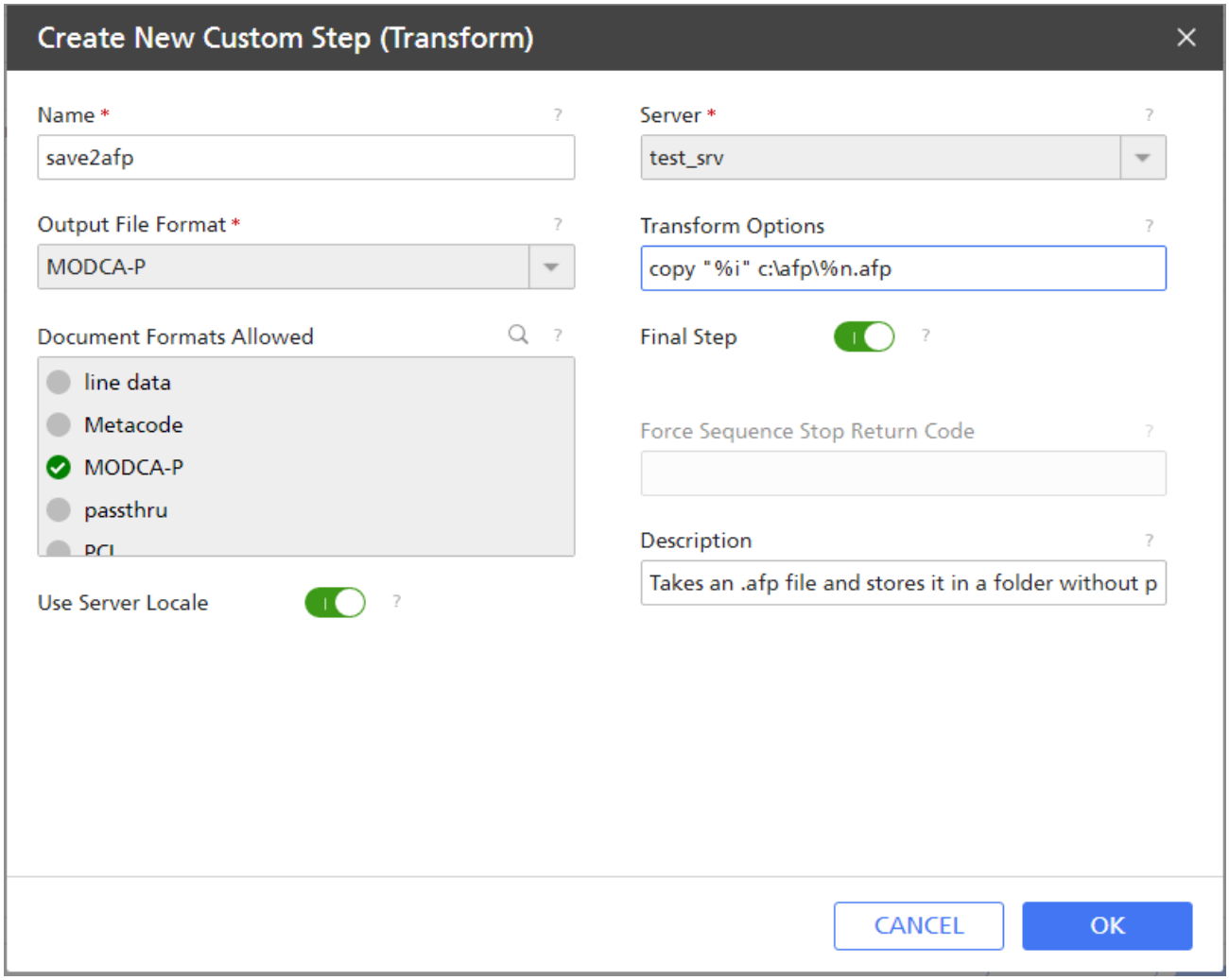
Creating a custom step (transform) object to copy to a file without printing

To define a data stream transform object from the InfoPrint Manager Web Administration Interface that takes an AFP input file and copies it to a file location on the hard drive without printing the output, use this procedure:

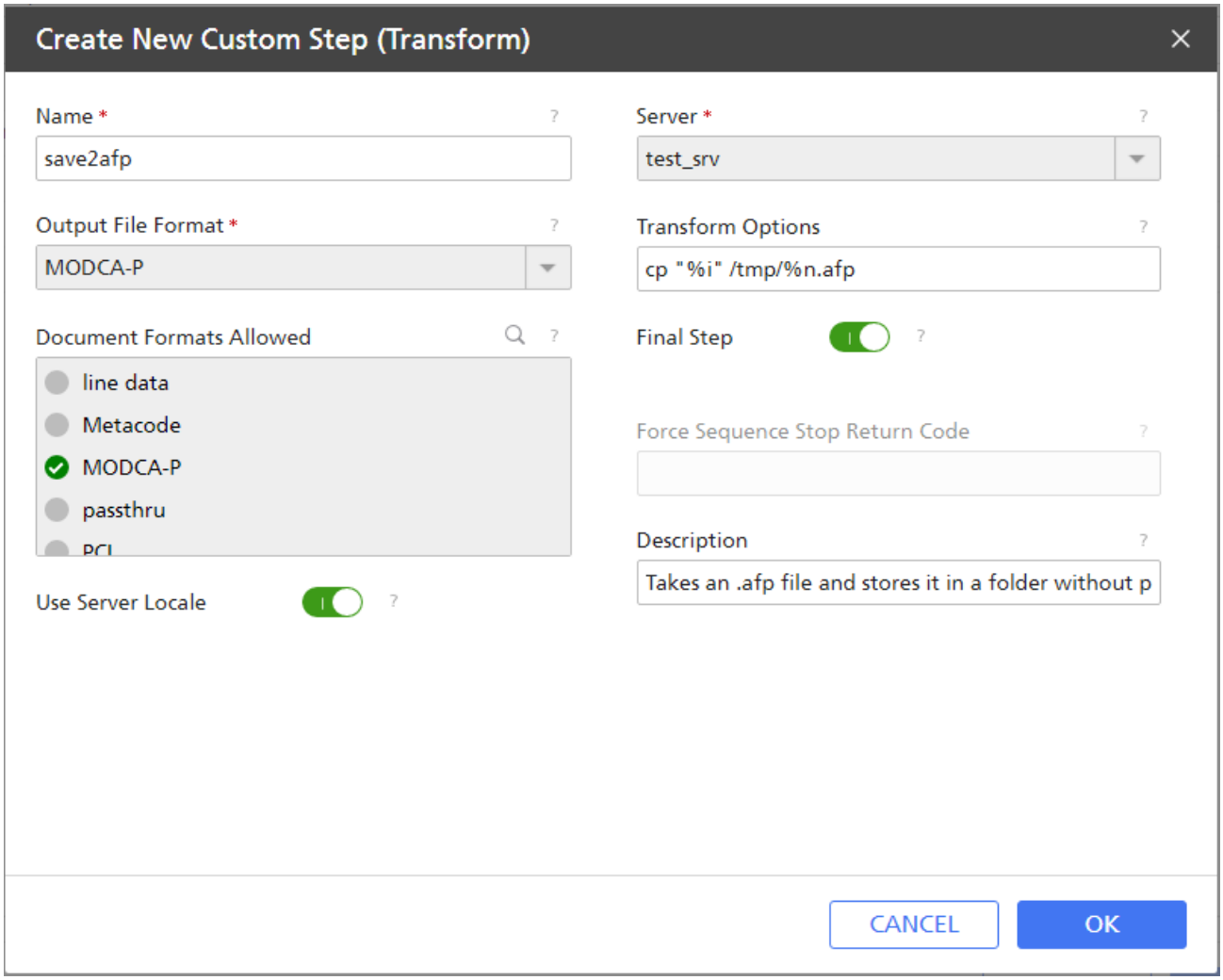
1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click , the Menu icon, and select **Custom Steps (Transforms)...**.

4. Click the **New** button in the upper right corner.
5. In the **Create New Custom Step (Transform)** dialog, fill in the fields as shown in the following figures, depending upon your operating system. These values create a transform that copies AFP input data to an **\afp** folder on a specific drive on Windows or a **/usr/samp** file system on AIX and Linux. Also, it names the transformed file with the name of the original file name because of the **%n** substitution variable.
6. Since you do not want the job to be printed, click **Final Step** to indicate that this transform is a terminating transform.

Create New Custom Step (Transform) dialog: Define a transform that copies an AFP file to a folder without printing on Windows



Create New Custom Step (Transform) dialog: Define a transform that copies an AFP file to a folder without printing on AIX and Linux



7. Click **OK**.

   InfoPrint Manager creates a transform object under the name you provided in the **Name** field. To see the values of this transform, click ⋮, the Menu icon, on the **Servers** tab and select the **Custom Steps (Transforms)...** option.

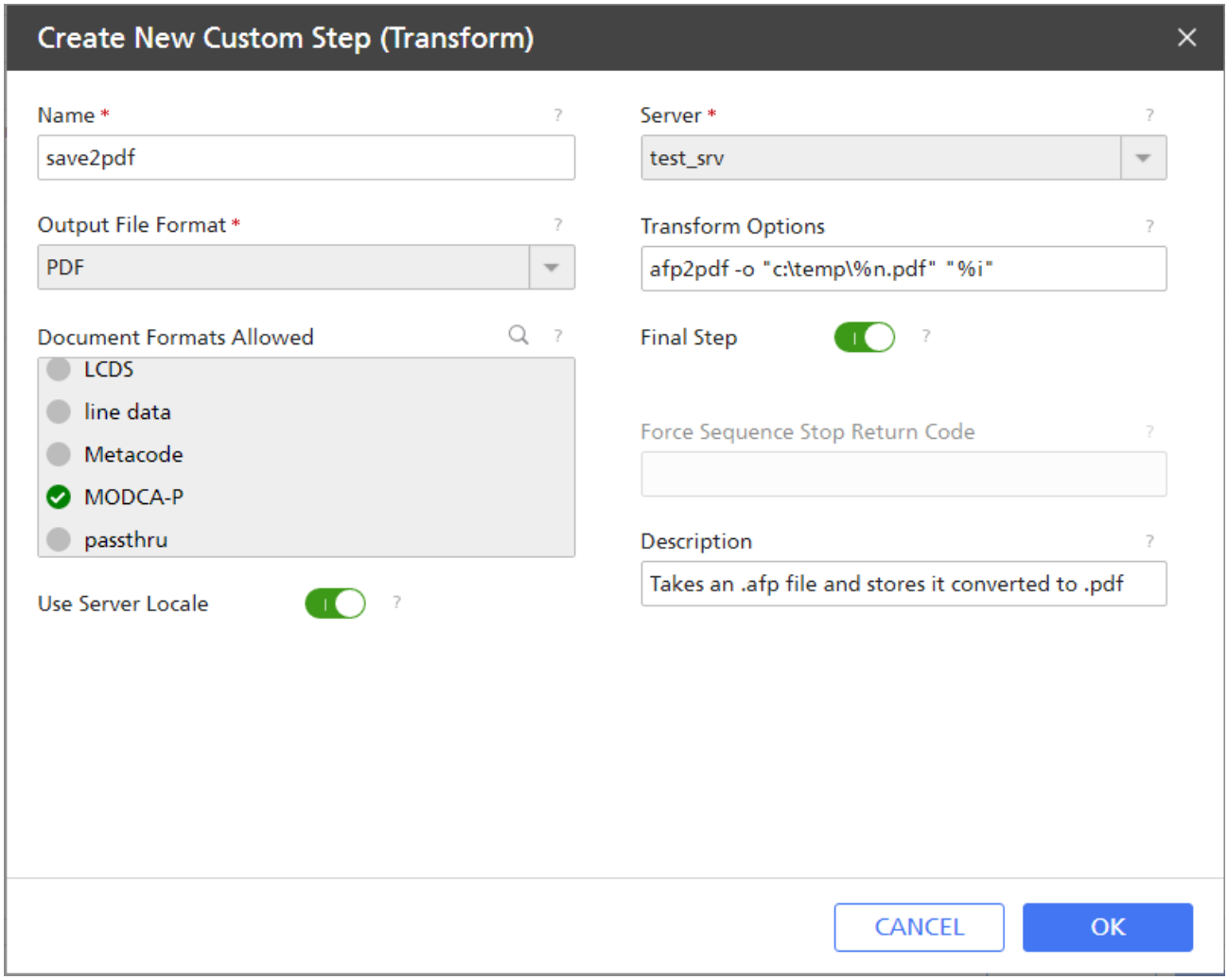
Creating a custom step (transform) object to convert AFP data into PDF data and store it for later usage

To define a data stream transform object from the InfoPrint Manager Web Administration Interface that takes an AFP input file and converts it into PDF data and stores it in a location on the hard drive without printing the output, use this procedure:

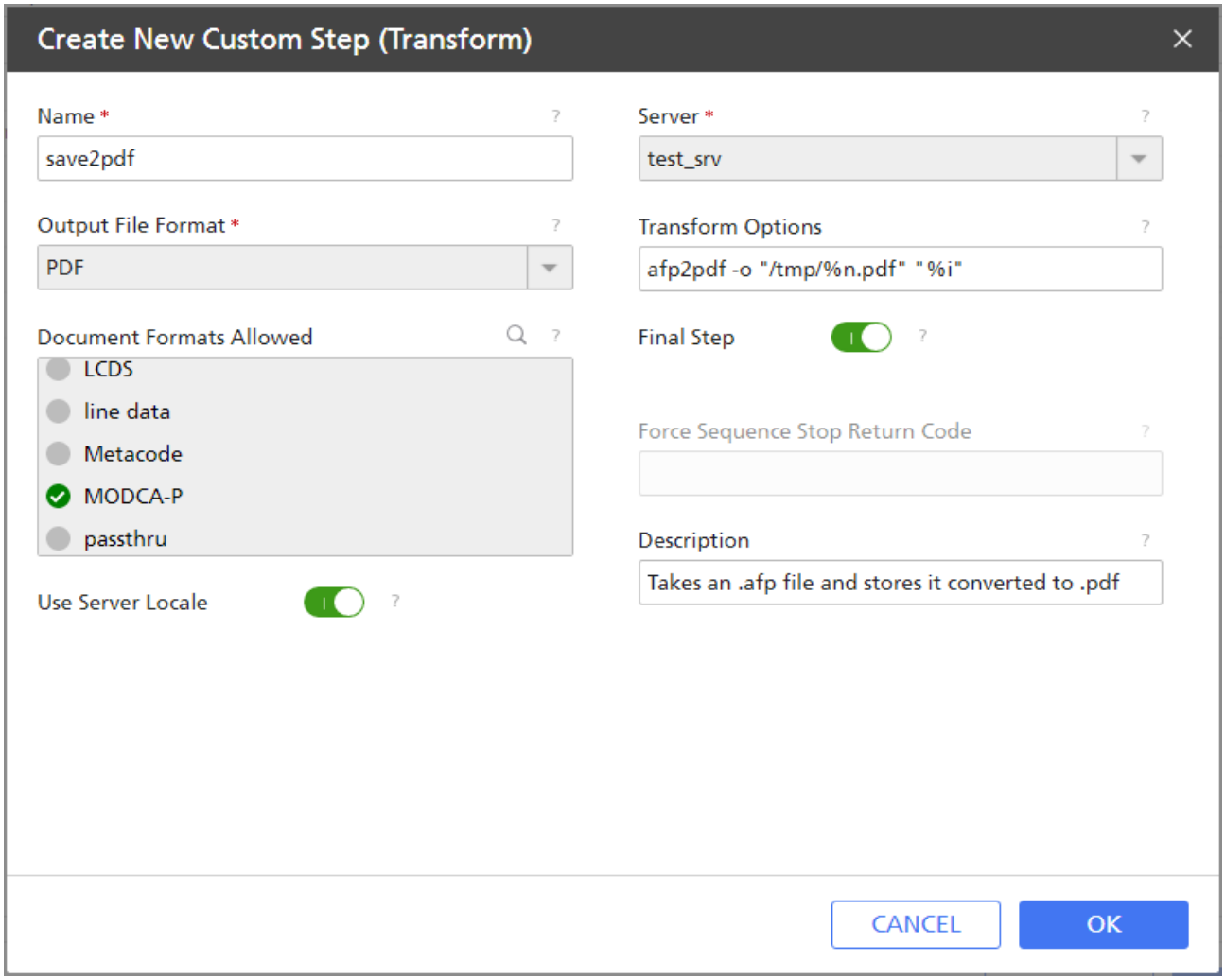
1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click ⋮, the Menu icon, and select **Custom Steps (Transforms)...**.

4. Click the **New** button in the upper right corner.
5. In the **Create New Custom Step (Transform)** dialog, fill in the fields as shown in the following figures, depending upon your operating system. These values create a transform that converts AFP input data into PDF data and saves the output in a file indicated by the afp2pdf **-o** flag value. Also, it names the transformed file with the name of the original file name because of the **%n** substitution variable.
6. Since you do not want the job to be printed, click **Final Step** to indicate that this transform is a terminating transform.

Create New Custom Step (Transform) dialog: Define a transform that converts AFP data into PDF data and store it for later usage on Windows



Create New Custom Step (Transform) dialog: Define a transform that converts AFP data into PDF data and store it for later usage on AIX and Linux



7. Click **OK**.

   InfoPrint Manager creates a transform object under the name you provided in the **Name** field. To see the values of this transform, click ⋮, the Menu icon, on the **Servers** tab and select the **Custom Steps (Transforms)...** option.

Note

- See *RICOH InfoPrint Manager AFP2PDF Transform Feature: Installing and Using* for information about installing and using the afp2pdf transform.

Creating a custom step (transform) object to convert data and print only smaller files

If your transform might want to terminate the processing of the job in some cases but return data to be printed in other cases, you can create a conditionally terminating transform. For example, you might want to transform and store some documents but transform and return for printing other documents. You

can define a return code that your transform can return to indicate when it has completed its task successfully, but is not returning any data for printing.

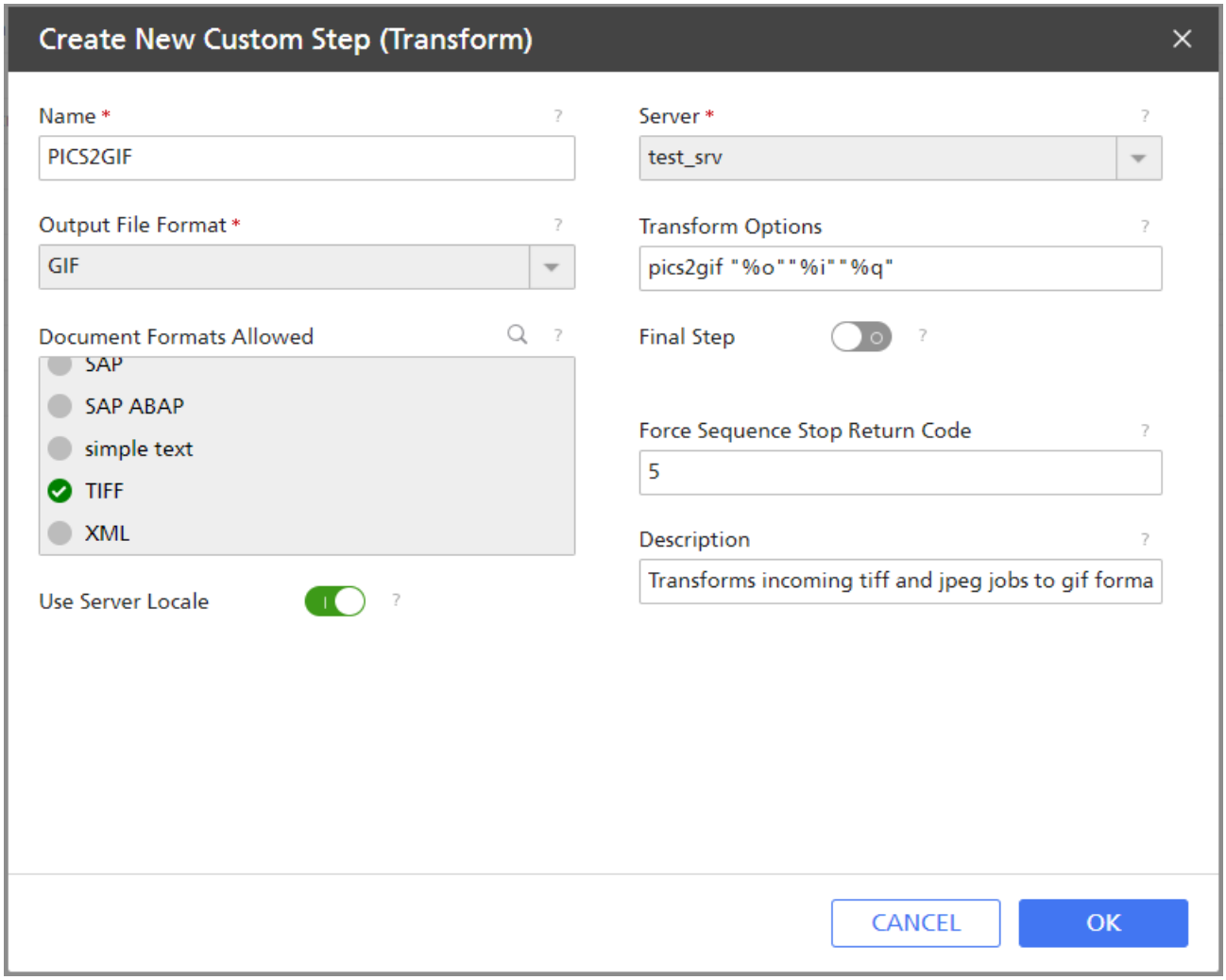
In the example below, the actual destination activates the **pics2gif** transform when incoming JPEG and TIFF jobs enter the system. These jobs are transformed into GIF format and then uploaded to the Internet by an application outside of InfoPrint Manager. When the transformed GIF files are smaller than 3 KB, they are returned to InfoPrint Manager for printing on a proof printer. To observe cost-cutting mandates across the installation, transformed GIF files that are larger than 3 KB are not printed. The transform checks the size of the transformed file and sends a return code of 5 if the job is too big to be printed.

To define a data stream transform object from the InfoPrint Manager Web Administration Interface as a *conditionally-terminating transform*, use this procedure:

1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click , the Menu icon, and select **Custom Steps (Transforms)...**.
4. Click the **New** button in the upper right corner.
5. In the **Create New Custom Step (Transform)** dialog, fill in the fields as shown in the following figure. These values create a transform that transforms both JPEG and TIFF input data into GIF output and then provides a return code of 5 if the transform has completed successfully without returning to InfoPrint Manager for printing.

   In the **Transform Options** field, you can use a series of substitution variables to provide special options for the transform sequences. In this example, besides the standard input (**%i**) and output (**%o**), this transform object also uses the **%q** flag so the InfoPrint Manager server passes the name of the actual destination processing the job.

Create New Custom Step (Transform) dialog: Define a conditionally terminating transform



6. Click **OK**.

   InfoPrint Manager saves the transform under the name you provided in the **Name** field. To see the values of this transform, click ⋮, the Menu icon, on the **Servers** tab and select the **Custom Steps (Transforms)...** option.

Creating a custom step (transform) object that updates the job's page count

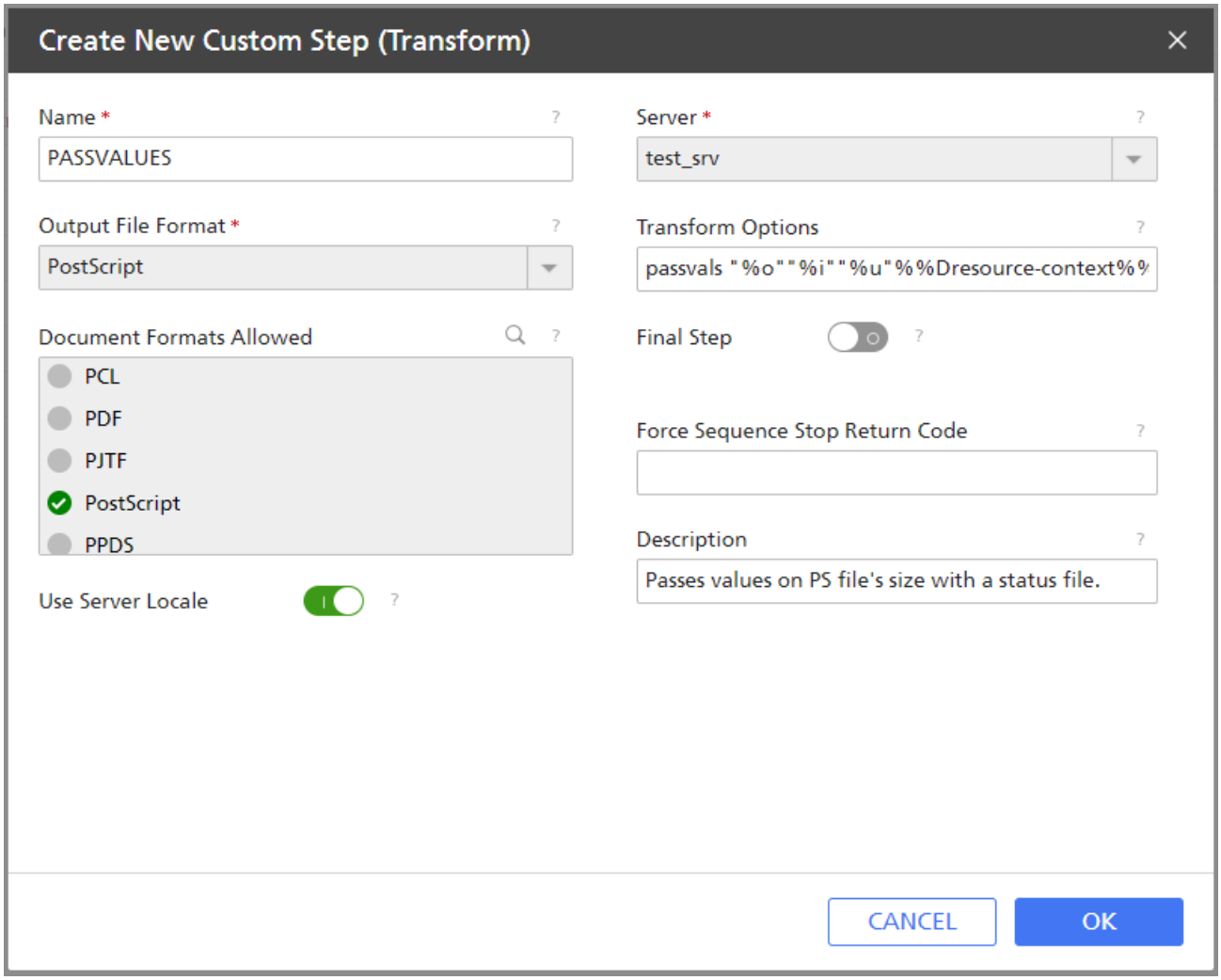
By default, InfoPrint Manager has no knowledge of the number of pages in a transformed job. However, if you have a transform that knows how many pages are in the job, you can create a transform object that can update InfoPrint Manager with this information. InfoPrint Manager can pass transform objects an *update_file* with the *%u* option. This *update_file* can be updated with the number of pages that are in the transformed job, and InfoPrint Manager then updates the job with this information. To modify the *update_file*, the user supplied transform needs to call the **transform_update** utility that is supplied with InfoPrint Manager. For the syntax of the **transform_update** utility, see Using the transform_update utility, p. 284.

To define a data stream transform object from the InfoPrint Manager Web Administration Interface, use this procedure:

1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click ⋮, the Menu icon, and select **Custom Steps (Transforms)...**.
4. Click the **New** button in the upper right corner.
5. In the **Create New Custom Step (Transform)** dialog, fill in the fields as shown in the following figure.

   For this example, the **passvals** program that is called by the **PASSVALUES** transform object gets passed both the **resource-context** document attribute and the **job-owner** job attribute, as well as an update file. You should change the **passvals** program to your transform program that knows how many pages are in the job, and then change the parameters in **Transform Options** to what your program expects with the *%u* being the *update_file* that will be passed to the **transform_update** utility when your transform calls it. Other attributes, like **resource-context** and **job-owner**, can be included or omitted as required by the needs of your transform.

Create New Custom Step (Transform) dialog: Define a transform to pass job attributes



6. Click **OK**.

   InfoPrint Manager saves the transform under the name you provided in the **Name** field. To see the values of this transform, click , the Menu icon, on the **Servers** tab and select the **Custom Steps (Transforms)...** option.

Using the transform_update utility

Use the **transform_update** utility to update InfoPrint Manager about the job page count or number of pages completed.

You can use the **%u** transform option to access an update file that lets you update the document page count or the job's **pages-completed** attribute. Never access the file identified with **%u** directly. The transform should only use the **transform_update** utility. InfoPrint Manager only checks for updates while the transform is running and immediately after it completes. The **transform_update** utility can be called either once or multiple times while the transform is running. Once the transform returns, InfoPrint Manager deletes the update file.

The **transform_update** utility provides these return codes:

1

An English message for basic problems, including:

```
Invalid option specified.
No option or option value specified.
No status file name specified.
Cannot open status file named ____.
Cannot read status file named ____.
```

0

A successful update

-1

The update of the status file was not successful.

```
transform_update [-t delta_pages_transformed]
                 [-d delta_pages_delivered]
                 update_file
```

-t

Specifies the number of additional pages since the last update in the transformed document file. InfoPrint Manager uses this information to recalculate the job's **job-page-count** attribute. The total of values that this option passes cannot exceed `2,147,483,647`.

-d

Specifies the number of additional pages delivered by the transform since the last update. InfoPrint Manager uses this information to update the job's **pages-completed** attribute. The total of values that this option passes cannot exceed `2,147,483,647`.

Often, it is unnecessary for transforms to update the **pages-completed** attribute because it indicates how many pages have actually printed or stacked for the job, and this attribute is updated by the print backend in InfoPrint Manager with the number of pages that were printed. Any pages referenced here should have been removed from any file returned to InfoPrint Manager for further processing.

update_file

The file passed to the transform through the **%u** option.

Note

1. If a transform calls the **transform_update** utility more than once, each call should pass a value for the change since the last call, not a cumulative value.
2. If InfoPrint Manager can neither open nor read the file, it logs an error and continues processing.
3. If you use a transform to update the **job-page-count** attribute, you must provide a page-count for the document. This page-count information provided by the transform should be the pages in the transformed document for a single copy of that document.
4. If a transform updates the **pages-completed** attribute and returns a job for printing, the pages the actual destination completes are added to the pages the transform reported it had completed.

Defining a custom step (transform) sequence

A custom step sequence lets you string together a series of transform objects to do a series of tasks. This topic provides an example of a custom step sequence, followed by the specific procedure you can use to create a sequence of transform objects.

You want to define an actual destination that will copy AFP versions of all print jobs sent to it into a directory on your InfoPrint Manager server system. This directory is used as an archive of AFP versions of print jobs that might need to be viewed later.

To accomplish this, use the InfoPrint Manager Web Administration Interface to create these three custom steps (transforms):

1. **transps**

   You set the input data format to PostScript, the **Transform Options** to use `ps2afp`, and the **Output File Format** to MODCA-P (Mixed Object Document Content Architecture).
2. **transpcl**

   You set the input data format to PCL, the **Transform Options** to use `pcl2afp`, and the **Output File Format** to MODCA-P.
3. **saveafp**

   You set the input data format to AFP, the **Transform Options** to issue a **copy** command, and the **Output File Format** to MODCA-P. In addition, you make this a terminating transform by selecting **Final Step** in the **Create New Custom Step (Transform)** dialog.

Then, create an actual destination that you want to do the sequence. When you create the printer, select **CREATE and EDIT**, then click the **Configuration** tab to define the steps to use as follows:

1. transps
2. transpcl
3. saveafp

Note

If you already have the destination created, open the **Properties** dialog for that actual destination and click the **Configuration** tab to define the steps sequence.

When you send a job to this actual destination, this sequence of actions occurs:

1. **transps** checks to see if the incoming data is PostScript. If it is PostScript,**transps** transforms the data into AFP using `ps2afp`. If it is not PostScript, **transps** does nothing.
2. **transpcl** checks to see if the incoming data is PCL. If it is PCL, **transpcl** transforms it to AFP using `pcl2afp`. If it is not PCL, **transpcl** does nothing.
3. **saveafp** copies the AFP file produced by either of the above steps into a specific directory.
4. Since **saveafp** is a terminating transform, the job is not printed but is marked complete at this point.

To create a custom step (transform) sequence, use this procedure from the InfoPrint Manager Web Administration Interface:

1. In the left-hand **Objects** panel, click the **Printers** tab and select the printer that you want to send the job to.
2. Click , the Menu icon, and select the **Properties** option.
3. In the **Properties for...** dialog, click **Configuration**.

   The data stream transforms that this printer can use are listed in the **Steps to Use** field.
4. Select the transforms that you want to use and drag and drop them in the desired order.

   For example, you might want to create the three-step transform sequence that runs the **transps** , **transpcl**, and **saveafp** transforms in sequence as described above.
5. When you have entered the transforms in the correct sequence, click **OK** to associate this transform sequence with the selected printer.

Using print rules

Print rules are optional rules that are applied to an InfoPrint Manager job if certain criteria are met. You can create, edit, copy, or delete a print rule.

Configuring print rules

To create a print rule, an InfoPrint Manager system administrator must follow this procedure:

1. Connect to the **InfoPrint Manager** server using the **InfoPrint Manager Web Administration Interface**.
2. In the left-hand **Objects** panel, click the **Servers** tab.
3. Click the menu icon ( ) and select **Print Rules**.
4. Click the **New** button in the upper right corner.
5. In the **Create New Print Rule** dialog, fill in or select these values:

   Name

Type a meaningful name for the rule. InfoPrint Manager references the print rule name in print rule sequences that are associated with logical destinations.

Server

If not filled in, either type or select the InfoPrint Manager server on which you want to create the rule.

Use Server Locale

Specify whether to use the locale language of the server (**On** - default) or English (**Off**) for input and output of information.

Description

Type a description of this print rule to appear whenever you show this print rule through the InfoPrint Manager Web Administration Interface. A detailed description of the print rule can be helpful to users who need to determine which rule to use.

Find items which

Indicate the logical conditions between existing attribute values to trigger the actions to be performed. Choose one of the options:

Match all conditions (AND logic)

All conditions must be satisfied for the defined actions to be performed.

Match any conditions (OR logic)

At least one of the conditions must be satisfied for the defined actions to be performed.

Add condition

Define one or more conditions for a print rule. From left to right, each condition consists of:

- a **job** or **document attribute**
- a **condition operator**
- a **value**

attribute

Select from the list the job or document attribute that you want to use for the condition.

condition operator

Specify the relation between the attribute and the value to decide whether a job meets the condition.

Note

- The list of available condition operators depends on the selected job or document attribute.
- All attributes allow the condition operators **Has a value (set)** and **Has no value (not set)**, for which you cannot select or enter any value.
- You can activate or deactivate the case-sensitive option for some attribute values.

value

Specify the value that InfoPrint Manager uses to decide whether a job meets the condition. Depending on the attribute, you can either choose from a list or enter an alphanumeric value.

Note

- To define an additional condition, click the **Add condition** button. To delete a condition, click the delete button ( ) to the right of the condition you want to delete.

Add action

Select one or more actions to be performed if the defined logical conditions are met. Choose one of the options:

Hold job

Change the state of the job to **held**.

Set attribute

Select a job or document attribute and specify a value to it. This action can be added multiple times.

Run script

Specify the custom script to be run if the defined logical conditions are met. You can use the supported substitution control sequences described in Running a script in print rules, p. 290.

In the **Script success return code** box, enter the return code that indicates the custom script has been executed successfully.

Route job

Specify the actual destination where to move the job.

Note

- You can select one value at a time from the list of possible destinations or enter a value.
- The **Route job** and **Split job** actions are mutually exclusive.

Split job

Specify the actual destinations to which to split the job, based on the number of pages.

Note

- This action is available only if the document format is MODCA-P and the condition **Number of Pages Is Greater Than or Equals** is applied. The maximum number of pages for each of the resulting jobs is equal to the number of copies multiplied by the number specified for the **Number of Pages Is Greater Than or Equals** condition.
- The maximum number of pages for each of the resulting jobs cannot exceed the number of pages of the document.
- The actual destinations attribute of the job is overwritten.
- If you select specific printers from the list, those printers must be assigned to the same queue as the queue of the logical destination where the print rule is attached.

Send email

Send an email to a specific recipient. Choose one of the options:

Send on error (default)

The email is sent only when errors occur.

Send always

The email is sent on all jobs from the rule.

Note

- For multiple recipients, use ',' as a separator.

Terminating rule

Specify whether a rule is the last one in a sequence.

Note

- This action is added automatically after the **Split job** and **Route job** actions.
- A terminating rule affects only the current print rule sequence. If a rule moves the job to a different logical destination, any existing print rule sequence assigned to that logical destination will still be run.

Note

- For the **Set attribute** and **Route job** actions, you can use the following supported substitution control sequences:

  %%J*job_attribute*

  Passes the job attribute value from the job containing the document being processed.

  %%D*doc_attribute*

  Passes the document attribute value from the document being processed.

  If an attribute is multi-valued (such as **%%Jattribute-name**, **%%Dattribute-name**), only the first value of the attribute is used.
- To define an additional action, click the **Add action** button. To delete an action, click the delete button (🗑) to the right of the action you want to delete.

6. Click the **OK** button.

Running a script in print rules

You can specify a command that executes when a print rule is run using the substitution values shown in the "Print rules substitution variables" table.

Print rules substitution variables

Substitution variable	Meaning
%i	The name of the input file.
%o	The name of the file in which to store the output.
%j	The name of the original print file, minus the path (from the **document-file-name** attribute). Note This should only be used to construct another name, such as a temporary file.
%n	The name of the original print file, minus the path and minus the extension (from the **document-file-name** attribute). Note This should only be used to construct another name, such as a temporary file.
%d	A text string that represents the document format of the input file.

Substitution variable	Meaning
	Note You can also acquire document format through the **%%Ddocument-format** substitution value.
%p	The value of the document's **destination-pass-through** attribute.
%s	The **document-sequence-number** for the document.
%t	The value of the document's **other-transform-options** attribute.
%g	The global job identifier for the job.
%#	The job-identifier (10 digits maximum).
%%D*doc_attribute***%%**	The value of the InfoPrint Manager document attribute specified. For example, if you use `%%Ddepartment-text%%`, InfoPrint Manager passes the document's **department-text** attribute value. Note See the "Considerations when using the %% *Nattributename* substitution control sequences" topic under the "Usage guidelines" for the **other-transform-options** attribute for transforms in the *RICOH InfoPrint Manager: Reference* before using this substitution variable.
%%J*job_attribute***%%**	The value of the InfoPrint Manager job attribute specified. For example, if you use `%%Jjob-owner%%`, InfoPrint Manager passes the **job-owner** attribute value of the job that owns the document. Note See the "Considerations when using the %% *Nattributename* substitution control sequences" topic under the "Usage guidelines" for the **other-transform-options** attribute for transforms in the *RICOH InfoPrint Manager: Reference* before using this substitution variable.

Note

- Each substitution variable must be passed as a single parameter to the script. You must not concatenate substitution variables with one another or with any other text.
- If you are specifying any substitutions for which the passed values might contain a blank space, such as `job name`, be sure to put quotes around the substitution so it is passed correctly. Note that this only applies to single % substitution variables. Do not put quotes around any %% substitution variables.

Important

- You cannot run a job modification command such as **pdset** or **pdmod**.

Defining a print rule sequence

A print rule sequence lets you string together a series of rules to do a series of tasks. This topic provides the specific procedure you can use to create a sequence of print rules.

To create a print rule sequence, use this procedure from the InfoPrint Manager Web Administration Interface:

1. In the left-hand **Objects** panel, click the **Logical Destination** tab and select the destination that you want to send the job to.
2. Click , the Menu icon, and select the **Properties** option.
3. In the **Properties for...** dialog, click **General**.

   The available print rules are listed in the **Print Rule Sequence** field.
4. Select the print rules that you want to use and drag and drop them in the desired order.

   For example, you might want to create a three-step print rule sequence that runs the previously defined print rules in a determined order as described above.
5. When you have entered the print rules in the correct sequence, click **OK** to associate this print rule sequence with the selected destination.

Printing PCL or PostScript to PSF destinations when using the CPSI transform engine: Specifying which paper bin to use

In general, when you submit a print job through a PCL or PostScript driver, you can use the printer options to specify which input paper bin you want the printer to use. For example, if you know that white 8.5 × 11 inch paper is always loaded in bin one and letterhead is always in bin two, you can select bin one when you want to print a draft and bin two when you're ready to print a real letter.

When you send PCL or PostScript print jobs to a PSF Destination in InfoPrint Manager, the data stream is transformed into Advanced Function Presentation (AFP) before it is printed. By default, most PCL jobs that are transformed use paper bin one, regardless of the bin you specify. Transformed PostScript jobs default to letter-sized paper if no other options are specified, and look for a bin that contains that size paper; they either print on paper from the first bin that contains the right size or fail because they do not find the right paper.

To set up bin mapping, see Editing configuration files, p. 298.

Determining bin mappings

If you want the transform program to check for your bin selection and include it in the AFP data that is sent to the printer, you must specify mappings between the PCL and PostScript bins and the AFP bins. The transform will use your mappings to convert the bin number it finds in the PCL or PostScript into an

AFP bin number. You can configure InfoPrint Manager so that you can specify up to 20 different input paper bins.

PCL printing

When you submit a print job through a PCL print driver and select a paper bin to print from, the driver generates a bin number and puts it into the data stream. Unfortunately, different drivers generate different bin numbers for the same bin. For example, one driver might make "Tray 1" equal to bin 4 and "Auxiliary tray" equal to bin 2, while another driver makes "Tray 1" equal to bin 1 and "Auxiliary tray" equal to bin 8. To add more difficulty, the generated PCL bin numbers can range from 0 to 59. While most drivers use low numbers (0 through 10), keep in mind the other possibilities.

During the transformation from PCL to AFP, InfoPrint Manager takes the bin number that the driver generated and compares it to the bin mappings you set. The pcl2afp transform inserts the AFP bin number that you assigned and sends the job to the destination you requested. The printer prints the job on paper from the bin you chose. As a result, it is essential that all of the users who submit jobs to a given InfoPrint Manager printer use the same driver. If they do not, the mappings you set up will not work correctly for them.

The first challenge for you in setting up the mappings is determining what bin number the PCL driver places in the data stream. For various InfoPrint printers, the drivers actually generate the same bin numbers. If you are using the PCL5e driver, the numbers listed in InfoPrint PCL5e bin numbers, p. 293 are placed in the data stream.

InfoPrint PCL5e bin numbers

Tray (what you select in Device Settings)	PCL bin number (what the driver puts in data stream)
Automatically Select	0
Tray 1	1
Manual feed paper	2
Manual feed envelope	3
Auxiliary tray	4
Tray 2	5
Envelope feeder or Envelope tray	6
Tray 3	7
Tray 4	8
Tray 5 or 2000 sheet input tray	9

If you are using a different driver, the documentation that came with your printer might provide these numbers; look for a section on the PCL driver. Alternatively, if you are able to capture the PCL data that is sent to a specific bin in your printer, you can look for the bin number in the data stream. In some cases, however, it might be just as easy to use trial and error to determine the correct mappings. Again, keep in mind that PCL bin numbers can range from 0 to 59.

The second challenge is in determining the AFP bin numbers that correspond to your printer. AFP bin numbers range from 1 to 255, and are not standard from printer to printer. Some printers have hard-coded bin numbers: each bin on the printer actually has a number written on it. In other cases, the AFP bin numbers are more difficult to discover. In fact, the numbering scheme for some printers changes completely when you add one of the optional paper bins, and still other printers allow you to change the bin numbers yourself. As a result, it is difficult to predict exactly what the AFP bin numbers for your printers are. However, there are some guidelines that you can use.

- Most AFP paper bins use numbers 1 through 10, with 1 being the biggest, topmost, or bottommost bin.
- Envelope feeders/bins start at number 65 and can go up to 69.
- The manual feed tray is usually number 100.

If you are printing to InfoPrint printers, the best place to find the default numbering schemes for each printer is the *IPDS and SCS Technical Reference* and the *IPDS Handbook for printers that use the AFCCU*, both of which are available from the Ricoh website at https://manuals.ricoh.com/pps/download/pdf/ipds_tech_ref.pdf.

If you find the AFP bin numbers for your printer in this documentation, be aware that the number is written in hexadecimal notation. You must convert the number to decimal before you put it in the mapping file. In addition, the numbers that are listed are the machine numbers, not the numbers the software requires. After you convert the number to decimal, add 1 to it to get the number that you should put in the mapping file.

Given this information, the best way to determine the correct AFP bin numbers is to start with these guidelines and use the trial and error method until you have the mappings set correctly.

For example, if you have a printer with three paper bins and a manual feed bin, you could do this.

1. On your InfoPrint Manager Windows client system, install the PCL driver that you plan to use.
2. Create a printer that uses the driver you just installed and has a PSF printer as its target destination.
3. From the Windows Client, click the **Start** button and select **Settings** → **Printers** to open the **Printers** window.
4. Right-click the printer you just created and select **Properties** from the pop-up menu.
5. Click the **Device Settings** tab to show the printer options that are available for this driver. For example, some drivers show a list of "paper trays" and allow you to select a paper size for each one.
6. Make sure all of the paper trays you plan to use are activated and have the correct size paper selected.
7. Click **OK** to close the dialog and make the settings take effect.
8. Follow the appropriate procedure below: Editing configuration files, p. 298.
   When you have to enter PCL and AFP bin numbers, start with the guidelines above and create some logical mappings, such as the ones shown in Sample mapping table for a Ricoh printer, p. 295.

Sample mapping table for a Ricoh printer

Tray selection	PCL bin number	AFP bin number
Tray 1	1	1
Tray 2	5	2
Tray 3	7	3
Manual feed tray	2	100
Envelope tray 1	6	60
Envelope tray 2	4	61

Complete the procedure.

9. Load each paper bin (including the manual feed bin) with a different kind, color, or specially marked paper.
10. Submit a test job to each bin and see which paper it is printed on. Keep track of your results and use them to change the mappings to meet your configuration.
11. Once you have the mappings worked out correctly, tell your operators this information:
    - What options to set in the device settings.
    - What kind of paper is in each bin.
    - How to print to each bin.

PostScript printing when using the CPSI print engine

PostScript drivers use different methods to determine what paper bin to print a job on. Some drivers allow you to select which paper tray you want the paper to come from. When you submit a job, the driver sends a number to the printer to ask for the corresponding tray. Other drivers allow you to select different paper options (such as size, color, or weight). When you submit a print job, these options are sent with the print job and are used to determine which paper bin to use (set the paper size option for this function to work).

When you configure InfoPrint Manager correctly, it works with either type of driver. There are three steps involved in performing this configuration:

1. Gathering paper tray information, p. 295
2. Identifying paper trays to InfoPrint Manager, p. 296
3. Testing tray mappings, p. 297

Gathering paper tray information

The first step in configuring InfoPrint Manager is to figure out :

- How many paper trays does your printer have?
- What is the AFP bin number for each tray?
  Guidelines for figuring out the bin numbers are provided below.
- What kind of paper will you be loading into each tray?

At a minimum, you should know what size paper will go in each tray. If you need to, you can also specify a color or weight of the paper.

Use this chart to record your data.

Paper tray chart

AFP bin number (in decimal notation)	Paper tray on printer	Paper loaded
100	Manual feed tray	

There is trial and error work involved in figuring out the correct bin numbers. AFP bin numbers range from 1 to 255, and are not standard from printer to printer. Some printers have hard-coded bin numbers; each bin on the printer actually has a number written on it. In other cases, the AFP bin numbers are more difficult to discover. In fact, the numbering scheme for some printers changes completely when you add one of the optional paper bins. Still other printers allow you to change the bin numbers yourself. As a result, it is nearly impossible to predict exactly what the AFP bin numbers for your printers are. However, there are some guidelines that you can use.

- Most AFP paper bins use numbers 1 through 10, with 1 being the biggest, topmost, or bottommost bin.
- Envelope feeders/bins start at number 65 and can go up to 69.
- The manual feed tray is usually number 100.

If you are printing to InfoPrint printers, the best place to find the default numbering schemes for each printer is in the *IPDS and SCS Technical Reference* and the *IPDS Handbook for printers that use the AFCCU*.

Important

If you find the AFP bin numbers for your printer in this documentation, be aware that the number is written in hexadecimal notation. You must convert the number to decimal before you put it in the mapping file. In addition, the numbers that are listed are the machine numbers, not the numbers the software requires (they're the *IPDS* bin numbers, not the *AFP* bin numbers). After you convert the number to decimal, add 1 to it to get the number that you should put in the mapping file.

Identifying paper trays to InfoPrint Manager

Once you have the chart filled in with your first attempt at mappings, you can move to the second step: identifying all of the bins to InfoPrint Manager.

By default, InfoPrint Manager assumes that all PostScript printers have only one paper tray. When you identify the trays, you let InfoPrint Manager know how many there really are. Then, when you submit a job, you can select any one of them.

You identify the paper bins to InfoPrint Manager either in the configuration file or in the transform you create. In this step, you will determine the syntax that you will be able to use in either place. The general syntax of the identification statement is:

input*X***=(***size***,type=***xxx***,weight=***xxx***,color=***xxx***),input***X***=(***size***, type=***xxx***,weight=***xxx***,color=***xxx***)**

Where these conditions apply:

- Each **input***X* corresponds to a different paper tray. You should have as many **input***X* entries as you have paper trays. Separate them with commas, no spaces.
- The *X* in each **input***X* is replaced with the **AFP bin number** from Paper tray chart, p. 296 for each tray.
- Size is the only required paper attribute. Replace the word *size* with the size of the paper that will be in the tray, such as **letter**, **legal**, **8.5ix11i**, or **A4**.
- Other paper attributes can be specified or left out entirely.

All of these are valid identification statements:

```
input1=(letter),input2=(legal),input100=(letter)
```

```
input1=(8.5ix11i,color=blue),input2=(8.5ix11i,color=yellow),
input3=(11ix17i),input100=(8.5ix11i)
```

```
input1=(A4,color=white),input2=(A4,color=blue),input3=(A4,color=yellow),
input65=(C5),input100=(A4,type=letterhead)
```

Write out your identification statement before moving on to the next step.

Testing tray mappings

Once you have your identification statement created, you can test your mappings to make sure they are correct. This procedure explains one way to verify the mappings.

1. On your InfoPrint Manager Windows client system, install the PostScript driver that corresponds to your printer.
2. Create a printer that uses the driver you just installed and has a PSF printer as its target destination.
3. Click the Windows **Start** button and select **Settings** → **Printers** to open the **Printers** window.
4. Right-click the printer you just created and select **Properties** from the pop-up menu.
5. Click the **Device Settings** tab to show the trays and options that are available.
6. If your driver lists the paper trays that are available, activate them and select paper sizes for each one. Set the options to match your identification statement.
7. Click **OK** to close the dialog and make the settings take effect.
8. See Editing configuration files, p. 298 so you can use the same mappings for all of your printers.
9. Follow the appropriate procedure.

   When you have to change the configuration file or enter the command that your transform will use, replace the **input***X***=**... statement with your identification statement from Identifying paper trays to InfoPrint Manager, p. 296, and complete the procedure.
10. Load each paper bin (including the manual feed bin) with a different kind, color, or specially marked paper.

11. Test your settings by submitting print jobs to each bin of the Windows gateway printer and making sure that it prints on the correct paper. If a job does not print on the correct paper, adjust your AFP bin numbers and input numbers as necessary.
12. Once you have the mappings worked out correctly, tell your operators this information:
    - What options to set in the device settings.
    - What kind of paper is in each bin.
    - How to print to each bin.

Editing configuration files

There are five configuration files that come with the InfoPrint Manager PCL to AFP and PostScript to AFP transforms. For the PCL transform, they are: `pcl2afp.cfg` and `pcl2afpd.cfg`. For the PostScript transform they are: `ps2afp.cfg`, `ps2afpd.cfg` (used with InfoPrint 4000 and InfoPrint 4100 printers), and `3160d.cfg` (used with all other IPDS printers). You only need to edit one of these files based on the data stream that your driver produces, the printer you send your jobs to, and on these considerations.

- Edit `pcl2afpd.cfg`, `ps2afpd.cfg`, or `3160d.cfg` if you want everyone to use the same mappings. This method is recommended.
- Edit `pcl2afp.cfg` or `ps2afp.cfg` if different people have to use different mappings.

Follow these steps to edit the appropriate file.

1. On the system that your InfoPrint Manager server is installed on, locate the file.
2. Modify the file according to the instructions in the file.

   Choose one of the options and uncomment that line, then enter your mapping numbers. If you are using the ps2afp transform, replace the **input***X***=**... portion of the **device_controls** line with your identification statement from Identifying paper trays to InfoPrint Manager, p. 296.
3. Save the file.

   Note

   It is recommended that you save a copy of your changes in a directory outside of the `/usr/lpp/psf` path so that you have a copy in case you ever have to re-install or apply service to InfoPrint Manager.
4. Close the file.
5. Restart the configuration daemon (for example, for PCL, specify `pcl2afpd -C /usr/lpp/psf/pcl2afp/pcl2afpd.cfg` at the Linux command line) to have the new settings take effect.
6. Tell your users what kind of paper is in each bin and how to print to each of them.

Creating a transform with bin mappings

You can use the InfoPrint Manager transforms to process print jobs with different options, including which paper bins to use. First, you create a transform using the InfoPrint Manager Administration GUI, specifying an input data stream and an output data stream. Then, you associate it with one or more

printers. When you send a print job to that printer, your transform checks to see if the file is the data stream that needs to be transformed. If it does, the transform converts the file as you specified; if it is not, the transform passes the file to the logical destination. As a result, if you do not want to use the same bin mappings for all of your printers, you can create several different transforms, all with different mappings. Then, associate each transform with whichever printer or printers you want.

Note

If you will be sending both PCL and PostScript jobs to the same printer, you can create two transform objects (one that accepts PCL and one that accepts PostScript) and associate them both with that printer.

Follow these steps to create a transform that maps PCL or PostScript bins to AFP bins.

1. In the InfoPrint Manager Administration GUI, click the **Server** menu and select **Transform** → **Create**.
2. In the **Create transform** dialog, fill in the fields:
   - **Name:** Type a name for the transform.
   - **Server:** Choose the correct InfoPrint Manager server.
   - **Format of output file:** Select **MODCA-P**.
   - **Description:** Type a description that will help you remember which printer uses this transform.
   - **Transform options:** The command line that will execute when the transform runs.

     If this transform accepts PCL, type:

```
pcl2afp  -o "%o" "%i" -device 'plex,inputX=(pcl_bin=Z),
   inputX=(pcl_bin=Z)'
```

     Replace the *X*s with the number of the AFP bin (on the printer) and *Z*s with the number of the PCL bin. You can specify up to 20 mappings.

     If this transform accepts PostScript, type:

```
ps2afp  -o "%o" "%i" -device 'plex,inputX=(size,type=xxx,color=xxx,
   weight=nnn), inputX=(size,type=xxx,color=xxx,weight=nnn)'
```

     Replace the **input***X***=**... portion of the command with your identification statement from Identifying paper trays to InfoPrint Manager, p. 296. Replace *size* with a paper size such as letter, legal, A4, or the dimensions of the paper in inches (`8.5ix11i`), millimeters (`216mx279m`), or points (`612px792p`) and the *X*s with the number of the AFP bin (on the printer). *Size* is the only required value; the others you can either fill in or delete.
   - Select **Transform and print**.
   - **Document formats allowed:** The data stream that your transform accepts. Select **PCL** or **PostScript** and add it to the allowed values.
   - Click **OK**.
3. Select the printer that you want to associate this transform with and open its properties notebook.
4. Click the **Configuration** tab.
5. Find the **Transforms to use** field.
6. Select the transform you just created in the **Possible values** box and add it to the **Values** box.
7. Click **Apply**.
8. Click the **Document** tab.

9. Find the **Formats allowed** section.and look in the **Values** box. Make sure that **modca-p** and either **pcl** or **PostScript** are listed.
10. Click **OK**.
11. Tell your users what kind of paper is in each bin and how to print to each of them.

Submitting PSF printer input to a PCL or PPDS printer

In addition to AFP printing, InfoPrint Manager allows you to submit PSF printer input data to PCL and PPDS printers as well. There are two methods that you can use to do this: using a PSF command DSS or PSF other-driver DSS. If you choose to use the PSF command DSS, you need to be aware of the following information. See the *RICOH InfoPrint Manager for Linux: Getting Started* guide for additional information.

When you use the Create Printer wizard in the InfoPrint Manager Administration GUI to create a printer object that will send jobs to a PCL or PPDS printer, you must specify the path for the command in the **Printer command** field. Because InfoPrint Manager does not search the **PATH** environment variable for the command, you must either add that directory to the **PATH** for that system or specify the entire path of the command, as in this example:

```
/usr/bin/lpr -H servername -P printername
```

If you need to specify a file name in the command in a location other than at the end of the command, you must include the text-string **-IPMF-** in the command string. The temporary file name used to store either the PCL or PPDS data generated by InfoPrint Manager is substituted into the command wherever the **-IPMF-** text-string is located.

Note

The dashes (**-**) on either side of the **-IPMF-** text-string are part of the syntax and cannot be omitted.

You can only specify one **-IPMF-** text-string in a print command, as in the example below:

where `copyit` represents a user-written application for moving either the PCL or PPDS file that InfoPrint Manager generated to the specified directory and giving it a unique name.

Setting up PCL tray mappings

Follow these steps to map the input trays requested in the AFP data stream to the input trays on your PCL printer:

1. Determine the AFP bin number and PCL bin number for each input tray. Default bin mappings, p. 301 shows the default bin numbers associated with some standard names for input trays.

   Note

   You can use any tray names you wish, but these names are defined by the PCL data stream and are the most common names. For more information about the limitations of trays and paper with a PSF other-driver or PSF command DSS see the **psf-tray-characteristics** attribute in the *RICOH InfoPrint Manager: Reference*.

Default bin mappings

Input tray name	AFP bin number	PCL bin number
alternate	3	5
bottom	2	4
envelope	65	6
manual	100	2
top	1	1

If these mappings are not appropriate for your printer, reverse the process described in Determining bin mappings, p. 292. In this case, the AFP bin number is the bin number in the data stream and the PCL tray number is the bin number used by the printer. Note that AFP bin numbers must be unique, but different AFP bin numbers can map to the same PCL tray number.

2. Open the InfoPrint Manager Administration GUI.
3. Click the printer that you want to set up trays on.
4. From the **Media/Bins/Trays** tab of the Actual Destination Properties notebook, in the **Media ready** table, select **Add** to define a new tray or **Modify** to change an existing one. You can define up to 15 trays.
5. Complete these fields:

Input bin number

This field specifies the AFP bin number.

Media ready

This field shows the medium loaded in the input tray. The drop-down list lets you select from all media supported by the printer.

Input tray name

This field shows the name associated with the input tray. Standard names for PCL input trays are:

alternate

envelope

lower

manual

upper

Formerly, these were the only names allowed. You can now use any name you like, subject to these restrictions:

- Manual trays must have the string **manual** somewhere in the name, in any combination of upper and lowercase, for example, `transparentMANUALa4`. You can define multiple manual trays.
- Envelope trays must have the string **envelope** somewhere in the name, in any combination of upper and lowercase, for example, `Number10Envelope`. You can define multiple envelope trays.

Tray number

This field specifies the PCL bin number.

Duplex

This field specifies whether two-sided printing is possible from this input tray.

6. Click **OK**.

To make your changes take effect, you must now stop and restart PSF, the printer, or both. Follow the steps corresponding to your job submission method.

- If you use DPF to submit jobs, perform the following steps:
  1. On your *host* system, stop PSF.
  2. In the InfoPrint Manager GUI, click the printer to select it, then select **Printer** → **Stop**.

     Note

     If you do not see **Stop** on the Printer menu, use **Add/Remove Menu Items** to add it.
  3. Open the Management Console and click the DPF Host Receivers item in the left pane.
  4. Click the DPF Host Receiver that feeds this printer to select it.
  5. Click **Actions**→ **Obtain Printer Characteristics**.
  6. Go back to the InfoPrint Manager GUI and start the printer again.
  7. Restart the host PSF.
- If you use any other method to submit jobs:
  1. In the InfoPrint Manager GUI, click the printer to select it, then select **Printer** → **Stop**.

     Note

     If you do not see **Stop** on the Printer menu, use **Add/Remove Menu Items** to add it.
  2. In the **Stop command** dialog, select **Shut down** and click **OK**.
  3. Select **Printer** → **Enable** to restart the printer.

Color and Grayscale Printing

Printing documents in full color or with very high-quality black and white (grayscale) images is more complex than printing black and white or spot-color documents. Understanding some of the principles of color and grayscale printing and how various products can fit into color and grayscale solutions can help you integrate color and grayscale printing with your current operations or expand to implement new color workflows.

AFP color and grayscale solutions

You can assemble printing products from Ricoh and other companies in different configurations to support Advanced Function Presentation (AFP) color and grayscale printing, including configurations that use the AFP Color Management Object Content Architecture (CMOCA) to provide optimal performance and color accuracy in high-speed color printing.

Color printing without explicit color management

You can include color images or specify colors for AFP objects in your print jobs and send them to an AFP color printer. The color images and objects print in color, based on the default settings in your print server and printer.

If you like the colors that the default color management settings provide, or if it is not essential that you print in exactly the right colors, you probably do not need to implement a full color management solution. However, if you want better control over the consistency and accuracy of your colors for devices, you might consider color management at some point in the future.

To understand some of the basic concepts about color printing, see:

- Color Printing Concepts, p. 304
- Grayscale Printing Concepts, p. 308

Resources that document composition software includes inline

Most often, print bureaus use document composition software to generate highly customized and personalized color output. AFP color management is largely built into the document composition tools that support their processes. The software puts all the resources that the printer needs into the print job and sends it to a print server. The print server sends the print job to the printer, and the printer uses the resources as appropriate.

By using this method, you know that the required resources, including the resources required for color management, are available for any print job that is sent to the printer. However, including all the resources can make the print job very large, and moving large print jobs through your system might slow down system performance. Also, you might not be able to save resources that are downloaded with a print job on the printer so they can be reused without being downloaded later.

For more information about color printing, see:

- General information about color printing and color management:
  - Color Printing Concepts, p. 304
  - Color Management, p. 309
- Tips for images, p. 320
- Scenario describing a possible implementation:
  Eliminating physical inserts, p. 330

For a list of the companies that participate in the AFP Consortium and support AFP color management in their products, see:

http://www.afpcinc.org

Resources stored and managed centrally

To take full advantage of the AFP CMOCA, you can store your color and image resources in a central resource library, and let your print system manage those resources. This option optimizes system performance by:

- Creating some of the color management resources for you automatically

- Reducing the number of color conversion resources that the system creates at print time by generating link color conversion color management resources in advance
- Reducing the size of some images by removing embedded profiles when you store them, yet still retaining the association between the image and the profile
- Letting you mark resources as *capturable*, so they can be saved on the printer and used in other print jobs without being downloaded again

For an introduction to the full AFP CMOCA and how you might implement it, see:

- General information about color and grayscale printing and color management:
  - Color Printing Concepts, p. 304
  - Grayscale Printing Concepts, p. 308
  - Color Management, p. 309
- AFP Color Management, p. 311
- RICOH AFP Resource Installer, p. 323
- RICOH AFP color and grayscale products, p. 321
- AFP color solution scenarios, p. 325
- Related publications, p. 331
- Scenario describing a possible implementation:
  Replacing pre-printed forms, p. 327

Color Printing Concepts

Color printing is significantly more complicated than black and white printing. If you understand some of the complexities, you can make the transition from black and white printing to grayscale or color printing more smoothly.

Color Spaces and ICC profiles

Presentation devices, such as computer monitors and printers, create colors differently. Because of these differences, colors must be described differently for each device. The different methods of describing colors are called *color spaces*. In addition, each device might have one or more International Color Consortium (ICC) profiles associated with it. ICC profiles are used when an image or another object is converted to the color space of a different device.

Each device has its own individual color space and range of colors that it can display or print. The color space specifies how color information is represented in an image when it is displayed on a particular device. As the image is passed from one device to the next, the color information about the image is converted from the color space of the source device to the color space of the destination device. Because color spaces do not exactly match between devices, some of the color information can be lost or modified in the conversion process.

A *color space* is a representation of the individual colors that can be combined to create other colors. Some color spaces that are relevant to printing are:

- **RGB**
  In an RGB color space, red, green, and blue light are combined in different amounts and intensities to create different colors. RGB colors are often specified as single-byte integers numbered from 0 through 255. You can specify 256 levels of intensity for each of the three colors. For example:
  - R=0, G=0, B=0 yields black
  - R=255, G=255, B=255 yields white
  - R=251, G=254, B=141 yields a pale yellow
  - R=210, G=154, B=241 yields a light purple

  Devices such as monitors, digital cameras, and scanners generally use RGB color spaces to describe colors. Two standard implementations of RGB color spaces are **sRGB**, which is most often used for web graphics, and **Adobe RGB (1998)**, which we recommend for graphics that are printed.
- **CMYK**
  In a CMYK color space, cyan (bright blue), magenta (bright red-pink), yellow, and black pigments are combined to create different colors. CMYK values are often represented as a percentage. The percentage represents the portion of a particular area of paper that is covered by ink or toner. For example:
  - C=0%, M=0%, Y=0%, K=100% yields black
  - C=0%, M=0%, Y=0%, K=0% yields a blank area on the page
  - C=1.6%, M=0%, Y=44.7%, K=.4% yields a pale yellow
  - C=17.6%, M=39.6%, Y=5.5%, K=5.5% yields a light purple

  Color printers use the CMYK color space; they are loaded with ink or toner in each color. When the printer places dots of the correct sizes next to and on top of each other on a page, your eye interprets them as the intended color.

  Implementations of the CMYK color space vary from printer to printer and from paper to paper. Because the original color space of most images is an RGB color space, it is best to leave images in an RGB color space so they retain their original characteristics. That way, your print server or printer has as much of the original color information as possible when it converts the images to the most appropriate CMYK color space for the printer and paper combination.

  If you save an image by using the CMYK color space, make sure that you either save an ICC profile for that color space or use a standard non-device specific CMYK color space like **SWOP** or **Coated FOGRA27** and associate the appropriate ICC profile with the image.

Note

- Both RGB and CMYK values can be expressed in different ways. For example, in the PostScript data stream, the values range from 0.0 to 1.0, while in some graphic arts programs they can be expressed in hexadecimal numbers or as percentages.

An *ICC profile* contains information for converting an image between a device-specific color space and a *device-independent color space*. A device-independent color space is a color space that does not depend on or relate to the characteristics of any particular device, but rather contains all colors for all gamuts. The ICC identified a specific profile connection space (PCS) as the target device-independent color space for all ICC profiles.

You can use an input ICC profile to translate color data created on one device (such as a digital camera) into the PCS. Then you can use an output ICC profile to convert from the PCS into the native color space of a different device (such as a printer). Converting images from one color space to another is process-intensive and can affect performance in your print system, although it is the best way to maintain consistent color for the devices in your system.

Gamut and Rendering Intent

Every device has a *gamut*, a range of colors or shades of colors that it can display or print. Some devices have larger gamuts than others; some devices have gamuts that are similar sizes, but that contain slightly different colors. When an image or a print job is created on a device with a gamut that is different from the printer, you can use a *rendering intent* to tell the printer how to adjust the colors that are outside the gamut of the printer.

The gamut of a printer is almost always significantly smaller than the gamut of a monitor, digital camera, or scanner. Images or graphics nearly always have to be adjusted to print appropriately because some of the colors that they require might be outside the gamut of the printer.

A rendering intent tells the printer how to adjust the image when it encounters colors that it cannot reproduce. Each rendering intent has different benefits and trade-offs, so you can choose one based on how the print output should look.

Color Mixing and Calibration

Four standard colors (cyan, magenta, yellow, and black) are blended to create all the colors in the gamut of a printer. A printer mixes colors by printing four layers of a page or an image, one in each color. If the printer registration is not set correctly, the images do not line up properly and the colors appear wrong. In addition, the printer must be calibrated to ensure that all its systems are functioning correctly and that it is in a known good state.

By using the color information described in its color space, each device determines the amount of cyan, magenta, yellow, or black to use. Dots of each color are printed in overlapping patterns that, when interpreted by your eyes, blend the colors appropriately. To ensure the colors are created accurately, the color planes must be perfectly aligned. If they are not, you might see *moire patterns*, unintended patterns in the printed images, or poorly blended colors, which are especially noticeable on the edges of your images.

Color printers must be calibrated regularly, in some cases daily, to ensure that the colors they produce are consistent. In addition, follow the recommended printhead maintenance procedures and schedule to ensure that the printer operates optimally. Even when a printer is calibrated correctly, its gamut is much smaller than that of any monitor, so images do not look the same when they are printed as they do when they are displayed on a monitor.

Halftones and Tone Transfer Curves

Halftones are used to convert images (such as photographs, drawings, logos, or charts) from the continuous tones that you see on a monitor into a pattern of dots that a printer can put on paper. Tone transfer curves are used to modify the values of a particular color component and thus adjust the look and feel of some of the colors. For example, you can apply a tone transfer curve to emphasize the brightest parts of an image.

Halftones and tone transfer curves are used with both color and grayscale print jobs.

There are several different kinds of halftones, including clustered-dot, stochastic, and error diffusion. For simplicity, this discussion only covers clustered-dot halftones.

Clustered-dot halftones are generally characterized by:

- **Line screen frequency**

  Line screen frequency is a measure of the resolution of a halftone, expressed in lines per inch (lpi). A low line screen frequency, such as 80 lpi, creates coarser images because they use larger halftone dots. A high line screen frequency, such as 150 lpi, can produce higher quality images by using smaller halftone dots.

- **Halftone pattern**

  Halftone dots are printed in various shapes and patterns. For example, dots can generally be round, elliptical, or square, and they can be arranged in slightly different orientations. The halftone pattern also describes how the size of the dot is increased to cover a larger percentage of the total area and yield darker colors. Different patterns might produce better results for some print jobs.

- **Rotation**

  Lines of halftone dots do not run parallel with the top or side of the paper because that could cause unintended patterns to emerge, resulting in lower quality output.

  In addition, the dots for each of the four colors in a CMYK printer cannot all be printed at the same angle because they would overlap incorrectly and the colors would not appear as intended. Instead, the lines of dots are printed on the page at specific angles so your eye blends them appropriately.

  For example, the black layer of an image might be printed so the lines of dots run across the page at a 45 degree angle to the top of the paper, while the cyan layer is printed so that its lines of dots are at a 105 degree angle to the top of the paper.

Tone transfer curves are most often used to offset the effects of dot gain. *Dot gain* is the tendency for printed dots to be larger than intended, often because of the way ink reacts with paper. If the ink soaks into the paper and spreads out, the resulting dot is much larger (and possibly much lighter in color) than the printer intended it to be. Tone transfer curves can increase or reduce the amount of ink used in proportion to the dot gain.

File Size

Color print jobs can have a file size that is much larger than black and white print jobs. The larger file size can lead to longer processing times and increased traffic on your network.

Because color images must contain data about each layer of color, the file might contain three to four times more information than a grayscale file and over 24 times more information than a black and white file. In addition, ICC profiles are embedded in some file types (such as EPS, JPEG, and TIFF images). While ICC profiles by themselves might not be very large, they do increase the size of an image. If you only have one image repeated throughout a print job, and if you construct your job so the image is downloaded only once, the embedded profile is of little concern.

However, if you use a variety of different images, each with an embedded profile, or if you construct your print job so that each image is downloaded every time it appears, the embedded profiles can add unnecessary volume to the print job. If you plan to use a wide variety of color images, create or save

them with the same color space so they all use the same ICC profile. You can also install color images in a resource library so they can be reused.

Grayscale Printing Concepts

With grayscale printing you can reproduce color images as high-quality black and white images by using many shades of gray to represent subtle variations in color and light. Printing solutions that produce high-quality grayscale output use color printing concepts with a black and white printer that supports them to achieve that effect.

Moving to grayscale printing might be a first step in a migration to full color printing. You can start to create color print jobs and print them on an existing printer until you are ready to invest in color printers. In addition, you can use a grayscale printer as a backup system for a full color printer.

Some color concepts are much less important in grayscale printing than they are in color printing:

- The gamut of a black and white printer is much smaller than that of a color printer; essentially all the colors in an image must be adjusted.
- The rendering intent you choose has little effect on the appearance of the image because the colors are already being changed significantly.
- Page registration is less important. Because the printer only uses one color, you do not have to line up the color planes to create the correct color.
- Paper characteristics have minimal effect on grayscale output; one output profile is usually adequate for all types of paper.

Other color concepts are more essential to grayscale printing.

Color spaces and ICC profiles

The color space of a black and white printer is much smaller than that of a color printer. Even so, printers that can print grayscale images have output ICC profiles, just like color printers. The ICC profiles for black and white printers map colors from the profile connection space (PCS) to shades of gray. Otherwise, the color conversion process is the same.

The print job should specify the appropriate input profile; if there is no input ICC profile, the printer uses a reasonable default. The printer has its own default ICC profile installed and available; it should be adequate for nearly all print jobs.

Halftones

Grayscale printers apply halftones to print jobs to print them; halftones let the printer produce many shades of gray and high-quality images. Generally, the most important characteristic to consider for halftones in grayscale printing is line screen frequency, expressed in lines per inch (lpi). Each printer supports a set of line screen frequencies natively; when you specify the desired line screen frequency in a print job, the printer chooses the available line screen frequency that best matches it.

Tone transfer curves

Tone transfer curves are used in grayscale printing to adjust the amount of toner that is used at different levels of gray, thus adjusting the appearance of images. You can use the appearance value of a tone

transfer curve in grayscale printing to indicate how much the tone transfer curve should adjust the color values. Some sample appearance values could be:

- Dark
- Highlight Midtone
- Standard

Color Management

Images, graphics, and photographs often appear different depending on the monitor or the printer you use. The colors printed by one printer might not match the colors printed on another printer, even if they came from the same source. If it is important that colors stay consistent from camera, scanner, or monitor to printer, you must use *color management* practices.

It is virtually impossible to accurately reproduce the colors that you see on your monitor on a printed page. Because printers typically have smaller color gamuts than other devices, some of the colors must always be adjusted when images are transformed for printing. With color management, you can control the adjustments so they are less noticeable than they might be if you use the default settings of your image creation software, print server, and printer.

Several factors play significant roles in color management, including ICC profiles, rendering intents, and paper characteristics.

ICC Profiles

The International Color Consortium (ICC) is an organization that has established open standards for color management. These standards help products work together by identifying a device-independent color space and defining the elements of an ICC profile.

The device-independent color space that the ICC defined is called the profile connection space (PCS). The PCS is a color space large enough to include all the color gamuts of different input, display, and output devices. An ICC profile contains methods that map the colors that a device can create or display to the values of the corresponding colors in the PCS. The ICC profile can be used to convert an image from a device-specific color space to the PCS, or from the PCS to a device-specific color space.

Product manufacturers create ICC profiles that you can use with their devices. For example, if you take a photograph with a digital camera, you can associate the photograph with the ICC profile for your camera. Then, when you want to print that photograph, the color management system converts the color data from the camera into the PCS. The printer then uses its ICC profile to convert the photograph data from the PCS into its color space, and prints the photograph as accurately as it can.

For more information about the ICC, ICC profiles, and the PCS, see the ICC website:

`http://www.color.org`

To visit this website, make sure you are on a system that is connected to the Internet.

Rendering Intents

Rendering intents indicate what you want a printer to do with colors that are outside its gamut.

ICC profiles support these rendering intents:

- **Perceptual**
  If an image includes any colors that are out-of-gamut for the printer, the printer adjusts all the colors in the image, even those that are already in the gamut of the printer, so they are all in-gamut and maintain their color relationships to each other. The result is an image that is visually pleasing, but is not colorimetrically accurate. The perceptual rendering intent is useful for general reproduction of images, particularly photographs.
- **Saturation**
  If a print job includes colors that are out-of-gamut for the printer, the printer replaces the out-of-gamut color with the nearest color in the gamut. It also adjusts the in-gamut colors so that they are more vivid. Saturation is the least used rendering intent, but it is useful for business graphics, such as images that contain charts or diagrams.
- **Relative colorimetric**
  If a print job includes colors that are out-of-gamut for the printer, the printer substitutes the nearest in-gamut color; in-gamut colors are not adjusted. Colors printed on papers with different media white points might not match visually. The *media white point* is the color of the paper that the print job is printed on. For example, if you print an image on white paper, on off-white paper, and on blue paper by using the relative colorimetric rendering intent, the printer uses the same amount of ink or toner for each one and the resulting color is technically the same. However, the images might seem different because your eyes adjust to the color of the background and interpret the color differently. This rendering intent is typically used for vector graphics.
- **Absolute colorimetric**
  All colors are mapped by using the same method as the relative colorimetric rendering intent, however, all colors are adjusted for the media white point. For example, if you print an image on white paper, on off-white paper, and on blue paper by using the absolute colorimetric rendering intent, the printer adjusts the ink or toner used for each one. The resulting color is technically not same, but the images might look the same because of the way your eyes interpret them in relationship to the color of the paper. The absolute colorimetric rendering intent is typically used for logos.

Paper Characteristics

The paper that you use has a significant impact on the colors that you see. Even if you use the same ICC profile and the same printer, printing on a different paper can result in a very different color appearance.

Colors can change from paper to paper, particularly if you change from coated to uncoated paper or from sheet-fed to continuous forms paper. The changes can be so noticeable that printer manufacturers generally test and certify papers with certain characteristics for use with their printers. They also create different ICC profiles for their printers based on paper characteristics. Some ICC profiles can be used for groups of papers that have similar characteristics.

When you load paper, you set certain paper characteristics on the printer. When the printer chooses the correct device-specific output profile to use, it takes the characteristics into consideration. The paper characteristics are:

- **Media brightness**
  The percentage of light that the paper reflects
- **Media color**
  The color of the paper
- **Media finish**
  The characteristics of the surface of the paper, such as: glossy, satin, matte
- **Media weight**
  The basic weight of the paper

AFP Color Management

There are various ways to print color data with Advanced Function Presentation (AFP). However, to implement an AFP color printing solution with full color management, you must use color management resources (CMRs). We also recommend that you install all your color images as data objects and associate CMRs with them.

Color Management Resources

Color management resources (CMRs) are the foundation of color management in AFP print systems. They are AFP resources that provide all the color management information, such as ICC profiles and halftones, that an AFP system needs to process a print job and maintain consistent color from one device to another.

CMRs share some characteristics with other AFP resources, but are different in some important ways.

CMRs are similar to other AFP resources in these ways:

- CMRs can be associated with elements of a print job at various levels of the hierarchy.
  Typical hierarchy rules apply, so CMRs specified at lower levels override those at the higher level. For example, a CMR set on a data object overrides a default CMR set on a print file.
- CMRs can be included in a print job in an inline resource group and referenced in a form definition, page environment, object environment, or an include Object (IOB) structured field.

  Note

  – CMRs can vary in size from several hundred bytes to several megabytes. If your print job uses relatively few CMRs, including them in the print file might not have an impact on the performance of your system. However, if your print job uses more than 10 CMRs, the size of the print job can increase so much that file transfer rates and network traffic are affected.
- CMRs can be stored centrally in a resource library, so you do not need to include them in every print job.
  You can configure all your print servers so they can access the CMRs.

- For the print server to find CMRs, the resource library must be listed in the AFP resource search path on the print server.

CMRs are different from other AFP resources in these ways:

- You cannot copy CMRs into a resource library as you can other AFP resources.
  To store CMRs in a central resource library, you must install them by using an application such as RICOH AFP Resource Installer.
- CMRs and data objects must be stored in resource libraries that have resource access tables (RATs).
  AFP Resource Installer creates the RAT when CMRs and data objects are installed. We recommend that CMRs and data objects be installed in separate resource libraries and that you store resources that do not require RATs (such as form definitions, page definitions, and overlays) in other resource libraries.
- CMRs installed in a resource library can have names longer than 8 characters, and you can use the names in the print data stream.
  These names are created when you install the CMR by using AFP Resource Installer and are UTF-16BE encoded.

Types of CMRs

Different situations call for different types of CMRs. Some CMRs are created by product manufacturers so you can download and use them, while others are created by your printer or other color management software. If you have the appropriate information, you can also create CMRs yourself.

Some CMRs are used to interpret input files (similar to the function performed by ICC input profiles), while others are used to prepare the final print job output for a specific printer (similar to the function performed by ICC output profiles).

Color Conversion CMRs

Color conversion (CC) CMRs are used to convert colors to and from the ICC profile connection space (PCS), a device-independent color space. You can use them to prepare images for color or grayscale printing.

Color conversion CMRs are an essential element of any AFP color management system because they are ICC profiles encapsulated in AFP structures. The AFP structures add information that your color management system can use, but it leaves the ICC profile unaltered.

You can use color conversion CMRs to produce consistent colors on different devices. In a color system, they help ensure that the colors on your monitor are as close as possible to those that are printed. If you move the print job to a different printer, the colors are adjusted again to match the new printer.

In a grayscale system, color conversion CMRs map colors to appropriate shades of gray to produce high-quality black and white images.

Passthrough CMRs are color conversion CMRs that indicate that no color processing should be done if the color space of the presentation device is the same as the color space of the CMR. Passthrough CMRs contain no data.

Link Color Conversion CMRs

Link color conversion CMRs combine the processing information required to convert an image directly from the color space of an input device to the color space of the output device. Essentially, link color conversion CMRs replace a pair of color conversion CMRs.

Converting color images to and from the PCS takes a significant amount of processing resources, in part because the process includes two conversions. Link color conversion CMRs combine the two conversions and make them more efficient. The printer can use the link color conversion CMR to convert colors directly from the color space of the input device to the color space of the output device with the same color fidelity they would have if the printer did both of the conversions. As a result, link color conversion CMRs can improve system performance.

The two types of link color conversion CMRs are:

Link CMRs

Link (LK) CMRs are unique. You cannot create a link CMR yourself and you do not include references to link CMRs in your print jobs. The print system creates and uses link CMRs automatically.

If you use AFP Resource Installer, link CMRs are generated automatically when you create or install a color conversion CMR. As a result, your resource library always contains link CMRs for every combination of color conversion CMRs in audit (input) and instruction (output) processing modes. When link CMRs are created, AFP Resource Installer marks them as *capturable*, so the printer can save them to be used in other print jobs.

If you do not use AFP Resource Installer, some printers can create link CMRs when they process print jobs. Then, the print controller looks at the link CMRs that it has available to find one that combines the audit color conversion CMR with the appropriate instruction color conversion CMR. If it does not find one, the print controller creates the link CMR and uses it. The print controller can be configured to save the link CMRs that it creates. However, the link CMRs are sometimes removed during normal operation, for example, if the printer runs out of storage or is shut down. If the link is removed, the printer must create a new link CMR the next time it is needed.

When a link CMR is created, the print system evaluates the conversion algorithms to and from the PCS. The system then combines the algorithms, so a data object can be converted directly from one color space to the other without actually being converted to the PCS.

Device link CMRs

Device link (DL) CMRs use an ICC device link profile to convert directly from an input color space to an output color space without reference to an audit-mode or instruction-mode CMR. An ICC device link profile is a special kind of ICC profile that is used to convert the input device color space to the color space of an output or display device. ICC device link profiles are not embedded in images.

You can create, install, and uninstall device link CMRs yourself. Device link CMRs are referenced in the MO:DCA data stream and take precedence over audit color conversion CMRs. A device link CMR specifies its own rendering intent, which is indicated in the header of the ICC device link profile. This rendering intent overrides any other rendering intent that is active.

The biggest advantage of using device link CMRs is that they preserve the black channel (K component) of the input color space when converting from CMYK to CMYK.

Halftone CMRs

Halftone (HT) CMRs carry the information that a printer uses to convert print jobs into a pattern of dots that it can put on paper. Halftone CMRs can be used with both color and grayscale print jobs.

Halftone CMRs generally specify the line screen frequency, halftone pattern, and rotation of the halftone that they carry. Some device-specific halftone CMRs also include the printer resolution.

A printer that uses AFP color management to print color or grayscale print jobs must use a halftone CMR to convert the print job into a format that the printer can reproduce in ink or toner. If a halftone CMR is not specified in the print job, the printer applies a default halftone CMR.

You can associate device-specific halftone CMRs or generic halftone CMRs with print jobs:

- If you know which printer is printing the job, you can associate a device-specific halftone CMR with the print job (or with AFP resources inside the print job). The printer uses the halftone CMR that you specify.
- If you do not know which printer is printing the job, but you want to ensure that it uses a halftone CMR that has certain characteristics, such as a specific line screen frequency, you can associate a generic halftone CMR with the print job.

Because it is difficult to know which halftone CMRs should be used for the current conditions on the current printer, we recommend that you specify halftone CMRs generically and let the printer choose the most appropriate CMR that it has available.

Generic halftone CMRs

You can use generic halftone CMRs when you want to choose one or more characteristics of the halftone CMR for a print job, but you do not know exactly which halftone CMRs are available.

When a print job specifies a generic halftone CMR, the print server looks in the resource library for halftone CMRs that match the printer device type and model. If the print server finds an appropriate CMR, it sends the device-specific halftone CMR to the printer with the print job. If the print server does not find an appropriate halftone CMR, it sends the generic halftone CMR to the printer.

If a print job arrives at the printer requesting a generic halftone CMR, the printer compares the requested characteristics with the available device-specific halftone CMRs. If there is a match, the printer uses the selected device-specific halftone CMR when it processes the print job. If there is no match, the printer uses the halftone CMR whose line screen frequency value is closest to the one requested.

The Color Management Object Content Architecture (CMOCA) has defined a variety of generic halftone CMRs, which cover the most common line screen frequencies and halftone types. A print server that supports CMOCA can interpret generic halftone CMRs if it has device-specific halftone CMRs available to it in a resource library. If you use AFP Resource Installer, the generic halftone CMRs are installed in every resource library that you create and populate by using AFP Resource Installer.

Printers that support CMOCA should be able to interpret those generic CMRs and associate them with device-specific halftone CMRs.

Tone Transfer Curve CMRs

Tone transfer curve (TTC) CMRs are used to carry tone transfer curve information for an AFP print job, so you can modify the values of a particular color component and adjust the appearance of some of the colors by increasing or decreasing the amount of ink used to emphasize or reduce the effects of dot gain on the final output.

Like halftone CMRs, tone transfer curve CMRs are associated with print jobs specifically or generically. If they are specified generically, the print server looks in the resource library for tone transfer curve CMRs that match the printer device type and model. If the print server finds an appropriate CMR, it

sends the device-specific tone transfer curve CMR to the printer with the print job. If the print server does not find an appropriate tone transfer curve CMR, it sends the generic tone transfer curve CMR to the printer.

If a print job arrives at the printer requesting a generic tone transfer curve CMR, the printer compares the requested characteristics with the device-specific tone transfer curve CMRs that it has available. If there is a match, the print server or printer uses the selected device-specific tone transfer curve CMR when it processes the print job. If the printer cannot find a good match for the generic tone transfer curve CMR, it ignores the request and uses its default tone transfer curve CMR.

The Color Management Object Content Architecture (CMOCA) defines several generic tone transfer curve CMRs with different appearance values. You can use the appearance values to specify how to print your job with regard to the reported dot gain of the printer.

Generic tone transfer curves can be used to select these appearance values:

Dark

The output is adjusted to show a dot gain of 33% for a 50% dot.

Accutone

The output is adjusted to show a dot gain of 22% for a 50% dot.

Highlight Midtone

The output is adjusted to show a dot gain of 14% for a 50% dot. This appearance is often used to emphasize the brightest part of an image.

Standard

The output is adjusted just enough to account for the effects of dot gain, effectively counteracting the dot gain.

If you use AFP Resource Installer, it installs the generic tone transfer curve CMRs on your system automatically.

CMR Processing Modes

CMR processing modes tell the print system how to apply a CMR to the print data it is associated with. You specify a CMR processing mode whenever you specify a CMR, although not all modes are valid for all CMR types.

Audit Processing Mode

CMRs with the audit processing mode refer to processing that has already been applied to a resource. In most cases, audit CMRs describe input data and are similar to ICC input profiles.

The audit processing mode is used primarily with color conversion CMRs. In audit processing mode, those CMRs indicate which ICC profile must be applied to convert the data into the profile connection space (PCS).

For example, to take a photograph with a digital camera and then include the photograph in an AFP print job, you can use AFP Resource Installer to:

1. Create a color conversion CMR by using the ICC profile of your camera.
2. Install your photograph in a resource library.

3. Associate the color conversion CMR with the data object, indicating the audit processing mode.

Then, you create a print job that includes the data object. When processing the print job, the system uses the color conversion CMR to convert the colors in the image into the PCS. The colors can then be converted into the color space of the printer that is printing it.

Instruction Processing Mode

CMRs with the instruction processing mode refer to processing that is done to prepare the resource for a specific printer that uses a certain paper or another device. Generally, instruction CMRs refer to output data and are similar to ICC output profiles.

The instruction processing mode is used with color conversion, tone transfer curve, and halftone CMRs. In instruction processing mode, these CMRs indicate how the system must convert a resource so it prints correctly on the target printer. The manufacturer of your printer should provide ICC profiles or a variety of CMRs that you can use. Those ICC profiles and CMRs might be installed in the printer controller, included with the printer on a CD, or available for download from the manufacturer's website.

If you send a color AFP print job to a printer that supports AFP Color Management, color conversion and tone transfer curve CMRs in instruction processing mode can be associated with the job. When the printer processes the print job, it applies the CMRs in this order:

1. Color conversion CMRs in audit processing mode to convert the resources into the ICC profile connection space (PCS).
2. Color conversion and tone transfer curve CMRs in instruction processing mode to convert the resources into the color space of the printer.
3. Halftone CMR in instruction processing mode to convert the job pages from their digital format into the pattern of dots that the printer can produce.

In some cases, CMRs that are usually used as instruction CMRs can be used as audit CMRs. For example, if you send a very large print job to a high-speed printer, the images in the print job are converted into the color space of that printer by using a color conversion CMR with the instruction processing mode. However, if you have to reprint part of the job on a different printer, the system must convert the print job into the color space of the second printer. In that case, the color conversion CMR of the first printer is used in the audit processing mode to move the images back into the PCS. Then, the system uses a color conversion CMR of the second printer in instruction mode to convert the images into its color space.

Link Processing Mode

CMRs with the link processing mode are used to link an input color space in the presentation data (sometimes defined by an audit CMR) to the output color space of the presentation device (sometimes defined by an instruction CMR). Only link (LK) and device link (DL) CMRs can be used in link processing mode.

Whenever you install or uninstall audit or instruction color conversion CMRs in your resource library by using AFP Resource Installer or a similar software product, the AFP Resource Installer automatically creates or deletes link (LK) CMRs for every combination of audit and instruction color conversion CMR.

When a print job calls for a given audit-instruction combination, the print server checks the resource library for a link (LK) CMR for that combination. If the print server finds an appropriate link CMR, it sends the CMR to the printer with the print job. Your printer can use the link (LK) CMRs whenever a print job indicates that it uses a particular combination of audit and instruction CMRs.

If you do not use AFP Resource Installer or a similar program to install your resources, your color printer must either create link (LK) CMRs while it processes your print jobs or convert the colors in your jobs twice, first from the original color space to the PCS and then from the PCS to the color space of the printer.

CMR Creation and Installation

Device manufacturers and groups that support AFP color standards create CMRs that you can use in your color printing systems. You can also create CMRs yourself, based on your needs.

The AFP Consortium, the group that defined the AFP Color Management Object Content Architecture (CMOCA), identified a set of color conversion CMRs that are most often used in audit processing mode. The set includes color conversion CMRs for common color spaces, such as:

- Adobe RGB (1998)
- sRGB
- SMPTE-C RGB
- SWOP CMYK

The standard CMRs are included with AFP Resource Installer, although they are not installed by default. You can install the standard CMRs that you plan to use. In addition, AFP Resource Installer automatically installs all the generic halftone and tone transfer curve CMRs in any resource library you create.

If you need more CMRs, you can create them by using wizards provided in AFP Resource Installer. See the online help for details about the wizard.

If you use AFP Resource Installer to create a CMR, the software automatically installs the CMR in a resource library. You can also use AFP Resource Installer to install CMRs that you get from your printer manufacturer.

Data Objects

Presentation data objects contain a single type of data (such as GIF, JPEG, and TIFF images) and can be used in your print jobs. These data objects can be placed directly in a page or overlay or can be defined as resources and included in pages or overlays. Using a data object as a resource is more efficient when that object appears more than once in a print job; resources are downloaded to the printer just once and referenced as needed.

Data objects can either be included inline with a print job or installed in a resource library by using software such as AFP Resource Installer. If you install your data objects in a resource library, you can associate color conversion CMRs with them.

Types of Data Objects

Image data objects can be stored in a number of different formats, including AFPC JPEG Subset, EPS, GIF, IOCA, PDF, PNG, and TIFF. These image types are device-independent so they can be used by different systems and still be interpreted consistently.

- AFPC JPEG Subset (JPEG)
  AFPC (AFP Consortium) JPEG Subset files, formerly called JPEG File Interchange Format (JFIF) files, are bitmap image files that are compressed by using Joint Photographic Experts Group (JPEG)

compression. As a result, AFPC JPEG Subset files are most commonly referred to as JPEG files. JPEG files most commonly use the file extension .jpg, but can also use .jpeg, .jpe, .jfif, and .jif.

JPEG compression deletes information that it considers unnecessary from images when it converts them. JPEG files vary from having small amounts of compression to having large amounts of compression. The more an image is compressed, the more information is lost. If the image is compressed only once, there usually is no noticeable effect on the image. However, if the image is compressed and decompressed repeatedly, the effects of deleting information become more noticeable.

JPEG compression is commonly used for photographs, especially photographs that are transmitted or displayed on webpages. The compression makes the files small enough to transmit on a network efficiently, but leaves enough information that the image is still visually appealing.

- Encapsulated PostScript (EPS)

  EPS is a PostScript graphics file format that follows conventions that Adobe Systems defined. EPS files support embedded ICC profiles.

- Graphics Interchange Format (GIF)

  GIF files are bitmap image files that are limited to a palette of 256 RGB colors. Because of the limited color range that it can contain, GIF is not a good format for reproducing photographs, but it is generally adequate for logos or charts. GIF images are widely used on the Internet because they are usually smaller than other image formats. GIF files use the file extension .gif.

- Image Object Content Architecture (IOCA)

  IOCA is an architecture that provides a consistent way to represent images, including conventions and directions for processing and exchanging image information. The architecture defines image information independently of all data objects and environments in which it might exist and uses self-identifying terms; each field contains a description of itself along with its contents.

- Portable Document Format (PDF)

  PDF is a standard file format that Adobe Systems developed.

  PDF files can be used and stored on various operating systems and contain all the required image and font data. Design attributes in a PDF are kept in a single compressed package.

  Note

  – Single-page and multiple-page PDF files can be used as data objects in AFP print jobs.

- Portable Network Graphics (PNG)

  PNG files are bitmap image files that support indexed colors, palette-based images with 24-bit RGB or 32-bit RGBA colors, grayscale images, an optional alpha channel, and lossless compression. PNG is used for transferring images on the Internet, but not for print graphics. PNG files use the file extension .png.

- Tagged Image File Format (TIFF)

  TIFF files are bitmap image files that include headers to provide more information about the image. TIFF files use the file extensions .tif or .tiff.

  TIFF files support embedded ICC profiles. If an ICC profile is embedded in a file, the characteristics of the input color space are known whenever the file is used; however, the profiles increase the file size. When you save a file in the TIFF format, you can use various compression algorithms.

  Note

  – Single-image and multiple-image TIFF files can be used as data objects in AFP print jobs.

Not all printers support all types of data objects.

The embedded ICC profiles in EPS, JPEG, and TIFF files contain the information that a printer uses to convert colors in the image from an input color space into the profile connection space (PCS). The input color space is either an industry-standard space or a custom space that describes the color reproduction capabilities of a device, such as a scanner, digital camera, monitor, or printer.

Data Object Creation and Installation

You can use a wide variety of software applications to create or manipulate images to include in print jobs. If you want to store them in central resource repositories, you can use AFP Resource Installer to install them.

Data object creation

Most types of data objects are images of some kind. Examples include: photographs taken with a digital camera; charts or diagrams generated by a software tool; and digital drawings created using graphics software. Regardless of how images are created, you generally need to manipulate them to include them in print jobs.

The changes include:

- Convert the image into a file type that is appropriate for printing. For example, the file types that many graphics applications (such as Adobe Illustrator, CorelDRAW, and Corel Paint Shop Pro) use to store images while you work on them are not appropriate for printing. To use images that you create from any of those programs, you can save or export those files as a different file type, such as EPS, JPEG, or TIFF.
- Make sure that your image files are associated with an appropriate color space or input profile. Follow the instructions provided with your graphics software to set up color management, including installing and using ICC profiles for digital cameras and monitors, and customizing color management settings. The instructions should also explain how to change the color profile that an image uses and how to save an image with an embedded profile.
- Follow the tips and best practices provided in the other sections below for creating images and managing them as data object resources.

Data object installation

You can use AFP Resource Installer to install your images in a resource library. AFP Resource Installer includes wizards that can guide you through the process of installing an image as a data object. When you install an EPS, JPEG, or TIFF image with an embedded ICC profile by using AFP Resource Installer, you can choose how you want to handle the profile:

- Leave the profile in the file without creating a CMR.
- Leave the profile in the file, but also copy the profile and create a CMR from the copy. Associate the new CMR with the data object.
- To reduce the file size, remove the profile from the file and make the profile into a CMR. Associate the new CMR with the data object.

Resource Library Management

If you store CMRs and data objects in central resource libraries, you must understand some of the characteristics of resource libraries to make sure that your resources are available when and where you need them.

Resource libraries that AFP Resource Installer creates use a *resource access table* (RAT) as the index of the resource library. The index is stored as a file in the library that it refers to. You must store CMRs in resource libraries that use a RAT. We recommend that you store data objects in resource libraries that use a RAT as well.

When you use AFP Resource Installer to create a resource library, it creates a RAT and stores it in the library. When you install a CMR or data object, AFP Resource Installer updates the RAT with information about the resource. When a print server looks in a resource library for a resource, it first looks in the RAT to see if the resource is listed.

The print server relies on the RAT; if it is incorrect, the print server cannot find resources in the resource library. As a result, you must always use AFP Resource Installer to manage your resource libraries, including to:

- Add CMRs and data objects to a resource library.

  Do not copy CMRs or data objects directly into the resource libraries that AFP Resource Installer uses. If you copy CMRs or data objects into these resource libraries, the RAT is not updated so the print server cannot use it to find the CMRs or data objects.
- Modify properties of data objects and CMRs listed in the RAT.

  Do not directly edit the RAT or any of the files in a resource library. Do not replace an existing version of a CMR or data object with a new version by copying the new version directly into the resource library; use AFP Resource Installer to update the resource.
- Install CMRs or data objects in a different resource library or replicate a resource library in a different location.

  Do not copy CMRs or data objects from a resource library and store them in another location.

For more information about completing these tasks, see the AFP Resource Installer online help.

Tips and Best Practices

These general guidelines about creating and managing images and other color resources can improve the performance of your AFP color printing system.

Tips for images

To optimize the performance of your AFP color printing system, we recommend that you follow some guidelines for creating and including images in print jobs.

When you want to use color images in your print jobs:

- Get the original electronic versions of images instead of scanning existing documents.

  Almost unnoticeable specks of color in the background of images that have been scanned can greatly increase the size of the image. If you must scan an image, use an image editing tool to clean up the background as much as possible.

- Save all images in the same standard color space so you only need one input profile for all of them.
  Adobe RGB (1998) is the recommended color space for images that are to be printed.
- Flatten multi-layer images (such as the ones you can create in graphics tools like Adobe Illustrator and Corel Paint Shop Pro) before including them in print jobs.
  Unflattened images are extremely large and more difficult to work with. Save a copy of the original image for future editing, but flatten the version that you include in your print job.

Tips for Resources

To optimize the performance of your AFP color printing system, we recommend that you follow some guidelines for managing color resources.

You can use AFP Resource Installer to:

- Install all the CMRs for your printer in a resource library.
- Install the data objects that you use frequently in a resource library.
- Mark the CMRs and data objects that are reused regularly as non-private, capturable resources so they can be saved on the printer and used for other print jobs without being downloaded every time.

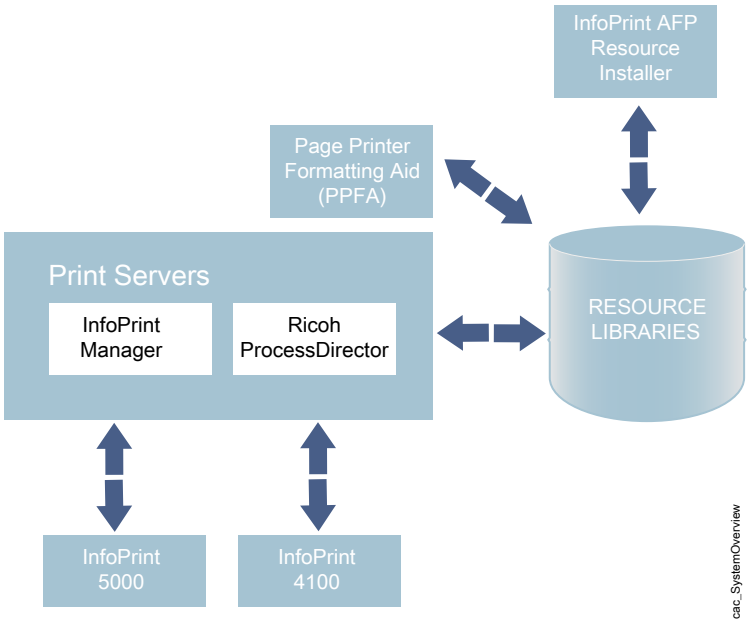
  Note

  – This option is not advisable for secure resources, such as signature files.
- Install CMRs and data objects in resource libraries that the print server can access, so they only need to be stored in one place and can be used by all print servers.
- Associate audit color conversion CMRs with data objects that require color management, so the embedded profiles can be removed from the image files.

RICOH AFP color and grayscale products

Ricoh provides a variety of products that support AFP color and grayscale printing. They can be used in various combinations to create color workflow solutions.

The RICOH products that support AFP color management



Printers

Some RICOH printers support color or grayscale printing with full AFP color management, while others support AFP color printing without color management.

InfoPrint 5000

The InfoPrint 5000 is a full-color, high-speed, continuous forms Intelligent Printer Data Stream (IPDS) printer that supports AFP color management.

The InfoPrint 5000 uses piezo-electric drop-on-demand inkjet technology with water-based pigment inks that are designed to resist fading and smearing.

The InfoPrint 5000 receives AFP print jobs from RICOH print servers. When used as part of a full AFP color management system, the InfoPrint 5000 can receive color conversion, link, and tone transfer curve CMRs and apply them to print jobs. Because the printer engine applies its own halftones, it does not support halftone CMRs.

The InfoPrint 5000 can capture resources and store them in a repository in its controller so they can be reused in other print jobs. The printer only captures resources that are marked non-private and capturable using an application such as AFP Resource Installer.

You can also submit PostScript and Portable Document Format (PDF) jobs to hot folders on the InfoPrint 5000, but those print jobs are not processed using AFP color management.

InfoPrint 4100

The InfoPrint 4100 family of printers uses a laser, electrophotographic print technology and Advanced Function Presentation (AFP) licensed programs to create high-quality text and graphics output. InfoPrint 4100 printers with the InfoPrint POWER architecture and microcode release 15.4 or later support the AFP color management architecture and can be used to print very high-quality grayscale images.

The base level of AFP color management support in InfoPrint 4100 printers with the InfoPrint POWER architecture and microcode release 15.4 or later lets you apply color management to grayscale text, graphics (GOCA), bi-level images (IOCA FS10), and barcodes. To use color management functions with other types of objects, the AFP Color Emulation feature is required.

The AFP Color Emulation feature lets you print color objects in grayscale. When the feature is installed, AFP Color Management lets you natively print GIF, IOCA FS11, IOCA FS45, JPEG, and TIFF images in high-quality grayscale.

Note

- Single-page PDF and EPS data objects are not supported on the InfoPrint 4100.

InfoPrint 4100 printers support these types of CMRs:

- Color conversion
- Link color conversion
- Tone transfer curve
- Halftone

For even better images on InfoPrint 4100 Models MS1, MD1, and MD2, you can also install the Image Enhancement feature. Benefits of this feature include:

- Enhanced fusing technology to minimize toner distortion
- Optimized developer unit for high coverage and high-density applications
- Improved halftones tailored to new hardware and toner technology

RICOH AFP Resource Installer

RICOH AFP Resource Installer is a key element of an AFP color management system when resources are stored in central libraries. You can use it to create, install, and manage color management resources (CMRs) and data objects for use in your system.

AFP Resource Installer is a Java application that you install on a Windows workstation. You can use it to install and work with fonts in addition to CMRs and data objects.

You can use AFP Resource Installer to:

- Create CMRs from existing data, including ICC profiles.
  You can use a wizard to guide you through the process.
- Install CMRs, fonts, and data objects in resource libraries on the local system or on any system that you can access with FTP.
- Associate CMRs with data objects, so data objects can be reproduced accurately on different printers.
  In some cases, you can reduce the file size of your images by removing the embedded color profile from the file and using an associated CMR.
- Mark resources as capturable.

Capturable resources can be captured and saved in the printer for use with other print jobs, which can help improve system performance. The print server queries the printer before it sends any resources; if the printer already has the resource, the print server does not have to send it.

- Mark resources as private.
  Private resources cannot be captured in the printer and must be downloaded with every print job that uses them. For example, you can mark signature files used for company checks as private for security reasons.

When you use AFP Resource Installer to install a color conversion CMR, the software automatically creates link (LK) CMRs between the new color conversion CMR and the existing color conversion CMRs. When a print file references the new CMR, the print server automatically downloads the link CMRs that match the target device type and model and sends them to the printer with the print job. If one of those link CMRs is appropriate, the printer can use it instead of having to spend extra time creating a link CMR.

To let a print server use resources installed by AFP Resource Installer, you must add the path to the resource libraries to the AFP resource path in the server.

Print servers

RICOH print servers receive print jobs from various sources and prepare them to be sent to a printer. After print jobs have been prepared, the print server interacts with the printer to make sure that all the required resources are available, then sends the print job data to be printed.

Several different print servers are available. Although they provide much of the same function, they are appropriate for different environments.

RICOH InfoPrint Manager

RICOH InfoPrint Manager is a flexible and scalable print management solution for AIX, Linux, or Windows that provides many choices in expanding and managing your print environment. RICOH InfoPrint Manager can process print jobs that contain references to color management resources and interact with centralized resource libraries.

When RICOH InfoPrint Manager receives AFP or line data print jobs, it processes CMRs and data objects similar to the way it processes other AFP resources. You can add the resource libraries that you create using AFP Resource Installer to the Location of presentation object containers property (also called the resource-context-presentation-object-container attribute) of your actual destinations. Then, RICOH InfoPrint Manager can search those resource libraries to find data objects and CMRs when print jobs request them.

Enabling capture for CMRs

If the jobs that run through your system use the same CMRs, you might want to enable your actual destination to capture inline CMRs.

These inline CMRs must have an object identified (OID) so RICOH InfoPrint Manager can capture them.

From the **RICOH InfoPrint Manager Administration GUI**:

1. Click on the actual destination and select **Printer** → **Properties**.
2. From the actual destination's **Printer Properties** notebook, click the **Tuning** tab.

3. Find the **Capture inline CMR resources** radio button.

   If you can't see the **Capture inline CMR resources** radio button, click **Show more**.

4. Scroll to the bottom of the screen and click the **Yes** radio button for the **Capture inline CMR resources** setting.

Page Printer Formatting Aid (PPFA)

PPFA is a feature of RICOH InfoPrint Manager that lets you create form definitions and page definitions for use with your AFP print jobs. You can use PPFA to associate CMRs with form definitions and page definitions for your color print jobs.

The form definitions and page definitions that you create using PPFA can be used in print jobs that are sent to RICOH InfoPrint Manager and RICOH ProcessDirector.

RICOH ProcessDirector

RICOH ProcessDirector is a database-driven print workflow system that lets you manage all aspects of your printing process. The server runs on Linux or Windows systems and is accessed using a web browser-based interface. RICOH ProcessDirector can receive and process AFP print jobs that include AFP color management objects.

RICOH ProcessDirector can also receive line data print jobs that refer to CMRs and data objects and convert them into AFP using a step based on the **ConvertLineDataJobIntoAFP** step template.

RICOH ProcessDirector processes CMRs and data objects similar to the way it processes other AFP resources. You can add the resource libraries that you create using AFP Resource Installer to the AFP resource path property on a print job or in the job defaults for your workflows. Then, RICOH ProcessDirector can search those resource libraries to find data objects and CMRs when print jobs request them.

AFP color solution scenarios

AFP color solutions from Ricoh can be assembled in a variety of configurations based on your environment and needs.

Printing high-quality grayscale output on an InfoPrint 4100 printer

An insurance company wants to migrate to AFP color printing slowly, so they start the process by using AFP color management to print high-quality grayscale output on their InfoPrint 4100 printers.

The insurance company runs an in-house print shop that has four duplex InfoPrint 4100 lines. They use a document composition tool to create AFP print jobs and submit the jobs to InfoPrint Manager for Windows, which they use to balance the printing loads and keep all the printers running as much as possible.

They have decided that they want to add images to some of their output. To start, they want to add photographs of their insurance agents to the letterhead on letters to their clients. They are not ready to move into full color printing, so they want to print black and white images using their existing printers. Unfortunately, they have not been satisfied with the appearance of the images; they are too black and one-dimensional, so they look unprofessional.

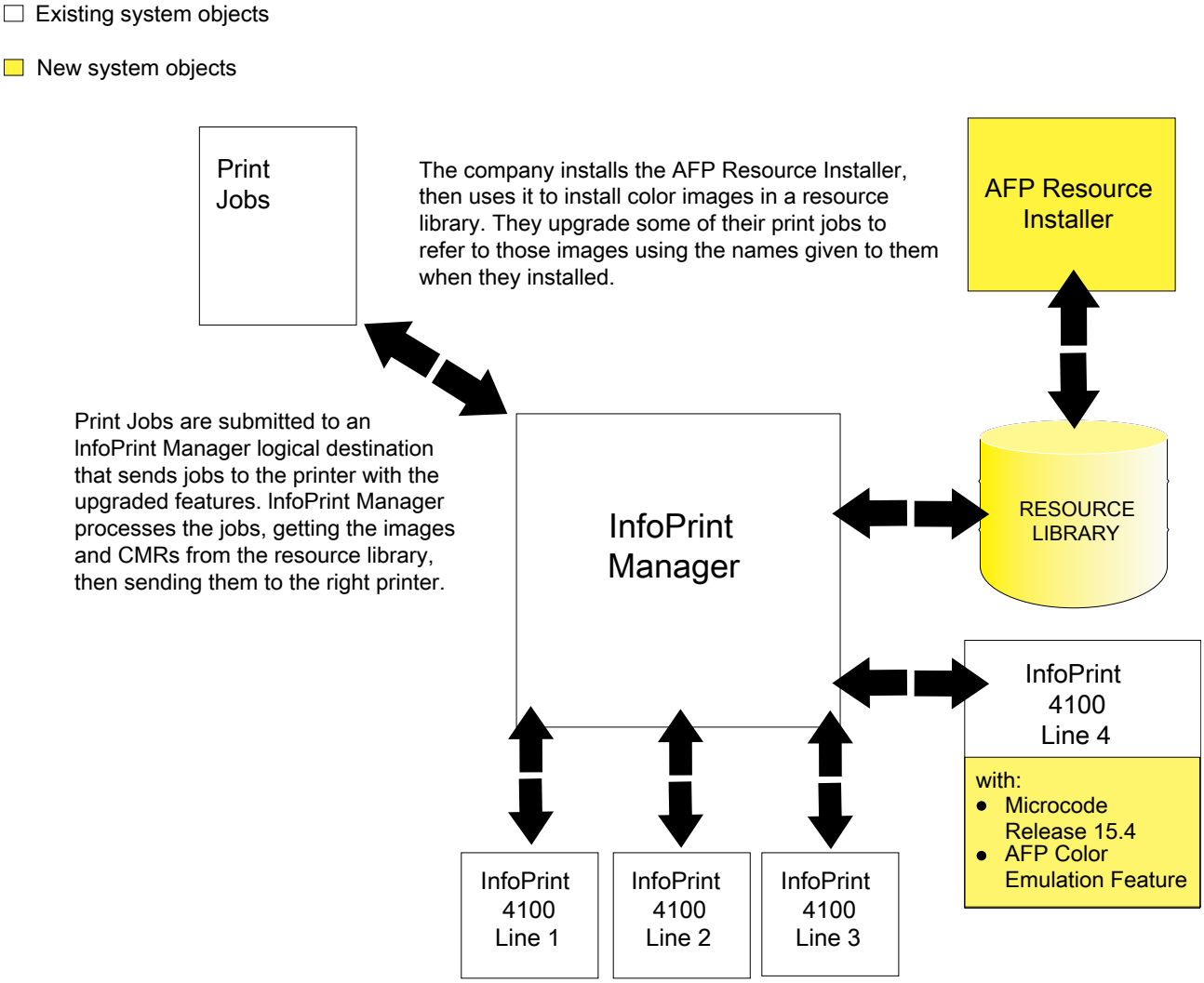
The solution

The Ricoh team suggests these changes to improve image quality:

- Upgrade the printer to use a controller with the InfoPrint POWER architecture with microcode release 15.4 or later.

  This update adds support for the AFP color management, which includes high-quality grayscale printing using halftones and tone transfer curves.
- Purchase and install the AFP Color Emulation feature for the printer.
- Use AFP Resource Installer to manage the images, including associating them with the correct halftones and tone transfer curves to create the best quality output.

Solution diagram for adding AFP Resource Installer and a resource library, and updating one InfoPrint 4100 printer line to use microcode release 15.4 and the AFP Color Emulation Feature.



Implementation

To implement the solution, the print shop works with their Ricoh representatives to upgrade one of their print lines to try the new function. They:

- Upgrade the print controller to the microcode release 15.4 or later.
- Install the AFP Color Emulation feature on the printer controller.
- Install AFP Resource Installer.
- Use AFP Resource Installer to:
  - Create and install the color conversion CMRs for the InfoPrint 4100 printer and for the digital camera used to take the photographs.
  - Choose the generic halftone and tone transfer curve CMRs to use with the InfoPrint 4100 based on the line screen frequency and appearance values that you want to use.
  - Install the photographs of the insurance agents in a resource library, then associate them with the appropriate CMRs.
- Update the RICOH InfoPrint Manager destinations so they send the print jobs that contain photographs to the printer line that has the AFP Color Emulation feature installed.
- Create print jobs that call the images using the names that they were given when they were installed.

Replacing pre-printed forms

A bank wants to reduce the amount of paper that it stores. If the bank eliminates some of the pre-printed forms that they use by printing some statements on a color printer, they can use the same type of plain paper for various applications.

Over the last five years, the bank has purchased ten smaller banks and is in negotiations for several others. The parent bank uses an AFP system to print statements in-house on pre-printed forms. All the pre-printed forms contain color logos. Some of them also contain background images or blocks of color that divide the page into regions. The existing system consists of:

- RICOH InfoPrint Manager, including the Page Printer Formatting Aid (PPFA) feature
- Three duplex InfoPrint 4000 lines
- Two duplex InfoPrint 4100 lines

Print jobs are submitted as line data and RICOH InfoPrint Manager transforms them into AFP. AFP resources are stored in a central location. There are no color production printers in use.

The banks that were acquired handled printing in different ways. Some of them had their own print shops; some of them outsourced print to other companies. The print staff at the parent bank has spent a significant amount of time moving the printing operations of the acquired banks to their in-house print operation, starting with the ones that outsourced their print. Five of them have been converted. The print shop administrators are starting to face some interesting issues:

- The increased workload for the in-house operation means that they need to increase their print capacity.
- Each of the banks that the parent has purchased has maintained its own logo and branding. Each time one of the banks has been integrated with print operations, it has brought with it at least five different pre-printed forms that need to be ordered and stored.
- Even though each kind of pre-printed form comes from the same vendor, the colors vary noticeably.
- The acquired banks will eventually replace their old logos and forms with the logo and forms of the parent bank, but the dates are not yet planned.

With the increase in the number of forms that must be stored, space is becoming a problem. The team knows that they have to expand printing capacity, but they do not have space for a new duplex line. If they could eliminate some of the paper that they have to store, they could use that space for the new line.

The solution

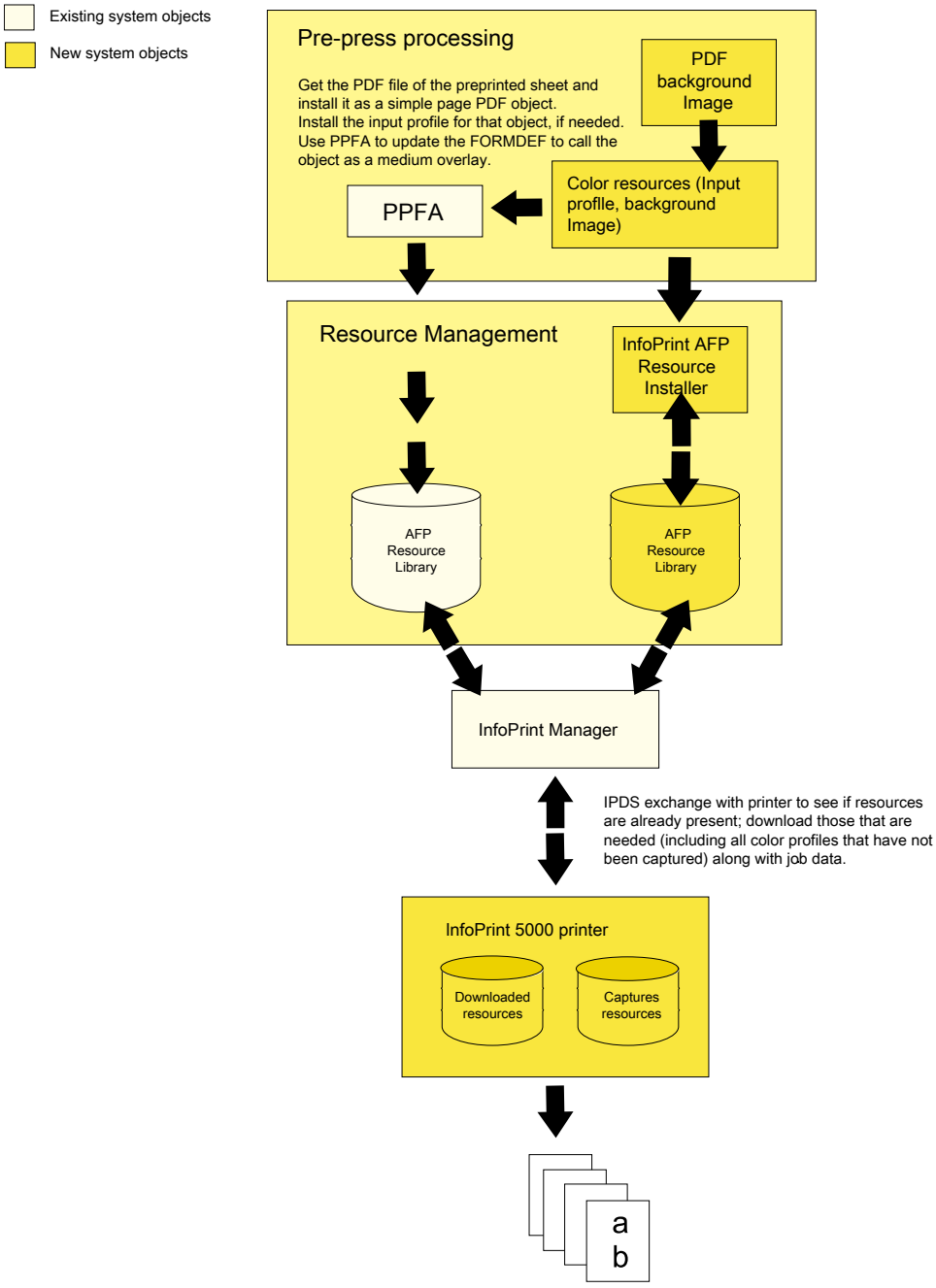
The Ricoh team proposes an AFP color solution to:

- Transform their pre-printed forms to simple color statements and eliminate the need for pre-printed forms, thus reducing the paper that needs to be stored.
- Streamline the process of moving the acquired banks to the parent bank's logo and forms.
- Optimize resource management to maximize throughput.
- Make the colors more uniform from job to job.

The solution adds a duplex InfoPrint 5000 line and AFP Resource Installer to the existing system and updates the other components to add AFP color support. The bank can choose which applications to move to the color workflow, taking into consideration the fact that the throughput of the InfoPrint 5000 line is less than that of an InfoPrint 4100 line. In addition, those print jobs cannot be run on the existing InfoPrint 4100 or InfoPrint 4000 printers, since those printers cannot print the required color elements.

As the printing staff becomes proficient in using the system, they can start to prepare more applications for an eventual move to color by migrating them to use the high-quality grayscale function available on their InfoPrint 4100 printers. Then, when they are ready to add another color line (either by eliminating more paper storage or by replacing one of the InfoPrint 4000 lines), they can move those print jobs to the color printers.

Solution diagram for adding RICOH AFP Resource Installer and an InfoPrint 5000 printer



Implementation

To implement the color solution, the bank printing staff works with the Ricoh team through several phases:

- **Planning the color solution**
  - Determine where the resource library should be located so that it is accessible to all the applications that need to use it.

  - Determine which types of resources can or should be saved on the printer for reuse. For example, signature files should probably not be saved on the printer; company logos and background images probably should be saved.

- **Installing and configuring the system**
  - Install the InfoPrint 5000 printers.
  - Apply service updates to RICOH InfoPrint Manager and PPFA to add color support.
  - Install AFP Resource Installer.
- **Configuring resource management for optimal performance**
  - Use AFP Resource Installer to install the CMRs for the InfoPrint 5000 printers.
  - Get the PDF file of the pre-printed form or use graphic art software to create a full page image to use as the background image.
  - Use AFP Resource Installer to install the PDF or image as a data object resource and associate a color conversion CMR with it.
  - Configure RICOH InfoPrint Manager to recognize the resource libraries that AFP Resource Installer uses and to send print jobs to the new printer.
  - Use PPFA to create a form definition and page definition that calls the new data object resource. Refer to the object using the ID that you specified when you installed it.
  - Copy any new form definitions or page definitions that you created into an existing resource library.
- **Testing system operations**
  - Create a print job that calls the form definition and page definition, and that points to the new resource libraries.
  - Submit the print job.
  - Check the color output.

Eliminating physical inserts

A print service bureau wants to reduce the number of physical inserts that they have to add to envelopes by printing full color electronic inserts in blank areas on credit card statements.

The print service bureau prints statements and one-to-one advertising brochures for a variety of customers. The customers send their resources and print job data to the service bureau. The pre-press staff uses a document composition tool to create AFP jobs. The AFP print jobs are submitted to RICOH ProcessDirector, which sends them to ten duplex InfoPrint 4100 systems. After the jobs are printed, they go to post-processing to be folded, inserted, and mailed.

The service bureau asks all their customers to submit required resources with every print job; they do not store resources indefinitely. Many of their customers (such as credit card companies) include inserts in the envelope with their statements. Some problems with this arrangement are:

- Inserters often ruin mailpieces, so they frequently have to reprint individual statements. Reprinting adds time and cost to the process.

- Customers tend to ignore the inserts, separating them from the statement immediately after opening the envelope and discarding them.

The service bureau would like expand its offerings to include full-color variable-data printed using an AFP color solution. They also think that they can save time, money, and paper by reducing the complexity of the tasks that their inserters do. If the inserter only has to fold each statement and put it in an envelope (instead of adding two more sheets of paper, folding, and inserting), it has fewer opportunities for ruining mailpieces.

The solution

The service bureau works with the Ricoh team to create a solution based on its existing process and environment that will help their customers move away from using physical inserts for advertising and toward electronic inserts. They hope to eliminate one of their inserters by printing full color ads directly in blank areas or on sides of customer statements.

If a credit card company can identify data and rules that determine the characteristics of each customer, the electronic inserts can be targeted for each statement. For example, the bill could show an ad for a minivan, a sports car, or a motorcycle depending on the demographics of the customer.

Implementation

To take advantage of this new color capability, the service bureau makes a few changes to their workflow:

- They add a duplex InfoPrint 5000 system and configure RICOH ProcessDirector to submit print jobs to it.
- They give guidelines to their customers so that it includes guidelines for using color images and logos in addition to fonts and graphics.
- They train the pre-press department is trained to use a graphics application for color profile management and image normalization.
- They add a step to their process to make sure that their customers have followed the guidelines. If they have not, the pre-press staff adjusts the resolution and input color profiles to conform to the guidelines before they submit the print job.

Related publications

You can find more information about AFP color printing on the websites of Ricoh and the AFP Color Consortium.

For more information about the AFP color management and the Color Management Object Content Architecture, see:

- *AFP Color Management Architecture*, G550-0526
- *Color Management Object Content Architecture Reference*, S550-0511

These AFP Architecture reference books contain information about color management resources:

- *Mixed Object Document Content Architecture Reference*, SC31-6802
- *Advanced Function Presentation: Programming Guide and Line Data Reference*, S544-3884

- *IPDS Reference*, S544-3417

To download or order these publications or, for more information about the activities and publications of the AFP Color Consortium, see the Consortium website:

http://afpcinc.org

For more information about the ICC, ICC profiles, and the PCS, see the ICC website: http://www.color.org

Working with fonts when using the CPSI processing engine

This section describes the fonts required for printing different data formats. It also describes what you must do so that InfoPrint can use them.

Fonts for printing transformed PostScript and PDF data

The InfoPrint PostScript transform program uses Type 1 outline fonts when transforming PostScript and PDF files. InfoPrint includes the IBM Type 1 outline fonts and these fonts are installed in subdirectories of the **/usr/lpp/psf/ps/fonts** directory during installation. If you have other Type 1 outline fonts, you can also use them with the transform program.

Note

- When InfoPrint runs the PostScript transform program during job submission, it can search other directories for fonts.

Font-mapping files

A font-mapping file identifies the file names of the Type 1 outline fonts on the AIX or Linux system. The InfoPrint PostScript transform program requires at least one font-mapping file. InfoPrint installs the default font-mapping file, **psfonts.map** in the **/usr/lpp/psf/ps** directory during installation. If you have other Type 1 outline fonts installed on your system, you can create additional font-mapping files to define their names.

PostScript font-mapping file considerations

When you create your own font-mapping files, consider that:

- You can create more than one font-mapping file.
- At least one font-mapping file must contain the Courier font. The PostScript transform program substitutes Courier for any missing fonts.
- Fonts cannot span network file systems.

Creating a PostScript font-mapping file

Use the InfoPrint **mkfntmap** command to create a font-mapping file for use with the InfoPrint PostScript transform program. You specify one or more input files containing Type 1 outline fonts and a name for the font-mapping file.

For example, to use the input file `FontFile` and create a font-mapping file named `font.map`, enter:

```
mkfntmap FontFile > font.map
```

Using font-mapping files with the ps2afp or the pdf2afp command

By default, InfoPrint uses the **psfonts.map** font-mapping file when you enter the **ps2afp** or the **pdf2afp** command; you do not have to specify the name of the default font-mapping file with the command.

To use font-mapping files that you have created when you enter the **ps2afp** or the **pdf2afp** command, you must do one of these:

- Enter the path and name of the font-mapping file as the value of the **ps_font_map_files** keyword in the transform daemon configuration file. For performance reasons, this is the recommended method. You can specify more than one font-mapping file, if necessary. Separate the font-mapping files with colons.

  For example, to specify the font-mapping files `font1.map` and `font2.map`, enter this:

```
ps_font_map_files=/path/font1.map:/path/font2.map
```

- Enter the path and name of the font-mapping file as the value of the **ps_font_map_files** transform command configuration file. Use the same syntax as described for entering the value in the transform daemon configuration file. Using this method can result in significant performance degradation.
- Specify the font-mapping file, including its path name, from the command line using the **-F** flag of the **ps2afp** or the **pdf2afp** command. You can specify multiple files using multiple **-F** flags; InfoPrint concatenates them from left-to-right (first entered to last entered) in that order.

  For example:

```
ps2afp -F/path/font.map
```

  This method can also result in degraded performance.

Note

If you use the **ps2afp -F** or the **pdf2afp -F** command flag or the **ps_font_map_files** keyword in the transform command configuration file to specify a font-mapping file, and the value is different than that specified in the transform daemon configuration file, the transform daemon restarts the PostScript interpreter so that the new value for the font-mapping file takes effect. However, restarting the PostScript interpreter decreases performance.

Specifying font substitution through initialization files

Sometimes jobs request fonts that are neither included inline with the job nor resident on the printer. If InfoPrint Manager cannot find the fonts that the job requires, the job might fail. To avoid job failure because of missing fonts, you can issue a PostScript command that substitutes a Courier font to make sure that the job prints.

To do this task, use this procedure:

1. From a command line, access the **UserInit** file that is located in the **/var/psf/ps2afp2** directory on an InfoPrint Manager AIX or InfoPrint Manager for Linux server. If you are using a configuration file and daemon other than the default **3160d.cfg** postscript daemon configuration file, the you will need to determine which daemon is being used and change the initialization file for that daemon.

2. Using your editor of choice, edit the file.
3. Add this to the **UserInit** file to make sure that missing fonts are replaced by Courier fonts to allow the job to print :

```
%!
turnFontSubstitutionOn
```

Note

If you want to make sure that fonts not inline with the job or resident on your printer do not print, type this change in the **UserInit** file:

```
%!
turnFontSubstitutionOff
```

4. Save and close the file.

Note

For your convenience, InfoPrint provides samples (**UserInit** file, **font.subs.on**, and **font.subs.off**) in the **/usr/lpp/psf/ps2afp** directory on an InfoPrint Manager AIX or InfoPrint Manager for Linux server.

You can use the desktop **Text Editor** to rename the **font.subs.on** or **font.subs.off** files and use this **UserInit** file at your installation.

When a job requests fonts that are not included inline with the job or resident on the printer, and you do not want to substitute a Courier font to make sure that the job prints, you need to embed the font or make the font available on the rip machine. If the required font is not available, the selection mechanism searches for the best match for the required font.

You can set up the pragma in the daemon configuration file used for all jobs through that daemon:

```
pragma pdf2psOptions=abortiffontmiss=yes
```

Adding non-IBM fonts for a PostScript job

If your InfoPrint system receives a PostScript job from a personal computer system, the fonts required for that job might not reside on the system. An InfoPrint Submit operator might find that either:

- Fonts are not available
- Fonts are not in the `.pfa` or `.pfb` format required for PostScript fonts

If either of these conditions occur, the AIX administrator must make these fonts available before printing the PostScript job.

Adding Type 1 fonts to a personal font library

You can add **.pfa** or **.pfb** files that print regularly at your installation to your InfoPrint Manager AIX server by using the **/var/psf/fonts** directory. Use this procedure, ensuring that you are the **root** user:

1. Open a terminal session on your InfoPrint Manager Linux server.
2. From the Linux command line, access the correct directory by typing:

```
cd /var/psf/psfonts
```

Note

To further qualify the location of your personal fonts, you might want to create a subdirectory or directories, such as **pkg_a** or **pkg_b**, using the **mkdir** command.

3. From either a DVD-ROM or your location of choice, download the files to your InfoPrint Manager server, ensuring that you copy the files in binary format.
4. From your InfoPrint Manager server in either the **/var/psf/fonts** directory or the subdirectory you created, make sure that the downloaded files have the correct permissions by entering these commands:

   1. `chown ipm1.sys*`

      to make sure that the **imp1:sys** ID owns all downloaded files.

   2. `chmod 444 *`

      to make sure that all files have read access.

   3. `chmod 664 user.map`

      to make sure that you can write to the **user.map** file.

5. From the command line in either the **/var/psf/fonts** directory or the subdirectory you created, create a font mapping file that lists PostScript Type 1 font names and their locations on the InfoPrint Manager AIX server by typing:

```
mkfntmap pkg_a/*.pfb pkg_b/*.pfb >> user.map
```

   InfoPrint Manager appends the `*.pfb` files that you stored in the **pkg_a/*.pfb** and **pkg_b/*.pfb** subdirectories to the **user.map** file. This **user.map** file is referenced in the **ps_font_map_files** resource path in the **/usr/lpp/psf/ps2afp/ps2afpd.cfg**file:

```
ps_font_map_files = /usr/lpp/psf/psfonts.map :\
                    /var/psf/psfonts/user.map
```

   The >> option adds the names to the **user.map** file without overwriting existing information.

   Note

   If you did not create a subdirectory, you can map the files to the **user.map** file by entering this command:

```
mkfntmap *.pfb >> user.map
```

Create a backup directory on your InfoPrint Manager server where you can store system changes. For example, if you create a **/customSYS** directory at / on your server, you can copy both the new font and the resulting **user.map** file to that directory:

```
cp /var/psf/psfonts/CodeZ /customSYS/
cp /var/psf/psfonts/user.map /customSYS
```

Using OpenType Fonts

This section describes what you need to use OpenType Fonts for printing with InfoPrint Manager.

These fonts offer many benefits over printing with standard font resources. They provide users with these benefits:

- A wide range of non-Latin typefaces from which to choose
- A multilingual presentation environment through the support of Unicode
- A migration path towards a single font technology across all presentation environments

OpenType Fonts have these three characteristics in the font file:

1. A Microsoft Unicode version of Version 3 Release 1.
2. Microsoft Unicode Full Font Names
3. A Unicode character map (CMAP)
   A CMAP defines the mapping of code points to glyph indices, which are used to index the actual character shape information.

If your fonts do not contain all these elements, they cannot be used with InfoPrint Manager. If the InfoPrint AFP Resource Installer (Program Number 5639-EE2) does not install a font, that font does not meet this criteria.

Note

You can use the standard AFP code pages you use for other font resources with OpenType fonts. For more information about doing this task, see *Using OpenType Fonts in an AFP System*.

Using the Resource Installer to install OpenType fonts

If you have OpenType fonts, you must install them using the InfoPrint AFP Resource Installer (Program Number 5639-EE2). If you create a new library or subdirectory for these fonts, you must also update the global search path on your InfoPrint Manager server. If you place them in an existing subdirectory, InfoPrint Manager should already have this location in its global search path.

Each time you either add new OpenType fonts or modify existing ones, you must reinstall them with the Resource Installer to make sure that InfoPrint Manager can access them correctly. The search order of directories for a data object font resources is the same as the search order for AFP font resources. Make sure that the path where you install the OpenType Fonts is in the font search order that InfoPrint Manager uses. For more information about how the font search order is defined see The search order for AFP resources, p. 109.

An administrator installs the OpenType fonts in a particular directory on each system. On an AIX server, the default path for installing these fonts is the **/usr/lpp/ipfonts** directory. On a Linux or Windows server, the default path is set up by the administrator who configured the system.

Once the OpenType fonts have been installed through the InfoPrint Fonts Collection product, you must run the Resource Installer to create the Resource Access Table (**IBM_DataObjectFont.rat**) and store it in the same directory where the fonts reside. The Resource Installer does not prompt you to do this task because the Resource Installer is not aware of changes to the directory. As a result, the system administrator who installed the new fonts must do this task whenever fonts are either added or removed from the directory.

The Resource Access Table contains an entry for each installed font so that InfoPrint Manager can access it.

For more information about how to use the Resource Installer of InfoPrint Fonts, either access the online help system of the Resource Installer or see *Using OpenType Fonts in an AFP System*.

Using font capture with OpenType fonts

Font capture allows a printer to capture a downloaded font and treat it as if it were a printer-resident font. Treating a downloaded font as a printer-resident font improves performance for future jobs that use the same fonts. Printers retain captured fonts across job boundaries and power cycles. Captured fonts can be deleted from the printer when additional space is required. Because OpenType fonts are large, you might want to make sure that your InfoPrint printer is set to support font capture.

If you want to capture OpenType fonts that reside in a library on a particular drive, click the Font Installer's **Capture** check box. (far right column of the Top pane) for each font that you want to capture. For more information, see the Resource Installer online help system.

Note

Do not capture sensitive fonts, such as signatures, because an unauthorized person could access the captured font, even from another system.

Enabling font capture for inline fonts

If the jobs that you run through your system use the same fonts, you might want to enable your actual destination to capture inline font resources. These inline fonts must have an object identifier (OID) so InfoPrint can capture them.

Note

For many systems, this procedure is not recommended because captured inline fonts can be deleted when your printer needs more storage. If InfoPrint Manager cannot access these fonts, the jobs can either fail or print incorrectly.

From the **InfoPrint Manager Administration GUI**:

1. Click the actual destination and select **Printer** → **Properties**.
2. From the actual destination's **Printer Properties** notebook, click **Tuning**.
3. Find the **Capture inline OTF resources** radio button. If you can't see it, click **Show more**.
4. Scroll to the bottom of the screen and select the **Yes** radio button for the **Capture inline OTF resources** setting.

Using multiple fonts with limited printer memory

When you are using the OpenType fonts and you receive the `Out of memory` error message, you can decrease the **Maximum fonts to keep** configuration item for InfoPrint Manager or increase the size of the character cache from the Printer console.

To decrease the number of **Maximum fonts to keep**, complete these steps:

1. Right-click the actual destination in the InfoPrint Manager GUI and click **Properties**.
2. Click **Tuning**.
3. Find and change the **Maximum fonts to keep** value to a smaller number, such as **5**.

To increase the character cache size from the Printer's console, complete these steps:

1. Click **Printer Definition**.

2. Click **Printer**.
3. Click **Resource Utilization**.
4. Increase **Font Usage** to the highest setting.

Using Unicode extended code pages

Unicode Extended Code Pages (ECP) are code pages with additional information used by OpenType and TrueType fonts. This is a introduction for using extended code pages:

- If OpenType and TrueType fonts are used without ECP, Unicode scalars must be defined in the hardcoded GUM (GCGID-to-Unicode-Mapping, where GCGID stands for Graphic Character Global IDentifier) table resident in the printer or code points will be undefined. While the use of predetermined mappings might meet many needs, if customization of GCGID to Unicode mappings is required, an extended code page must be used.
- The extended code pages can be used on printers that do not support the extensions. However, the extensions will be stripped out of the code page as it is downloaded to the printer appearing as if an unextended code page was used.
- The extended code pages are files on disk just like unextended code pages. The search order defined in *RICOH InfoPrint Manager: Reference* in the description of the resource-context-font attribute still applies. However, a new file extension of .ECP is supported for all code pages. Inside the same directory, a code page with the .ECP file extension will be selected before any other code page with a matching name but a different file extension.
- File extensions are secondary search criteria. The search order of directories is the primary search criterion when searching for any code page resource, including Unicode extended code pages. The first possible match on the resource name is used. For example, there might be two code pages called T1USV500.ECP and T1USV500. The T1USV500.ECP code page is located in a directory only specified by resource-context. The T1USV500 code page is located in a directory specified in resource-context-font. In this case, the T1USV500 code page is used. Conversely, if T1USV500.ECP and T1USV500 are located in the same directory, the T1USV500.ECP code page is used.
- Code pages created or modified by the AFP Resource Installer will always have an .ECP file extension. Use installation software to manage the installation of this new kind of resource.
- To force a code page to be downloaded instead of using a printer resident code page and to prevent run-time substitution of code pages, do the procedure Using the 'no resident' version of the sample grid files, p. 361.

Updating the global search path on an InfoPrint Manager server

If you did not use default directories when installing OpenType fonts on your system, you must update the global search path on an InfoPrint Manager server. You update the global search path by updating the **PSFPATH** environment variable to point to the directory or directories where you have installed the OpenType fonts.

Note

If you do not update the global search path, you must update each actual destination that you use to print OpenType fonts.

To set the **PSFPATH** environment variable, use the procedure Editing the /etc/profile.d/ipm_environment.sh file, p. 164 to add this line:

```
PSFPATH=$PSFPATH:path1:path2
```

where *path* is the path name to the directory containing the AFP resources. You can specify multiple path names, just separate these with a colon (:).

Specify these values in the `/etc/profile.d/ipm_environment.sh` file so that the system retains them after your InfoPrint Manager server is rebooted.

Using OpenType fonts with line data

When printing line data with OpenType fonts through InfoPrint Manager, you need to specify the appropriate parameters on the **line2afp** command. To define the location of the OpenType fonts to the **line2afp** command, you can either use the InfoPrint Manager global search path (using Updating the global search path on an InfoPrint Manager server, p. 338) or use the appropriate values on the **line2afp** command. For more information about these values, see either *Using OpenType Fonts in an AFP System* or *ACIF: User's Guide*.

For more information about either creating or modifying page definitions for line data jobs to print with OpenType fonts, see *Page Printer Formatting Aid: User's Guide*

Limitations when printing line data with OpenType fonts

Before you can print this data, you must transform it to Unicode Transformational Format (UTF-16), with big-endian byte order. Big-endian byte order means that in a two-byte UTF-16 code point the high-order byte is specified first, followed by the low-order bytes. You can transform it through the **uconv** utility, or any other application that transforms ASCII to Unicode. For more information about the **uconv** utility see the Linux operating system documentation.

Line data using OpenType fonts supports using the Byte Order Mark (BOM) and little endian data

InfoPrint Manager can process the Unicode Byte Order Mark (BOM) when printing line data with OpenType fonts. (For UTF8 or UTF16-BE data, the BOM is skipped and not printed. For UTF16-LE, the data is reversed to BE before it is printed, and the BOM is not printed.) In addition, InfoPrint Manager can now recognize multi-byte carriage return and line feed characters when parsing Unicode line data.

This support lets you process UTF8 and UTF16 line data in either little-endian or big-endian order using a traditional line data or record-format page definition. As many editors default to UTF8 or UTF16LE encodings and automatically insert a BOM at the beginning of the file, you can now print documents created with these editors using Unicode-enabled Open Type Fonts.

Mixed mode data is line data mixed with MODCA structured fields. You can use mixed mode data with UTF8, UTF16LE, and UTF16BE with a BOM as long as the BOM is the first bytes (following any cc or trc bytes) of the first line of data.

To mix UTF8 or UTF16BE data with other data encodings, you can do that if there is **no** BOM. In this case, the font selected for the data tells InfoPrint Manager the data encoding. You can't do this with UTF16LE data because it always requires a BOM.

This support includes a **-o newlineencoding** flag for the **-o** flag, a **stream,(newline=*characters, encoding)*** value for the **fileformat** parameter on the **line2afp** transform, and a **new-line-option-data-encoding** document/default document attribute. For more information about this support, see *RICOH InfoPrint Manager: Reference*.

Fonts for printing ditroff data

This section describes the files that InfoPrint provides for printing `ditroff` files on AFP printer devices and the directories in which they reside.

AFP code pages for AFP fonts

InfoPrint supplies the following three code pages for use with the AFP fonts for `ditroff`. These code pages map the existing characters in `troff` files to their corresponding AFP characters in the AFP character sets.

T1001038

Adobe Symbols (ASCII)

T1001108

`ditroff` Base Compatibility

T1001109

`ditroff` Specials Compatibility

When you install InfoPrint, both the 240-pel and 300-pel code page files are automatically installed in the **/usr/lpp/psf/fontlib** directory.

Note

If you transform a file for printing that requires fonts on a font cartridge and the printer device does not have the font cartridge installed that supports the fonts specified in the **/usr/lib/font/devhplj** directory, your file might not print correctly. See your printer documentation to install bit-mapped fonts into the **/usr/lib/font/devhplj** directory.

AFP characters mapped to ditroff characters

InfoPrint supplies AFP character sets in raster format. The character sets, which map to corresponding `ditroff` fonts, are listed in the **devafp.fontmap** file located in the **/usr/lib/font/devafp** directory. The character sets themselves are installed in the **/usr/lpp/psf/fontlib** directory.

InfoPrint supplies AFP character sets for these `troff` type families: B, BI, C, CB, CI, CBI, H, HB, HI, HBI, I, R, SG, and SS.

The following character list provides a mapping from the `ditroff` names of characters to the IBM graphics character identifier. The "IBM Graphic Character Identifier" column can be used to view the printed character as printed in a code page table. The "`ditroff` Name of Character" column can be used to print any of these characters by typing it into a text file as a special character that is processed by `troff`.

Mapping for code pages T1001038, T1001108, and T1001109

Mapping for code pages T1001038, T1001108, and T1001109

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
GA010000	*a	alpha
GA020000	*A	capital Alpha
GB010000	*b	beta
GB020000	*B	capital Beta
GD010000	*d	delta
GD020000	*D	capital Delta
GE010000	*e	epsilon
GE020000	*E	capital Epsilon
GE310000	*y	eta
GE320000	*Y	capital Eta
GF010000	*f	phi
GF020000	*F	capital Phi
GG010000	*g	gamma
GG020000	*G	capital Gamma
GH010000	*x	chi
GH020000	*X	capital Chi
GI010000	*i	iota
GI020000	*I	capital Iota
GK010000	*k	kappa
GK020000	*K	capital Kappa
GL010000	*l	lambda
GL020000	*L	capital Lambda
GM010000	*m	mu
GM020000	*M	capital Mu
GN010000	*n	nu
GN020000	*N	capital Nu
GO010000	*o	omicron
GO020000	*O	capital Omicron
GO310000	*w	omega
GO320000	*W	capital Omega

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
GP010000	*p	pi
GP020000	*P	capital Pi
GP610000	*q	psi
GP620000	*Q	capital Psi
GR010000	*r	rho
GR020000	*R	capital Rho
GS010000	*s	sigma
GS020000	*S	capital Sigma
GS610000	ts	terminal sigma
GT010000	*t	tau
GT020000	*T	capital Tau
GT610002	*h	theta
GT620000	*H	capital Theta
GU010000	*u	upsilon
GU020000	*U	capital Upsilon
GX010000	*c	xi
GX020000	*C	capital Xi
GZ010000	*z	zeta
GZ020000	*Z	capital Zeta
LA010000	a	a
LA020000	A	A
LB010000	b	b
LB020000	B	B
LC010000	c	c
LC020000	C	C
LD010000	d	d
LD020000	D	D
LE010000	e	e
LE020000	E	E
LF010000	f	f
LF020000	F	F
LF510000	ff	ff ligature
LF530000	fi	fi ligature

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
LF550000	fl	fl ligature
LF570000	Fi	ffi ligature
LF590000	Fl	ffl ligature
LG010000	g	g
LG020000	G	G
LH010000	h	h
LH020000	H	H
LI010000	i	i
LI020000	I	I
LJ010000	j	j
LJ020000	J	J
LK010000	k	k
LK020000	K	K
LL010000	l	l
LL020000	L	L
LM010000	m	m
LM020000	M	M
LN010000	n	n
LN020000	N	N
LO010000	o	o
LO020000	O	O
LP010000	p	p
LP020000	P	P
LQ010000	q	q
LQ020000	Q	Q
LR010000	r	r
LR020000	R	R
LS010000	s	s
LS020000	S	S
LT010000	t	t
LT020000	T	T
LU010000	u	u
LU020000	U	U
LV010000	v	v
LV020000	V	V

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
LW010000	w	w
LW020000	W	W
LX010000	x	x
LX020000	X	X
LY010000	Y	y
LY020000	Y	Y
LZ010000	z	z
LZ020000	Z	Z
ND010000	1	one
ND020000	2	two
ND030000	3	three
ND040000	4	four
ND050000	5	five
ND060000	6	six
ND070000	7	seven
ND080000	8	eight
ND090000	9	nine
ND100000	0	zero
NF010000	12	1/2
NF040000	14	1/4
NF050000	34	3/4
NF180000	18	1/8
NF190000	38	3/8
NF200000	58	5/8
SA000000	\-	minus
SA000000	mi	minus
SA010000	+	plus sign
SA010000	pl	math plus
SA020000	+-	plus or minus
SA030000	<	less than sign
SA040000	=	equal sign
SA040000	eq	equal sign
SA050000	>	greater than sign

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
SA060000	di	divide sign
SA070000	mu	multiplication sign
SA160000	ap	approximates
SA240000	ib	improper subset
SA270000	ip	improper superset
SA380000	ca	intersection
SA390000	cu	union
SA400000	sb	subset
SA410000	sp	superset
SA450000	if	infinity
SA470000	pt	proportional to symbol
SA480000	==	identity
SA490000	pd	partial differential symbol
SA510000	is	integral symbol
SA520000	<=	less than or equal
SA530000	>=	greater than or equal
SA540000	!=	not equal
SA670000	mo	element of
SA700000	~=	approximately equal
SA800000	sr	square root
SA870000	es	empty set
SC030000	$	dollar sign
SC040000	ct	cent sign
SD110000	'	acute accent
SD110000	aa	acute accent
SD130000	`	grave accent
SD130000	ga	grave accent
SD150000	^	circumflex accent
SD190000	~	tilde accent
SD410000	cd	cedilla accent
SF640000	br	box rule
SA860000	rn	root extender

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
SV300600	ul	underrule
SV300200	ru	rule
SG050000	lc	left ceiling
SG060000	lf	left floor
SG070000	rc	right ceiling
SG080000	rf	right floor
SG090000	lt	upper left curly bracket
SG100000	lk	left middle curly bracket
SG110000	lb	lower left curly bracket
SG120000	rt	upper right curly bracket
SG130000	rk	right middle curly bracket
SG140000	rb	lower right curly bracket
SG170000	bv	vertical bold
SL030000	gr	gradient
SM010000	#	number sign
SM020000	%	percent sign
SM030000	&	ampersand
SM040000	*	asterisk
SM040002	**	math star
SM050000	@	at
SM060000	[	left bracket
SM070000	\	backslash
SM070000	\e	backslash
SM080000	]	right bracket
SM110000	{	left brace
SM130000	or	or bar
SM130000	|	or bar
SM140000	}	right brace
SM190000	de	degree
SM240000	sc	section
SM300000	<-	left arrow
SM310000	->	right arrow

IBM Graphic Character Identifier	`ditroff` ***Name of Character***	***Character Description***
SM320000	ua	up arrow
SM330000	da	down arrow
SM340000	dg	dagger
SM350000	dd	double dagger
SM450000	sq	open square
SM470000	bx	solid square, histogram, square bullet, filled box
SM520000	co	copyright
SM530000	rg	registered
SM540000	tm	trade mark
SM570000	bu	bullet
SM590000	rh	right hand
SM630000	lh	left hand
SM660000	no	logical not
SM750000	ci	circle
SM900000	em	em dash
SP020000	!	exclamation point
SP040000	"	double quote
SP050000	fm	foot mark (minutes symbol)
SP060000	(	left paren
SP070000	)	right paren
SP080000	,	comma
SP090000	_	underscore
SP090000	_	underscore
SP100000	hy	hyphen
SP110000	.	period
SP120000	/	slash
SP120001	sl	fraction
SP130000	:	colon
SP140000	;	semicolon
SP150000	?	question mark
SP190000	\`	open quote

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
SP200000	\'	close quote
SS000000	bs	smiley face
SS680000	-	dash, minus sign (en dash)
SS680000	en	dash, minus sign (en dash)

Special symbols added for code page T1001038

These non-standard special symbols were added:

Special symbols added for code page T1001038

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
SA370000	tf	therefore, math and proof symbols
SS020000	ht	heart
SS030000	dm	diamond
SS040000	cl	club
SS050000	sd	spade

Special symbols added for code page T1001108

These non-standard special symbols were added:

Special symbols added for code page T1001108

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
SM790000	hs	house character
ND011000	s1	One superscript
ND021000	s2	Two superscript
ND031000	s3	Three superscript
ND041000	s4	Four superscript
ND051000	s5	Five superscript
ND061000	s6	Six superscript
ND071000	s7	Seven superscript
ND081000	s8	Eight superscript

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
ND091000	s9	Nine superscript
ND101000	s0	Zero superscript

Special symbol added for code page T1001109

This non-standard special symbol was added:

Special symbol added for code page T1001109

IBM Graphic Character Identifier	ditroff *Name of Character*	*Character Description*
SV300400	ol	overrule

Adding other troff fonts

InfoPrint provides AFP fonts that map to existing troff fonts in the Courier, Helvetica, and Times type families and AFP fonts for special characters. The font-mapping table contained in the **devafp.fontmap** file maps the AFP fonts to the corresponding troff fonts.

If you want to add troff fonts or special characters, contact your service representative.

ditroff font files

InfoPrint installs other files for ditroff in the **/usr/lib/font/devafp** directory, including these:

devafp.fontmap

File that maps ditroff fonts to AFP fonts.

***.d2afp**

Special versions of the ditroff font descriptions. The **d2afp** command uses these files.

DESC (no extension)

troff device description, ASCII version.

DESC.out

troff device description, binary version.

***.out**

troff font descriptions, binary version.

Fonts for printing DBCS ASCII and EUC data

In ideographic languages, like Japanese, Chinese, and Korean, there are thousands of characters. A single byte of information cannot represent all of these characters because a single byte only accommodates 256 characters. Therefore, at least one more byte of information is required.

For DBCS ASCII files, each ideographic character is exactly two bytes. For Extended UNIX Code (EUC) files, each ideographic character can be two, three, or four bytes, depending on the EUC implementation and language. Single-byte ASCII characters can be mixed in with DBCS ASCII and EUC characters.

InfoPrint supports printing DBCS ASCII and EUC files in these formats:

- DBCS ASCII files in the Japanese and Traditional Chinese languages, which can contain formatting controls for the IBM *5577* or *5587* printers.
- EUC files in the Japanese, Traditional Chinese, and Korean languages. EUC files cannot contain any formatting control characters.

Fonts required to print a double-byte transformed file

You can use the **db2afp** transform command to transform input files that use these code pages:

- Japanese PC (code page IBM-932)
- Japanese EUC (code page IBM-eucJP)
- Traditional Chinese PC (code page IBM-938)
- Traditional Chinese EUC (code page IBM-eucTW)
- Korean EUC (code page IBM-ecuKR)

The transformed files contain font references to the double-byte character set. You set the character set, code page, you want by setting the **PSFDBLANG** environment variable; see Setting InfoPrint Manager environment variables, p. 161.

After you have used the **db2afp** command to transform DBCS ASCII and EUC files into AFP data stream files, you must have access to DBCS fonts when you print the transformed files. The DBCS fonts for Japanese, Traditional Chinese, and Korean are not part of InfoPrint but are available for the MVS, VM, AIX and LINUX operating systems as separately orderable licensed programs.

Installing DBCS fonts

To make the fonts available on your system and to InfoPrint, once you have the licensed programs, you must do one of these:

- Use the licensed programs to install the fonts onto your system.
- Use the licensed programs to install the fonts onto a system other than yours and use the Network File System (NFS) **mount** command to mount the file system onto your system.
- Use the NFS **mount** command to mount the MVS or VM minidisk to your system.

Setting up font resources for DBCS ASCII and EUC printing

To print the transformed files that contain font references to the double-byte character set, you must make the double-byte fonts known to InfoPrint through one of these methods:

- Create a resource-context object that identifies the location of the double-byte fonts and associate the resource-context object with a default-document object and an InfoPrint logical printer; see Creating and managing resource-context objects, p. 108 for these procedures. Using this method makes the fonts known to InfoPrint for any job submitted to the logical printer associated with the default-document object.
- Add the directories to the search path for actual destinations using the InfoPrint Manager Administration Interface Printer Properties notebook. In the AFP Resources tab, enter the directory path in the `Location of fonts` field. See Assigning resource-context objects to an actual destination, p. 117 for a Linux command line equivalent of the procedure. Using this method makes the fonts known to InfoPrint for any job processed by those physical printers.
- Set and export the **PSFPATH** environment variable to your environment by entering this command:

```
export PSFPATH=$PSFPATH:path1:path2:pathN
```

  Using this method makes the fonts known to InfoPrint for all print jobs submitted from the windows or High Function Terminal session in which the **PSFPATH** environment variable was set.
- Set and export the **PSFPATH** environment variable in your **/etc/profile.d/ipm_environment.sh** file (Bash and Bourne shells) or **.cshrc** file (C shells) by adding this line to the file:

```
export PSFPATH=$PSFPATH:path1:path2:pathN
```

  Using this method makes the fonts known to InfoPrint for any job you submit.
- Set the **PSFPATH** environment variable in the **/etc/environment** file by adding this line to the file:

```
PSFPATH=path1:path2:pathN
```

  Using this method makes the fonts known to InfoPrint for all users on the system.

Fonts for printing line data

Improve flexibility by specifying the fonts that you want to use for printing line data in your page definition. See the *PPFA User's Guide*, S544–5284 or the *PPFA Quick Reference*, G544–3701 for more information. The **chars** keyword can also be used.

When you specify a coded font name with the **chars** keyword of the **line2afp** command or with the **chars** document attribute, the font name is limited to four characters, excluding the two-character prefix.

Font mapping table for use with the chars keyword, p. 352 provides a list of the InfoPrint Expanded Core Fonts for use with unformatted ASCII input data. Because these fonts have eight-character names, the table also provides a list of six-character short names. InfoPrint stores these coded fonts in the **/usr/lpp/psf/reslib** directory. There are symbolic links of the eight-character names that correspond to the six-character names. You can use these short names, *without* the `X0` prefix, to satisfy the four-character limitation for specifying font names with the **chars** keyword of the **line2afp** command. You can use either the coded font name or the page definition.

For the names of other coded fonts, see *InfoPrint Fonts: Font Summary*.

Font mapping table for use with the chars keyword

Type Family	*Point Size*	*Coded Font Name*	*Linked Short Name (for* `chars` *keyword)*
Courier	7	X0423072	X04272
Courier	8	X0423082	X04282
Courier	10	X0423002	X04202
Courier	12	X04230B2	X042B2
Courier	14	X04230D2	X042D2
Courier	20	X04230J2	X042J2
Helvetica	6	X0H23062	X0H262
Helvetica	7	X0H23072	X0H272
Helvetica	8	X0H23082	X0H282
Helvetica	9	X0H23092	X0H292
Helvetica	10	X0H23002	X0H202
Helvetica	11	X0H230A2	X0H2A2
Helvetica	12	X0H230B2	X0H2B2
Helvetica	14	X0H230D2	X0H2D2
Helvetica	16	X0H230F2	X0H2F2
Helvetica	18	X0H230H2	X0H2H2
Helvetica	20	X0H230J2	X0H2J2
Helvetica	24	X0H230N2	X0H2N2
Helvetica	30	X0H230T2	X0H2T2
Helvetica	36	X0H230Z2	X0H2Z2
Times New Roman	6	X0N23062	X0N262
Times New Roman	7	X0N23072	X0N272
Times New Roman	8	X0N23082	X0N282
Times New Roman	9	X0N23092	X0N292
Times New Roman	10	X0N23002	X0N202
Times New Roman	11	X0N230A2	X0N2A2
Times New Roman	12	X0N230B2	X0N2B2
Times New Roman	14	X0N230D2	X0N2D2
Times New Roman	16	X0N230F2	X0N2F2
Times New Roman	18	X0N230H2	X0N2H2
Times New Roman	20	X0N230J2	X0N2J2
Times New Roman	24	X0N230N2	X0N2N2

Type Family	*Point Size*	*Coded Font Name*	*Linked Short Name (for `chars` keyword)*
Times New Roman	30	X0N230T2	X0N2T2
Times New Roman	36	X0N230Z2	X0N2Z2

Printing with the DBCS Simulation Fonts with the font metric adjustment triplet

InfoPrint Manager includes support for printing the DBCS Simulation Fonts (available with InfoPrint Fonts Collection) with the font metric adjustment triplet. The DBCS Simulation Fonts are available on the AIX and Windows operating systems.

Note

When you use the DBCS Simulation Fonts, set the **requested-font-resolution** appropriately or use the NORES grid files.

Installing the Japanese PostScript font feature

You can install Japanese PostScript fonts on your InfoPrint Manager server to print documents requiring these specialized fonts.

To enable your InfoPrint Manager server to support Japanese PostScript fonts, use the InfoPrint installer located on the InfoPrint Manager DVD-ROM. You will be prompted to supply your customer ID as well as the access key located on the invoice that was shipped with your DVD.

Using global resource identifiers

InfoPrint can identify the fonts required for a job, access AFP font files that are installed on or NFS-mounted to the pSeries, and download the appropriate fonts to an InfoPrint-managed printer device. However, some InfoPrint printers can use fonts that are resident in the printer, saving the time it takes to download these fonts. Because the font files on the pSeries do not share the same names as the printer-resident fonts, InfoPrint uses global resource identifiers (GRIDs) to map the pSeries-resident fonts to the printer-resident fonts. Use the information in this chapter to work with GRIDs.

How can you use GRIDs?

GRID files contain pre-assigned mappings that you can modify to map pSeries-resident font file names to printer-resident font identifiers. These pre-assigned mappings include both single-byte character set (SBCS) and double-byte characters set (DBCS) outline fonts.

If you have a printer that supports outline fonts, you can use the GRID files to map raster character-set names to outline font global identifiers.

A third use of GRID files is to support Mixed Object Document Content Architecture for Presentation (MO:DCA-P) files that specify fonts using GRIDs rather than character-set names. You can customize the GRID files to define which fonts InfoPrint uses when the data stream of the job specifies GRIDs.

Note

If a data stream specifies only GRID parts instead of specific names, then the **fgid.grd** table is used to find the name of the font to download or the outline font to activate. In this case, the data stream must specify a specific width value.

What GRID files ship with InfoPrint?

InfoPrint ships four GRID files that you can customize for use with the types of printer devices that you use and the types of jobs that you print. These files are **charset.grd.sample**, **codepage.grd.sample**, **cpgid.grd.sample**, and **fgid.grd.sample**, and they are installed in the **/usr/lpp/psf/grd** directory.

Note

You can either shut down and restart the InfoPrint server (**pdserver** process) or shut down and restart all your printers to make sure that **.grd** file changes are applied.

This section describes each file:

charset.grd.sample

Specifies which printer-resident fonts are activated for the character-set names found in the data stream of the job.

This file maps the font character-set name to a:

- Font global identifier (**fgid**)
- Graphic character set global identifier (**gcsgid**)
- Font width
- Vertical size value
- Set of font attributes (bold, italic, and doublewide)

Sample file for a character set, p. 354 shows an example of the format of a line in the character set sample file.

Sample file for a character set

```
#  FCS name ->  fgid  gcsgid  width  vsize attr                    #
#*************************************************************#
   C0L00GSC     398     -      96     120   -  # GOTHIC CONDENSED  1
```

codepage.grd.sample

Specifies which printer-resident fonts are activated for the code page names found in the data stream of the job.

This file maps a code page to the:

- Code page global identifier (**cpgid**)
- Graphic character set global identifier (**gcsgid**)

Sample file for a code page, p. 355 shows an example of the format of a line in the code page sample file.

Sample file for a code page

```
#  code page name  ->  cpgid   gcsgid                                  #
#*************************************************************************#
       T1000038           38    123   # US-ASCII Character Set     1
```

fgid.grd.sample

Specifies which font character set InfoPrint should download when substituting for a global resource identifier that it finds in the data stream of the job.

This file maps these to a font character set (FCS) name:

- A font global identifier (**fgid**)
- A graphic character set global identifier (**gcsgid**)
- A font width
- A vertical size value

Sample file for a global identifier, p. 355 shows an example of the format of a line in the font global identifier sample file.

Sample file for a global identifier

```
#  fgid   gcsgid  width   vsize -> FCS name                            #
#*************************************************************************#
      3        0    144      240   C0E0DE10  # DELEGATE 12 POINT   1
```

cpgid.grd.sample

Specifies which code page InfoPrint should download when substituting for a global resource identifier that it finds in the data stream of the job.

This file maps a code page global identifier and a graphic character set global identifier to a code page name. Sample file for a code page identifier, p. 355 shows an example of the format of a line in the code page identifier sample file.

Sample file for a code page identifier

```
#  cpgid  gcsgid  ->  code page name                                    #
#*************************************************************************#
      29       0      T1V10871  # OLD ICELANDIC                       1
```

For a detailed explanation of how the different types of identifiers combine to produce fonts, see *InfoPrint Fonts: Font Summary*

If you have a job that specifies a font that is not resident on your printer, you can use the GRID files to map the missing font to one with the same point size that does reside on your printer. To understand what fonts your printer device has, see your printer documentation.

Where does InfoPrint search for GRID files?

InfoPrint searches these directories, in the order shown, for GRID files:

1. An internal list that matches the **charset.grd.sample** file.
2. **/usr/lpp/psf/grd**
3. **/var/psf/***printername*, where *PrinterName* is the name of a valid InfoPrint physical printer.

   Note

   When the same mapping is found in one of these files, InfoPrint uses the last one (**/var/psf/***PrinterName*).

When you install InfoPrint, sample GRID files are installed in the **/usr/lpp/psf/grd** directory. You can maintain a central location for new or modified GRID files for specific printers by storing files in the **/var/psf/***PrinterName* directory.

If you are mapping raster fonts to outline fonts, you can map multiple names to the same GRID part so that different sized fonts can apply to your print job.

Note

1. The new or modified GRID files you create have the same names as the sample files except that you delete **.sample** from each file name.
2. After creating or modifying GRID files, you must shut down and restart the InfoPrint physical printer represented by *PrinterName* for the changes in the file to take effect.

Understanding the general syntax rules and allowable values for GRID files

To tailor the InfoPrint-supplied GRID files for your own system needs, you need to understand the general syntax rules for the four GRID files and the allowable values for the fields in the files.

Syntax rules applying to all types of GRID files

The GRID files are simple ASCII text files that you can modify through any standard Linux text editor. These rules apply to all of the GRID files:

- Each line in the file cannot contain more than 255 characters
- Separate fields within a line with one or more delimiters (spaces)
- A pound sign (#) indicates the start of comments, which causes InfoPrint to ignore the remainder of that line
- A dash, – , indicates the absence of a particular value, for which InfoPrint substitutes a zero (0)
- An asterisk, * , indicates that a wildcard value (any numeric decimal) is accepted.
- InfoPrint ignores blank lines and extraneous fields.

Allowable values for the charset.grd file

You can specify these values for fields in the GRID file for a character set for a font:

fgid

Identifies the character set identifier for the font. Specify a numeric value from 1 to 65534 or enter a dash, – , to disable the mapping of this character set to GRID values.

gcsgid

Identifies a global identifier for the font. Specify a numeric value from 0 to 65534 or enter a dash, – , to specify no **gcsgid**. If you specify an asterisk, InfoPrint issues an error message.

width

Identifies the width specified in the character set. Specify a numeric value from 0 to 32767 or enter a dash, – , to specify no width. If you specify an asterisk, InfoPrint issues an error message.

vertical_size

Identifies the vertical size specified in the character set. Specify a numeric value from 0 to 32767 or enter a dash, – , to specify no vertical size. If you specify an asterisk, InfoPrint issues an error message.

attribute_field

Indicates the attributes of the character set. Specify a lowercase b (bold), i (italic), d (doublewide), or enter a dash, – , to specify no attribute.

Allowable values for the codepage.grd file

You can specify these values for fields in the GRID file for a code page for a font:

cpgid

Identifies the code page identifier for the font. Specify a numeric value from 1 to 65534 or enter a dash, – , to disable the mapping of this character set to GRID values.

gcsgid

Identifies a global identifier for the code page. Specify a numeric value from 0 to 65534 or enter a dash, – , to specify no **gcsgid**. If you specify an asterisk, InfoPrint issues an error message.

Allowable values for the fgid.grd file

You can specify these values for fields in the GRID file for a font global identifier:

fgid

Identifies the global identifier for the font. Specify a numeric value from 1 to 65534.or enter an asterisk, * , to indicate a wildcard value. If you specify a dash, InfoPrint issues an error message.

gcsgid

Identifies the global identifier for the font. Specify a numeric value from 0 to 65534 or enter an asterisk, * , to indicate a wildcard value. If you specify a dash, InfoPrint issues an error message.

width

Identifies the number of characters per horizontal inch. Specify a numeric value from 1 to 32767 or enter an asterisk, * , to indicate a wildcard value. If you specify a dash, InfoPrint issues an error message.

vertical_size

Identifies the vertical size of the font. Specify a numeric value from 1 to 32767 or enter an asterisk, * , to indicate a wildcard value. If you specify a dash, InfoPrint issues an error message.

Allowable values for the cpgid.grd file

You can specify these values for fields in the GRID file for a code page global identifier:

cpgid

Identifies code page identifier for the font. Specify a numeric value from 1 to 65534.

gcsgid

Identifies the global identifier for the font. Specify a numeric value from 0 to 65534 or enter an asterisk, * , to indicate a wildcard value. If you specify a dash, InfoPrint issues an error message.

Modifying GRID files

This section describes the two ways you can modify GRID files to support the printer devices and types of jobs in your production printing environment.

Note

If you are using InfoPrint to print from the InfoPrint 6400 or 6500 series of line matrix printer in either native or 4234 emulation mode, you must modify both the **charset.64xx.grd** and the **codepage.64xx.grd** files as described in Modifying the charset and codepage GRID files, p. 358 and Modifying the FGID and CPGID GRID files, p. 359.

Modifying the charset and codepage GRID files

You can avoid having InfoPrint download fonts by adding font entries that map the font character set or code page that is most referenced in jobs to a global identifier for a resident font.

For example, if your jobs specify the Prestige 10-point font that is character set C0S0PR10, and that font is not available, you can map the Prestige 10-point font to a similar font that is either resident on your printer or available for InfoPrint to download to your printer. In this procedure, you map the Prestige 10-point font to Gothic 10-point font, character set C0D0GT10.

Perform these steps:

1. From the Linux command line, change to the **/var/psf/***PrinterName* directory, where *PrinterName* is the name of the InfoPrint actual destination representing the printer device.
2. Create the **charset.grd** file from the **charset.grd.sample** file. Enter the command:

```
cp /usr/lpp/psf/grd/charset.grd.sample charset.grd
```

3. Using an Linux editor such as vi, edit the **charset.grd** file you just created. Copy this line for the Gothic 10-point printer-resident font and paste it at the end of the **charset.grd** file:

```
C0D0GT10  40   - 144   240     #  GOTHIC TEXT 10
```

4. Change the first entry in the copied line, in the **FCS name** column, to the character set identifier of the Prestige 10-point font that you want to map to the Gothic 10-point font. You can change the comment at the end of the copied line to identify the mapping change you are making.

```
C0S0PR10  40   - 144   240     #  PRESTIGE 10 mapped to GOTHIC TEXT 10
```

5. Search the remainder of the **charset.grd** file to make sure that it contains no other C0S0PR10 entries. A later entry of the same character set would replace the entry you just made.
6. Save your changes and close the file.

To print jobs that specify the personal computer ASCII code page for Iceland, T1000861, but that code page is not available, you can map code page T1000861 to the personal computer multilingual code page for ASCII data, T1000850.

Perform these steps:

1. From the Linux command line, change to the **/var/psf/***PrinterName* directory, where *PrinterName* is the name of the InfoPrint actual destination representing the printer device.
2. Create the **codepage.grd** file from the **codepage.grd.sample** file. Enter the command:

```
cp /usr/lpp/psf/grd/codepage.grd.sample codepage.grd
```

3. Using an Linux editor, such as vi, edit the **codepage.grd** file you just created. Copy this line for the personal computer multilingual code page for ASCII data and paste it at the end of the **codepage.grd** file:

```
T1000850        850     980  # PC MULTILINGUAL
```

4. Change the first entry in the copied line, in the **code page name** column, to the code page identifier of the personal computer ASCII code page for Iceland font that you want to map to the multilingual code page. You can change the comment at the end of the copied line to identify the mapping change you are making.

```
T1000861        850     980  # ICELAND MAPPED TO MULTILINGUAL
```

5. Search the remainder of the **codepage.grd** file to make sure that it contains no other T1000861 entries. A later entry of the same character set would replace the entry you just made.
6. Save your changes and close the file.

Modifying the FGID and CPGID GRID files

You can add new entries to map your GRID data to the resident fonts in your printer by modifying either the **fgid.grd.sample** file or the **cpgid.grd.sample** file. If you have a printer that does not support the outline font technology, We recommend that you map all the `CZ`*nnnn* entries to raster font character-set names. If you do not create this mapping, InfoPrint issues an error message and does not print the job.

For example, if you wanted to map the Times New Roman Latin 1 character set to the raster version, you would do these steps:

1. From theLinux command line, change to the **/var/psf/**`PrinterName` directory, where *PrinterName* is the name of the InfoPrint physical printer representing the printer device.

2. Create the **fgid.grd** file from the **fgid.grd.sample** file. Enter the command:

```
cp /usr/lpp/psf/grd/fgid.grd.sample fgid.grd
```

3. Using an Linux editor, edit the file that you just created and copy this line:

```
2308    2039  *   *   CZN200      # Times New Roman Latin1
```

4. Change the entry in the **FCS name** column to `CON20000`.

```
2308    2039  *   *   CON20000    # Raster version of TNR Latin 1
```

5. Save your changes and close the file.

After this substitution, InfoPrint prints jobs that specify the font character-set name `CZN200` and that processed on the physical printer for which you made the change using the raster version of the font. The `CZN200` character set now maps to:

- An FGID of `2308`
- A GCSGID of `2039`
- A vertical size of `65535`
- A width of `0`

Note

For more information about the different values available for font character sets and code pages, see "IBM Font Interchange Information" in *InfoPrint Data Stream and Object Architectures Mixed Object Document Content Architecture Reference*.

When a printer does not support any resident outline fonts, the **fgid.grd.sample** file maps the **FGID** and the **GCSGID** back to the `CON20000` character set. Then, InfoPrint searches the resource library search path to locate the `CON20000` character for downloading with the job.

To modify the **cpgid** file so that the personal computer ASCII code page for Iceland maps to the multilingual code page for ASCII data as described in Modifying the charset and codepage GRID files, p. 358, do these steps:

1. From the Linux command line, change to the **/var/psf/***PrinterName* directory, where *PrinterName* is the name of the InfoPrint physical printer representing the printer device.
2. Create the **cpgid.grd** file from the **cpgid.grd.sample** file. Enter the command:

```
cp /usr/lpp/psf/grd/cpgid.grd.sample cpgid.grd
```

3. Using an Linux editor, , edit the **cpgid.grd** file you just created and copy this line for the personal computer multilingual code page for ASCII data:

```
850     980      T1000850  # PC MULTILINGUAL
```

4. Change the first entry in the copied line, in the **cpgid** column, to the code page identifier of the personal computer ASCII code page for Iceland font that you want to map to the multilingual code page. You can change the comment at the end of the copied line to identify the mapping change you are making.

```
861      980      T1000850 # ICELAND MAPPED TO MULTILINGUAL
```

5. Search the remainder of the **cpgid.grd** file to make sure that it contains no other 861 entries. A later entry of the same character set would replace the entry you just made.
6. Save your changes and close the file.

Verifying the GRID files

During physical printer initialization, InfoPrint examines the contents of the GRID files. If InfoPrint detects any syntax errors, it reports an error condition and fails to enable the physical printer. To correct these errors, you must edit the failing GRID files. For information about what files, lines, or fields are in error, you must access the GRID **error.log** file for the physical printer that is located in **/var/psf/***PrinterName*.

Preventing InfoPrint from using GRID files

If you have obtained a 300-pel printer but print 240-pel jobs, the printer-resident fonts might have different line spacings and endings in the printed output. For some types of billing statements, these line endings are inappropriate. You can solve this problem by preventing InfoPrint from using the GRID files. This prevents the use of the printer-resident fonts. Perform this procedure:

1. Using the Linux **mv** command, rename your customized GRID files. For example, if you are in the **/var/psf/***PrinterName* directory and you want to rename the **charset.grd** to `charset.grd.bak`, enter:

```
mv charset.grd charset.grd.bak
```

2. Copy the **/usr/lpp/psf/grd/charset.grd.sample** file to **charset.grd** in the **/var/psf/***PrinterName* directory.
3. Using an Linux editor, , edit the copied **charset.grd** file and place a dash (–) in the **fgid** column.
4. Copy the **/usr/lpp/psf/grd/codepage.grd.sample** file to **codepage.grd** in the **/var/psf/***PrinterName* directory.
5. Edit the copied **codepage.grd** file and place a dash (–) in the **cpgid** column.

Using the 'no resident' version of the sample grid files

There might be situations when you do not want InfoPrint Manager to use printer-resident fonts.

If you do not have customized GRID files and want to use the 'no resident' version of the sample grid files shipped with InfoPrint Manager, complete this procedure:

1. From the Linux command line:

```
cd /var/psf/PrinterName
```

   Where *PrinterName* is the name of the InfoPrint physical printer that represents the actual printer device.

2. Create the `charset.grd` file from the `charset.grd.nores` file:

```
cp /usr/lpp/psf/grd/charset.grd.nores charset.grd
```

3. Create the codepage.grd file from the codepage.grd.nores file:

```
cp /usr/lpp/psf/grd/codepage.grd.nores codepage.grd
```

Configuring and using cut-sheet emulation

To enable cut-sheet emulation, make sure that your printer has a microcode level of at least 9.216, then make sure that the appropriate personnel do these tasks at your installation on the systems specified:

1. At the InfoPrint Linux server, the print administrator for your installation must disable the physical printer that you intend to run in cut sheet emulation mode.
2. At the printer console, an operator must set the printer to cut-sheet emulation mode.
3. The operator must then restart the printer.
4. At the InfoPrint Linux server, the print administrator must create a logical printer for the physical printer.
5. The print administrator must make sure that both the logical and physical printer has been enabled.

To submit a job, see "Using Cut Sheet Emulation."

Using cut sheet emulation

Job submitters must specify a form definition that will print using cut sheet emulation through these application programming interfaces:

- Using InfoPrint Submit Express, p. 362
- Using the PSF actual destination properties notebook, p. 362
- Using the pdpr command, p. 363

Before using any of these interfaces, make sure that you are aware of this support's Defaults and limitations of cut sheet emulation support, p. 363.

Using InfoPrint Submit Express

From the **Resources** tab of the InfoPrint Submit Express Job Ticket window, you can leave the **Use form definition** field blank to get the default form definition on the InfoPrint Manager for Linux server, or specify a specific form definition.

By using the default form definition, cut sheet emulation will be invoked.

Note

InfoPrint Submit Express is part of the *RICOH InfoPrint Manager: Common Clients* distribution.

Using the PSF actual destination properties notebook

1. From the InfoPrint Administration *servername* window, use the middle mouse button to drag the printer notebook action icon onto the Actual Destination (printer) that you want to use for cut-sheet emulation.
2. From the View and Change PSF Actual Destination Properties window, click the **Customization** tab.
3. Specify the form definition of choice in the form definition field.
4. Click **OK** or **Apply** to make sure that this form definition is applied.

Using the pdpr command

The **form-definition** document, default document, or actual destination attributes on the **pdpr** command also change the form definition on a printer.

For specific examples of how to modify these values on the printer, see *RICOH InfoPrint Manager: Reference*.

Defaults and limitations of cut sheet emulation support

When a form definition is explicitly named, any attribute found in both the form definition and either the InfoPrint Manager graphical user interface or the **pdpr** command reverts to the value specified in the form definition.

Note that corresponding values specified on the **printer-pass-through** (or **destination-pass-through**) attribute override the values specified in either the **PSF Resource** tab of the PSF printer notebook or the **form-definition** document, default document, or actual destination attribute.

If you use a form definition that contains a medium map using **N_UP** commands, the printer will go out of cut sheet emulation mode while processing that **N_UP** command. If the **N_UP** processing ends, the printer will return to cut sheet emulation mode. This scenario can occur when printing a large image in the middle of a job.

Generating and submitting color mapping table source and output files

A color mapping table is a printer resource object that defines a translation from certain MO:DCA structured fields to new color structured fields used by newer printers. Color mapping tables are used to provide support for highlight color attachments with black-and-white printers, such as using the InfoPrint 4005 attachment with the InfoPrint 4100 printer. You can also use a color mapping table to map highlight colors into richer colors on a color printer like the InfoPrint 5000 printer. To apply color to black and white documents, you must use the following procedure to create Color Mapping Table Source and Output Files. See the *RICOH InfoPrint Manager: Reference* for information about the cmt utility or to the appropriate man page.

You can define translations from non-color fields to color, old color fields to new color fields, and from new color fields to different new color fields. Therefore, you can use existing applications and documents with new color fields without having to change the documents or applications. Also, you can use various color mappings with a single document to print the document using color in different ways without changing the original document.

Using the AFP architectural reference, you can apply color to existing black and white documents. The sample configuration file **cmt.cfg** contains user-defined specifications, and should be available in the **usr/lpp/psf/config** directory. The **cmt** utility that uses the **cmt.cfg** configuration file should be available in the **/usr/lpp/psf/bin** directory on your InfoPrint Linux server. Note that a MOD:CA sample color mapping table, named **cmtsampl**, can be found in **usr/lpp/psf/config/**. This color mapping table was produced by the sample **cmt.cfg** file.

The **cmt** utility lets you create color mappings in a variety of different color modes that are defined through the **ColorSpace** parameter. For example, the **Highlight** value can be used when printing to an InfoPrint 4000 printer with an InfoPrint Highlight Color Model HC2 Post-Processor (HCPP). For more information about the HCPP, see the *InfoPrint Hi-Lite Color Printer: Library Catalog*.

Parts of a color mapping table

A color mapping table consists of a base part, a set of source groups, and a set of target groups. The base part identifies the color mapping table with a type of either `reset` or `normal`.

The simplest possible color mapping table is a reset color mapping table, which tells the printer to do no transformations on the color information found in the document. A reset color mapping table has no source or target groups; all other color mapping tables have at least one source and one target group.

Using source groups

Each source group has an identification (ID) number, which is used to match the source group with a corresponding target group. You can have unique ID numbers; however, if you create the color mapping table without the **cmt** Utility and want to map several source groups to a single target group, you can use the same ID number on multiple source groups. Valid ID values are 1–127.

Note

The **cmt** Utility generates pairs of sources and targets, then assigns increasing, sequential IDs to each pair. For more information about the **cmt** Utility, see *RICOH InfoPrint Manager: Reference*.

Each source group must be classified as one of these color spaces:

Highlight color

Highlight color is used when your existing documents describe color in terms of the percent to be covered and the percent to be shaded for a color number. The colors are device dependent. For example, you can have a printer that allows the use of three colors for highlighting. You could specify percent coverage and percent shading for colors 1, 2, or 3. Your printer setup would determine the actual colors.

Standard Object Content Architecture (OCA)

Standard OCA uses defined combinations of red, green, and blue to create: blue, red, pink, magenta, green, turquoise/cyan, and yellow. Standard OCA also defines several defaults such as white on a black medium, black on a white medium, and the same color as the medium. The medium could be, for example, paper or a display.

GOCA pattern fill

GOCA pattern fill defines patterns for filling areas that you might want to map to colors with a color mapping table.

For detailed explanations about these color spaces, see *Mixed Object Document Content Architecture Reference*.

You can use a color mapping table to pick specific object types to map, such as:

- Object area
- IM image data
- PTOCA data
- Page presentation space

- GOCA data
- Overlay presentation space
- BCOCA data
- IOCA data (bi-level, FS10)
- All PTOCA, GOCA, BCOCA, IOCA, FS10, and IM object data
- All objects, object areas, and presentation spaces

After you pick the color spaces and object types you want to map, you can specify exact values for the fields you want to map.

Using target groups

If you create the color mapping table without the **cmt** Utility, you must specify a target group ID number to match the source groups for a target. Valid ID values are 1–127.

Note

The **cmt** Utility generates pairs of sources and targets, then assigns increasing, sequential IDs to each pair. For more information about the **cmt** Utility, see the *RICOH InfoPrint Manager: Reference*.

Each target group must be classified as a color space, which is the kind of color you want as output. The color spaces defined in the matching source groups are transformed to this color space. The color mapping table allows RGB CMYK, highlight, and CIELAB, but your actual hardware determines your available choices. For example, if your printer supports only highlight color, your target group must use highlight color.

You can specify exact values for output colors. For example, if your printer supports three highlight colors, you can specify colors 1, 2, or 3 with coverage and shading percentages that are supported by your printer.

Other publications that have helpful references regarding color devices include:

- *InfoPrint 4005 Hi-Lite Color Application Design Reference*, available from your Ricoh marketing representative.
- *InfoPrint Hi-Lite Color Introduction and Planning Guide for 4005*.

Creating a color mapping table

In the procedure described below, we create a color mapping table to print an Advanced Function Presentation (AFP) file that contains pie charts in the color green.

For our AFP files, this pie chart appears as a series of horizontal lines. By accessing the chapter on "Graphic primitives and attributes" in *Graphics Objects Content Architecture for Advanced Function Presentation Reference*, we can see that these horizontal lines are created by a Pattern Output Primitive with a hexadecimal value of `'0B'(11)`.

To turn these horizontal lines into a shade of green when using a **ColorSpace** source group of `GOCA` to map to a target group value of `RGB`, make sure that you specify a low value for red (12), a high value for green (252), and a value under 50 for blue (42) to provide a clear contrast.

You must do these tasks from the Linux command line:

1. From the **usr/lpp/psf/config** directory, specify:

```
cp cmt.cfg piel.cfg
```

   This copies the Color Mapping Table configuration file into a file you can customize for your own purposes (in this case, `pie1.cfg`)

2. Specify `vi piel.cfg`.

   This lets you edit the configuration file and insert the applicable values to create the appropriate color mapping (whether GOCA to RGB or OCA to CIELAB) table. See the sample **cmt.cfg** configuration file below:


```
# ********************************************************-
 # Required, starts  a Color Mapping Definition.
 # One definition for each Source to Target mapping.
# ********************************************************-
# ********************************************************-
 # BeginSourceDef:
 # Required, starts the Source Parameters
# ********************************************************-
BeginSourceDef:
# ********************************************************-
# ColorSpace:
# Required, values= OCA | Highlight | GOCA
# ********************************************************-
ColorSpace: GOCA
# ********************************************************-
# ColorValue:
# Required, values depend on Color Space
# ********************************************************-
ColorValue: 11
# ********************************************************-
# ObjectType:
# Optional, values= ObjArea | ImageData | PTOCData |
#GOCAData | BCOCAData | AllOCA | Page | Overlay |
#ObjsAll
# default= ObjsAll
# ********************************************************-
ObjectType: GOCAData
# ********************************************************-
# PercentShading:
# Optional, only valid for SourceColorSapce: Highlight,
# values= 0 .. 100, 255 (all percentages),
# default= 100
# ********************************************************-
# PercentShading:
# ********************************************************-
# PercentCoverage:
# Optional, only valid for SourceColorSpace: Highlight,
# values= 0 .. 100, 255 (all percentages),
# default= 100
# ********************************************************-
# PercentCoverage:
# ********************************************************-
# EndSourceDef:
# Required, ends the Source Parameters
# ********************************************************-
EndSourceDef:
# ********************************************************-
# BeginTargetDef:
```

```
# Required, starts the Target Parameters
# ******************************************************-
BeginTargetDef:
# ******************************************************-
# ColorSpace:
# Required, values= RGB | CMYK | Highlight | CIELAB
# ******************************************************-
ColorSpace: RGB
# ******************************************************-
# ColorValue:
# Required, values depend on Color Space
# ******************************************************-
ColorValue: 12 252 42
# ******************************************************-
# PercentShading:
# Optional, only valid for TargetColorSpace: Highlight,
# values= 0 .. 100, default= 100
# ******************************************************-
# PercentShading:
# ******************************************************-
# PercentCoverage:
# Optional, only valid for TargetColorSpace: Highlight,
# values= 0 .. 100, default= 100
# ******************************************************-
# PercentCoverage:
# ******************************************************-
# EndTargetDef:
# Required, ends the Target Parameters
# ******************************************************-
EndTargetDef:
# ******************************************************-
# Required, ends a Color Mapping Definition.
# ******************************************************-
# EndMappingDef:
```

3. Once you have finished editing the `pie1.cfg` file, save the file.
4. To create the Color Mapping Table object (MODCA) file that will be used when you submit the job for printing, run the **cmt** Utility as the input file (**-i** flag):

   ```
   cmt -i pie1.cfg -o pie1.set
   ```

   This command produces a **pie1.set** output file.
5. To verify your color mapping table values, or to create a configuration file from a MODCA object file, specify the **cmt** utility again, using the newly created **pie1.set** as the input file and **pie2.cfg** as the output file:

   ```
   cmt -i pie1.set -o pie1.cfg
   ```

   Note: **pie2.cfg** is specified so that you don't overwrite the original file. You can then compare **pie1.cfg** to **pie2.cfg**.
6. Before you try to use this color mapping table to submit a job to InfoPrint Manager, verify that the source file contains the intended values. In this case, the **pie1.cfg** file should resemble this example, where the source definition represents the input file (in this case AFP) that you are converting to the target definition (an RGB color value of green):

```
BeginMappingDef:
BeginSourceDef:
ColorSpace: GOCA
ColorValue: 11
ObjectType: GOCADATA
EndSourceDef:
```

```
BeginTargetDef:
ColorSpace: RGB
ColorValue: 12 252 42
EndTargetDef:
EndMappingDef:
```

7. Once you have verified the contents of the source file, see the section below.

Submitting jobs using a color mapping table

To submit a job using the color mapping tables for spot color, you must make sure that you also specify a resource-context attribute so that the server knows the location of the color mapping table. Once you have configured a Color Mapping Table, you must have defined a logical destination (named `prt1` for this example) to which you can send the job.

Submitting jobs through the Linux command line

Once the logical destination has been defined, you can submit a job to an InfoPrint Manager printer by specifying these from the command line:

Command Values	Meaning
`pdpr -d prt1`	Print Command with logical destination name.
`-t <jobname>`	Specifies the name that you want to assign to the job (optional).
`-x resource-context=/usr/lpp/psf/config`	Explicit path name where the AFP file (in this case `pie1`), is located on your InfoPrintLinux server.
`-x color-mapping-table=pie1.set`	Color Mapping Table that you created in Creating a color mapping table, p. 365 for printing this file.
`<path and file to print>`	Explicit path to the AFP file that you want to print.

Note

`/usr/lpp/psf/config` is a directory that is created automatically.

Example:

```
pdpr -d ld1 -x resource-context=/usr/lpp/psf/config -x
   color-mapping-table=pie1.set /home/user1/test.afp
```

This command will print the input file **test.afp** located in the directory **/home/user1/** using the color mapping table **pie1.set** located in **/usr/lpp/psf/config** to the logical destination **ld1**.

-x attributes

Submitting a job with **-x** attributes overrides any actual destination configuration settings. Actual destination settings only take effect by default if no parameters are passed to InfoPrint Manager when submitting a job. The hierarchy is:

1. **-x** attribute upon job submission
2. Job and document defaults; color-mapping-table attribute
3. Actual destination color-mapping-table attribute

Configuring an actual destination

When configuring an actual destination with the InfoPrint Manager Administration GUI, it is important to include the name of the color mapping table. Use this procedure to configure an actual destination to use the appropriate color mapping table:

1. From the **InfoPrint Administration Servers** main window, enable the appropriate actual destination (in this case, `NewP`, as shown below) by right-clicking the actual destination and then selecting **Enable**.
2. Ensure that the **Color mapping table** (`pie1.set`) has been specified on the **AFP Color** tab of the **Printer Properties** notebook and that the **Location of resources** (`/usr/lpp/psf/config/`) has been specified on the **AFP Resources** tab.
3. Submit your job to InfoPrint Manager, using either the command line or the InfoPrint Select client.

   Using the InfoPrint Select client to submit a job

   There are 3 ways to designate a color mapping table when using InfoPrint Select:

   1) Use the **Logical destination** to designate the color mapping table. Specify the **color-mapping-table** attribute in the **Job and document defaults** associated with the logical destination to which you are sending the job.
   2) Use the **Actual destination** to designate the color mapping table. Specify the **color-mapping-table** attribute of the Actual destination to which you are sending the job.
   3) Use the **File** → **Print** method to designate the color mapping table. If you are using a newer version of InfoPrint Select, a **Job settings dialog** appears after the **`Print`** button is clicked on the **Print** window. Specify the name of the color mapping table by clicking the **`Other`** tab of **Job settings dialog**. Enter **`color-mapping-table=pie1.set`**, where **`pie1.set`** is the name of the color mapping table.

Setting up high speed printers for spacing

In InfoPrint Manager, moving from page to page in a job is called *spacing*:

- *Forward spacing* is moving to a point later in the job (such as skipping from page 10 to page 15).
- *Backspacing* is moving to a point earlier in the job (such as backing up from page 40 to page 20).

For spacing to work effectively on InfoPrint 4000 and InfoPrint 4100, you must first enable the printer to work with the Simple Network Management Protocol (SNMP) and allow SNMP to configure the printer. These settings allow you to use the InfoPrint Manager GUI to stop a printer very quickly (in ten seconds or less) when you need to, instead of having to wait for all of the pages of a job that have been buffered to print.

Enabling SNMP on an InfoPrint 4000 printer that uses the operator console

The Operator console is the standard console for the InfoPrint 4000.

1. On the display/touch screen on the Operator console, open the **Configure** pull-down menu.
2. Select **Remote Access**.
3. Under **Select a Remote Access**, select **SNMP**.
4. Under **Enabled**, select **Yes**.
5. Under **Allow to Configure**, select **Yes**.
6. Touch **OK** to close the window.
7. You will be prompted to restart the printer so that these settings can take effect. Restart the printer, and wait for it to power back up before you continue.
8. Disable the printer and enable it again to refresh the printer configuration information in the server.
    1. Start the InfoPrint Manager Administration GUI.
    2. In the GUI, find the InfoPrint Manager printer (actual destination) that corresponds to this printer and select it.
    3. Use the GUI to disable the printer.
    4. Use the GUI to enable the printer again.

Enabling SNMP on a printer that uses the Enhanced Operator Console

The Enhanced Operator Console is the standard console for the InfoPrint 4100 and is available as an upgrade for the InfoPrint 4000.

1. On the Enhanced Operator Console, touch **Printer definition** at the top of the screen.
2. On the left side of the screen, touch **Remote Access**.
3. In the **Communities** list, select the name of the community you will be using to access the printer and verify the settings. Make sure that the community has **Write** access to the printer.
4. Select the **SNMP Agent Enabled** and **Allow SNMP to Configure Printer** checkboxes, so they both say **Yes**.
5. Touch **OK** to save the settings.
6. You might be prompted to restart the printer so that these settings can take effect. Restart the printer, and wait for it to power back up before you continue.
7. Disable the printer and enable it again to refresh the printer configuration information in the server.
    1. Start the InfoPrint Manager Administration GUI.
    2. In the GUI, find the InfoPrint Manager printer (actual destination) that corresponds to this printer and select it.

3. Use the GUI to disable the printer.
4. Use the GUI to enable the printer again.

Using InfoPrint Manager with the InfoPrint 5000 to print AFP

This procedure takes you through steps to submit different kinds of jobs to the InfoPrint 5000 using different methods. This list describes these scenarios:

- An Advanced Function Presentation (AFP) document created in a page layout application is submitted through InfoPrint Manager for printing to the InfoPrint 5000 using the command line. Color images are transformed offline using the Linux command line.
- A PostScript document is submitted for printing to the InfoPrint 5000 using InfoPrint Submit Express.
- A document created in a Windows application is submitted through InfoPrint Select for printing to the InfoPrint 5000.
- A document is submitted for printing to the InfoPrint 5000 using Download for z/OS.

Setting up

The scenarios in this procedure assume that you have completed these tasks before submitting the job to the printer, if they are applicable.

- Built a data stream, p. 371
- Transformed color images, p. 371

See Configuring InfoPrint Manager with the InfoPrint 5000 to print AFP, p. 19 for specific information about configuring InfoPrint manager for Linux with the InfoPrint 5000.

Building a data stream

Build your AFP data stream by using any AFP statement generator or application. See "Understanding Form Definitions and Page Definitions" in *Print Services Facility for z/OS: User's Guide* for more information about AFP architecture.

Build your PostScript data stream with any application that generates PostScript, such as Microsoft Word with a PostScript driver. InfoPrint Manager sends PostScript data streams to be transformed and RIPped at the server.

Transforming color images

Color images are transformed automatically at the server when you submit the job to the printer. However, you can transform your images separately by running transforms from the Linuxcommand line without printing the generated output before submitting the job. Jobs run more quickly if the images have already been transformed.

InfoPrint Manager provides transform programs for most data streams, such as GIF, PS, and PDF. The transforms convert the incoming data stream into the AFP format. For example, to transform the file `mypicture` to AFP using the **gif2afp** transform, enter this command:

```
gif2afp mypicture
```

By default, these transform programs produce monochrome FS10 output. To produce finely-tuned color FS45 output, see the list of flags or options and values for each transform in "InfoPrint Object Attributes" in the *RICOH InfoPrint Manager: Reference*.

Note

For best results, transform TIFF, JPEG, and GIF images from theLinux command line.

Using and changing color and resource attributes

You need to decide which color and resource attributes you use or need to change when you submit your job. The color and resource attribute-value pairs that are specific to the InfoPrint 5000 include these:

- **color-profile, p. 372**
- **color-profiles-supported, p. 373**
- **color-bits-per-plane, p. 373**
- **color-bits-per-plane-supported, p. 373**
- **color-rendering-intent, p. 373**
- **color-toner-saver, p. 374**
- **color-toner-saver-supported, p. 374**
- **resource-context-presentation-object-container, p. 374**
- **presentation-object-container-extraction-mode, p. 374**

See "InfoPrint Object Attributes" in the *RICOH InfoPrint Manager: Reference* for more information about these attributes.

Using color attributes

The color attributes can be modified from the command line using the **pdset** command or in the InfoPrint Manager Administration GUI. See "InfoPrint Object Attributes" in the *RICOH InfoPrint Manager: Reference* for the correct command syntax when using the command line.

color-profile

Use this attribute to specify the color profile, if any, to be used by PostScript jobs. This attribute can be equal to:

- **SWOP**
- **Euroscale**
- **none**

The default value is **none**, which means that the color rendition will be dependent on the output device.

The title of this attribute in the InfoPrint Manager Administration GUI is **Color profile**.

color-profiles-supported

Use this attribute to identify which color profiles can be used by the actual destination when printing a document. This attribute can be equal to one or more of these:

- **SWOP**
- **Euroscale**
- **none**

The default values for the InfoPrint 5000 are **SWOP**, **Euroscale**, and **none**. For all other actual destinations the default is equal to **none**.

The title of this attribute in the InfoPrint Manager Administration GUI is **Color profiles allowed**.

color-bits-per-plane

This attribute specifies whether the transform is to produce monochrome (FS10) or color images (FS42 or FS45). You can set this attribute equal to:

- **0–bits** (monochrome FS10)
- **1–bit** (color FS42)
- **8–bits** (color FS45)

The default for InfoPrint 5000 is **8–bits**. For all other actual destinations, the default is **0–bits**.

Note

If the attribute is set to **1–bit** for a PostScript data stream it will fail.

The title of this attribute in the InfoPrint Manager Administration GUI is **Color bits per plane**.

color-bits-per-plane-supported

This attribute specifies the types of images supported on the actual destination when printing a document. This attribute can be equal to one or more of these:

- **0–bits** (monochrome FS10)
- **1–bit** (color FS42)
- **8–bits** (color FS45)

The default values for InfoPrint 5000 are:

- **0–bits** (monochrome FS10)
- **1–bit** (color FS42)
- **8–bits** (color FS45)

The title of this attribute in the InfoPrint Manager Administration GUI is **Color bits per plane allowed**.

color-rendering-intent

While the InfoPrint 5000 produces vibrant full-color images, including images that contain gradients and shading, it is not able to reproduce every shade of every color. The **color-rendering-intent** attribute determines what InfoPrint Manager does when it encounters a shade in a PostScript image that the InfoPrint 5000 cannot reproduce exactly.

There are two possible values for **color-rendering-intent**:

Relative

When InfoPrint Manager encounters a color that InfoPrint 5000 cannot reproduce, it substitutes the closest color value that it *can* reproduce.

Perceptual

When InfoPrint Manager encounters a color that InfoPrint 5000 cannot reproduce, it adjusts all of the colors in the image so that they maintain their color relationships to each other. The result is an image that is pleasing to the eye, but that might not look exactly like the original image.

The default value is **Relative**.

You can set the *color rendering intent* value on a print job, a default document, or on an actual destination using either the command line or the InfoPrint Manager Administration GUI. The title of this attribute in the InfoPrint Manager Administration GUI is **Color rendering intent**.

color-toner-saver

When cyan, magenta, and yellow are combined in a certain proportion, the result approximates black. When the color-toner-saver attribute is set to **yes**, the proportion of cyan, magenta, and yellow are reduced and substituted with black toner, which reduces the amount of color toner used. You can enter the values:

- **true** or **yes**
- **false** or **no**

The default value is **false**, meaning that you do not want to use this attribute.

The title of this attribute in the InfoPrint Manager Administration GUI is **Color toner saver**.

color-toner-saver-supported

This attribute identifies the values that can be specified for color-toner-saver.

The title of this attribute in the InfoPrint Manager Administration GUI is **Color toner saver allowed**.

Using resource attributes

The resource attributes can be modified from the command line using the **pdset** command or in the InfoPrint Manager Administration GUI. See "InfoPrint Object Attributes" in the *RICOH InfoPrint Manager: Reference* for the correct command syntax when using the command line.

resource-context-presentation-object-container

This attribute adds paths to the directory that InfoPrint searches for presentation object containers (also called data object resources), such as EPS or PDF pages, for the actual destination.

The title of this attribute in the InfoPrint Manager Administration GUI is **Location of presentation object containers**.

presentation-object-container-extraction-mode

This attribute specifies the degree of optimization tried by color transforms. The possible values are:

- **ignore** (generates the document without resources by ignoring presentation object containers)
- **inline** (identifies and extracts the resources in a document for use)

The default value is **ignore**.

Note

Inline transforms always use an extraction mode of **ignore**.

The title of this attribute in the InfoPrint Manager Administration GUI is **Presentation object container extraction mode**.

Using the InfoPrint Manager Administration GUI to change color and resource attributes

The InfoPrint Manager Administration GUI is an interface for managing InfoPrint Manager jobs, queues, logical destinations, and actual destinations, such as printers. It runs on multiple platforms, so you can use it either from a Windows workstation, anLinux workstation, or directly on theLinux InfoPrint Manager server console.

From the server or from remote work stations, operators can use this interface to monitor all jobs and printers in the system and to route jobs appropriately to increase enterprise-wide printing productivity. With the InfoPrint Manager Administration GUI, you can change the color and resource attributes.

Starting the InfoPrint Manager Administration GUI

To open the InfoPrint Manager Administration GUI from the Windows desktop, double-click any of the **InfoPrint Administration** icons.

You can start the InfoPrint Manager Administration GUI from an **terminal**window. For example, to open an advanced view, type `ipguiadvadm` at the Linux command line. You can also use the basic (ipguibasicadm) view.

Changing the attributes

Now that you have opened one of the views of the InfoPrint Manager Administration GUI, you can change the attributes in the actual destination. Follow this procedure to set the attributes using the GUI.

1. Click your InfoPrint 5000 actual destination.
2. Select **Options** → **Customize**.
3. In the **Customize** dialog, click the **Printer** tab.
4. In the **Action** column, find **Change Tuning**, **Change AFP Resources**, and **Change AFP Color**, and select the check boxes in the **Menu** column next to them.
   This step adds these tasks to the **Printer** menu. If you want to add other tasks to the **Printer** menu or to any other menu, you can do it now or you can wait until later.
5. Click **OK** to apply the settings and close the dialog.
6. Select **Printer** → **Change AFP Resources** to open the **Change AFP Resources** dialog. These fields allow you to specify where resources, such as fonts and presentation object containers, are located. See Identifying where to store AFP resources, p. 21.

7. In the **Change AFP Resources** dialog, find the **Location of presentation object containers** field and type the path where your presentation object containers are stored. If they are stored in more than one location, list all of the paths separated with colons (:), for example:

```
/resources/containers:/dept123/priv-pres-obj- containers:/dept123/
public-pres-obj-containers
```

8. Fill in any other fields that are appropriate. For example, if you use form definitions, fill in the **Location of form definitions** field.
9. Click **OK** to apply the settings and close the dialog.
10. Select **Printer** → **Change AFP Color** to open the **Change AFP Color** dialog. These fields allow you to modify the color attributes in your job. See Using color attributes, p. 372 for more information.
11. Click **OK** to apply the settings and close the dialog.
12. Select **Printer** → **Change Tuning** to open the **Change Tuning** dialog. These fields allow you to modify resource attributes in your job. See Using resource attributes, p. 374 for more information.
13. Click **OK** to apply the settings and close the dialog.

Sending an AFP job to the printer using the Linux command line

In this example, you have created a document using an AFP statement generator. This job contains color images that you decide to transform offline. You now want to submit that document through InfoPrint Manager to the InfoPrint 5000 using the command line.

Setting up assumptions

This scenario assumes that you have completed these tasks:

- Built an AFP data stream
- Identifying where to store AFP resources, p. 21
- Transformed color images

See Setting up, p. 371 for more information.

Using Linux commands

On any Linux system, the InfoPrint Linux client allows authorized users to use these InfoPrint commands:

- **pdls**
  Lists selected attribute values for a job, destination, or other InfoPrint object.
- **pdpr**
  Submits a job to a logical destination.
- **pdq**
  Queries the status of selected jobs, of all jobs submitted to a logical destination, or of all jobs assigned to an actual destination.

See *RICOH InfoPrint Manager: Reference* or the associated man pages for a complete list of flags and attributes to use with these commands.

Submitting the job

Complete these steps to submit the job.

1. To determine whether the logical destination `IP5000-ld` is ready to receive jobs, type:

```
pdls -c destination -r "enabled" IP5000-ld
```

   You will see this message if the logical destination is enabled:

```
enabled=true
```

2. To submit the file `MyJob` to the logical destination `IP5000-ld` and request two-sided printing and the **color-toner-saver** attribute, type:

```
pdpr -d IP5000-ld -x "sides=2 color-toner-saver=true" MyJob
```

   Note

   The page size and orientation that is specified in the job and in InfoPrint Manager must match that specified at the printer. Otherwise, the job will print on the page size and with the orientation specified at the printer.

3. To query all the jobs that you submitted to logical destination `IP5000`, type:

```
pdq -d IP5000-ld
```

In the previous examples *–d* refers to the logical destination and *–x* refers to the attribute string that contains the attribute-value pairs you want to include.

You can use these attributes from the Linux command line or you can change them in the InfoPrint Manager Administration GUI:

- **color-bits-per-plane**
- **color-bits-per-plane-supported**
- **color-rendering-intent**
- **color-toner-saver**
- **color-toner-saver-supported**
- **resource-context-presentation-object-container**

See Using and changing color and resource attributes, p. 372 for details on when and how to use these attributes.

Sending an AFP job with color images to the printer using InfoPrint Submit Express

You have created an AFP document using a page layout application and are using InfoPrint Submit Express to send it to the InfoPrint 5000.

Note

This example uses InfoPrint Submit Express for Windows.

Setting up assumptions

This scenario assumes that you have completed these tasks:

- Built a data stream, p. 371
- Installed InfoPrint Submit Express

See Setting up, p. 371 for more information.

Submitting the job

When using InfoPrint Submit Express to create the job ticket, you must consider a number of things to prepare a job for printing. Primarily, however, during this phase of the process, you will identify and organize the printable files and other resources that make up the job. As you complete the job ticket, Submit Express provides a format in which you can specify the details necessary to process a job effectively. You generate the printable files that you submit to InfoPrint Manager file by file from the application that you use for authoring.

Note

Use the **Help** menu from the Job Ticket window or the **Help** button on the dialogs to guide you.

Creating a new job ticket

In this example, you will create a new job ticket named `MyJob.jtk`. You adjust color attributes and then submit the job.

Note

When you use InfoPrint Submit Express to send a job to the printer, you do not need to have previously created a form definition. If you have a form definition it overrides the attributes you specify in InfoPrint Submit Express.

1. Start InfoPrint Submit Express. The **Job Ticket** window displays.
2. Fill in all the fields with appropriate information.
3. Insert the files into your job ticket that you want to include in the job. Click **Help** → **Help...** if you have questions.

   To insert new files into the document, click **Item** → **Insert File** on the menu bar. The Select files dialog box displays. Choose the file(s) you want to add to the document from the Select files dialog box and click **Select**.
4. From the **Print** tab, select `InfoPrint 5000` from the **Printer type** drop-down list and specify the layout you want in the **Layout** field.
5. Specify page alignment on the **Align** tab.

Note

You must specify the same page size and orientation at the printer as you do in the job, otherwise the page size and orientation specified in the printer will override that specified in the job.

6. Choose the values for the color attributes on the **Look** tab. You can use these attributes in InfoPrint Submit Express:
   - **Color rendering intent**
   - **Color profile**
   - **Color bits per plane**
   - **Color toner saver**

   See Using and changing color and resource attributes, p. 372 for details on when and how to use these attributes.
7. Click **Advanced**, find **RIP for**, and select **InfoPrint 5000** from the drop-down list.
8. Save the job ticket by clicking **Ticket** → **Save As...**.
9. Submit the job by clicking **Ticket** → **Submit Job**.

See the "Using InfoPrint Submit Express Client" section in *RICOH InfoPrint Manager for Linux: Getting Started* or the RICOH Software Information Center at https://help.ricohsoftware.com/swinfocenter/.

Sending a job using Download for z/OS

You have created an AFP document using a page layout application and are using Download for z/OS to send it to the InfoPrint 5000.

Download for z/OS automatically sends data from the job entry subsystem (JES) spool to remote systems using TCP/IP. This eliminates the need for manual print file transfer using File Transfer Protocol. A cooperating print server running on a remote system receives the data sets for printing with InfoPrint Manager for Linux. For most jobs, you can use the same job control language (JCL) that you would use for non-AFP printers, changing only the MVS routing information to direct your job to the InfoPrint 5000.

Note

You can use InfoPrint's Page Printer Formatting Aid (PPFA) to create form definitions and page definitions when creating the data stream. See the RICOH Software Information Center at https://help.ricohsoftware.com/swinfocenter/ for information about how to use PPFA.

If you are using Download for z/OS to submit jobs to the printer, see "Using JCL for Advanced Function Presentation" and "Printing Tasks and Examples" in the *Print Services Facility for z/OS: User's Guide* for specific printing tasks and examples. You can also see *Print Services Facility for z/OS: Download for z/OS* for more information.

Using the InfoPrint 5000 and RICOH Pro VC series models with hot folders from InfoPrint Manager to print PostScript and PDF

You can use hot folders to print PDF and PostScript files on the InfoPrint 5000 printer models (AD1/AD2, AD3/AD4, AD3/AD4-XR3, AS1, AS3, KM3, KM3/MD4, MD3/MD4) and RICOH Pro VC

series models (RICOH Pro VC40000, RICOH Pro VC60000, RICOH Pro VC70000, and RICOH Pro VC80000) with InfoPrint Manager for Linux.

- To submit the file `MyJob` to the logical destination `IP5000-ld` and request two-sided printing and the **color-toner-saver** attribute, type:

```
pdpr -d IP5000-ld -x "sides=2 color-toner-saver=true" MyJob
```

Note

The page size and orientation that is specified in the job and in InfoPrint Manager must match that specified at the printer. Otherwise, the job prints on the page size and with the orientation specified at the printer.

- To query all the jobs that you submitted to logical destination `IP5000`, type:

```
pdq -d IP5000-ld
```

Transferring InfoPrint Manager objects across servers

You can use the InfoPrint Manager Web Administration Interface (Web GUI) to move InfoPrint Manager queues and actual destinations from one server to another, provided both servers are running on the same computer.

Alternatively, you can use the `pdset` command to change the value of the `associated-server` attribute for specific queues and actual destinations. As a result, the queues or actual destinations together with all their dependent objects are removed from the source server and recreated on the target server in a single step.

Important

The `associated-server` attribute is settable using the `pdset` command. When you change the value of this attribute, InfoPrint Manager automatically:

1. Removes the specified object and all its dependent objects from the source server
2. Recreates the object and its dependent objects on the target server
3. Preserves all attribute values and object states

Note

- Ensure that both the source and target servers are running and accessible.
- Ensure that all required resources are available on the target server (for example, transform scripts and resource paths used by transforms).
- All attribute values for the transferred objects are stored in memory and applied when the objects are created on the target server.
- The states of the objects are preserved (for example, a disabled logical destination on the source server will also be disabled on the target server).
- If an object fails to be created on the target server, all objects that were successfully created are automatically deleted, and all objects are restored on the source server.

Dependent objects

When you transfer an object, all dependent objects are automatically transferred with it. The following table shows the dependent objects for each object type.

Object type	Dependent objects
Queue	All logical destinations associated with the queue
Actual destination	None (actual destinations have no dependent objects)
Logical destination	Must be transferred by transferring its associated queue

Transferring an actual destination to a different server (Web GUI)

You can use the InfoPrint Manager Web Administration Interface (Web GUI) to transfer an actual destination/printer and its configuration to another server.

To transfer an actual destination to another active server:

1. Click the **Printers** icon in the left icon bar.
2. Click a printer in the Objects Panel and select **Transfer...**.
3. Or, right-click a printer in the printer details view and select **Transfer...**.
4. In the displayed dialog, select the target server from the list.
5. Click **OK**.

Transferring a queue to a different server (Web GUI)

You can use the InfoPrint Manager Web Administration Interface (Web GUI) to transfer a queue and its configuration to another server.

To transfer a queue to another active server:

1. Click the **Queues** icon in the left icon bar.
2. Click a queue in the Objects Panel and select **Transfer...**.
3. Or, right-click a queue in the queue details view and select **Transfer...**.
4. In the displayed dialog, select the target server from the list.
5. Click **OK**.

4. Operator and User Procedures

- **Finishing options**
- **Other procedures**

Finishing options

Setting up finishing options for IPDS-only printers

Depending on the printer model, you can set up several finishing options. InfoPrint Manager can be set up to support many of these possibilities, including edge stitching (stapling one edge of the print job to create a booklet), saddle stitch (stapling in the middle of the page to create a folded booklet), and corner stapling. These sections describe how to set up finishing options for IPDS printers.

Telling InfoPrint Manager what options the finisher supports

The first step in enabling the finishing options is to let InfoPrint Manager know what options the finisher supports. Follow the steps below to set the appropriate values.

Note

If you are only to submit print jobs that specify a form definition to this printer, you do not have to complete this procedure. You can proceed to Setting up InfoPrint Manager and clients to use finishing options, p. 386.

1. Start the InfoPrint Manager Administration GUI.
2. Find or create the PSF printer (actual destination) that sends jobs to the printer you are setting up.
3. Right-click the printer and select **Properties** from the pop-up menu.
4. In the Printer Properties notebook, click the **Job** tab.
5. Find the **Finishing options allowed** field.

   Note

   If you do not see the **Finishing options allowed** field, click **Show More**.
6. Select all of the finishing options that your finisher supports in the **Possible values** list and add them to the **Values** list.
7. Click **OK**.

Setting up InfoPrint Manager and clients to use finishing options

How you set up to use finishing options varies depending upon what kind of print jobs you create and how you submit them. There are several categories; find the one below that most resembles your situation.

Note

- You cannot set up InfoPrint Manager and clients to use finishing options with the AGSPDL processing engine.

Using the pdpr command

When you submit print jobs from the command line with the **pdpr** command, you can use the **job-finishing** attribute on the **–x** flag to specify finishing options. For example:

```
pdpr -d printername -x"job-finishing=staple-top-left" printfilename
```

You can also use the **form-definition** attribute on the **–x** flag to specify finishing options. For example:

```
pdpr -d printername -x"form-definition=F1name" printfilename
```

For a list of allowed values, proper syntax, and usage guidelines for the **job-finishing** and **form-definition** attributes, see *RICOH InfoPrint Manager: Reference*.

Using a form definition

If you submit your print jobs using a form definition (either by specifying a certain one or by using a default form definition), you can identify the finishing options that you want to apply in the form definition. When you specify printing and finishing options this way, no InfoPrint finishing attributes are necessary to request the finishing options. The print job is sent to the printer and the options are read out of the formdef. If the finisher supports the options chosen, the job should print correctly without any more configuration on your part.

Note

InfoPrint Manager does not check the finishing options you specify in the formdef against the options you set in Telling InfoPrint Manager what options the finisher supports, p. 386. As a result, if the finisher does not support the options, the job will fail.

Using InfoPrint Submit Express

When you submit a job using InfoPrint Submit Express, you can select a variety of options, including finishing. Build your job ticket as you usually would, and specify the finishing options that you want to use on the appropriate tab or page.

Using InfoPrint Select

The InfoPrint Select client lets you specify job-finishing attribute values or a form definition when submitting a job to the server. For more information about using the job-finishing attribute, see *RICOH InfoPrint Manager: Reference*. For instructions on setting attribute values in Select, see the help system.

Using the InfoPrint AFP driver

If you are interested in using the powerful features of AFP in a wide range of environments, you might want to install and use the InfoPrint AFP driver with InfoPrint Select. To install and set up the InfoPrint AFP driver, see "Setting up the AFP driver" in the *RICOH InfoPrint Manager for Linux: Planning Guide*. To specify the AFP driver for your printer, follow the directions in the *RICOH InfoPrint Manager for Linux: Getting Started* guide. Once installed, you can click either the **Printing Preferences** button on Windows systems from the **Print** window, and then use the AFP driver to specify inline form definitions with finishing options, or make use of other AFP features.

PSF printer

If you want to use InfoPrint Select to submit print jobs with finishing options to a PSF printer, you must create a Select printer that uses the specific printer driver for your printer shown here. The driver is installed on your system when you install InfoPrint Select. Use the procedure that is described in the

RICOH InfoPrint Manager for Linux: Getting Started guide for creating a Select printer, modifying it as described below.

1. Follow the procedure through creating a port and selecting an InfoPrint destination.
2. When the **Add Printer** wizard asks you to select a manufacturer and a printer model, do not select anything, just click **Have Disk**.
3. In the **Install from Disk** dialog, click **Browse** and navigate to the directory that you installed InfoPrint Select in. The default directory is `c:\Infoprint`.

   Note

   Because this dialog assumes the the driver is actually on a floppy disk or DVD-ROM, you might receive an error that the drive is not ready. In the error box, click **Cancel**. A Windows **Open** dialog should then open. Use that dialog to navigate to the correct directory.
4. Select the `Ibmprint.inf` file and click **Open**.
5. Select the appropriate driver based on your printer model and click **OK**.
6. In the **Install from Disk** dialog, click **Next**.
7. Complete the wizard as usual.

When you submit jobs to this printer, you can select the finishing option you want to use.

Using Download for z/OS

Download for z/OS is an optional feature of PSF for z/OS that uses TCP/IP to automatically send data sets from the JES spool, without formatting them, directly to InfoPrint Manager for AIX, InfoPrint Manager for Linux or InfoPrint Manager for Windows. InfoPrint Manager receives the data sets into files, formats them, and sends them to the printer. When you submit a print job with Download for z/OS, do one of these:

- Specify a mapping from a JCL keyword to a finishing attribute in the Linux shell script. The selected JCL parameter you specify in z/OS maps to the InfoPrint Manager **pdpr** finishing attribute.
- Use the existing form definition JCL keyword to form definition attribute mapping. The JCL FORMDEF parameter you specify in z/OS maps to the InfoPrint Manager **pdpr** form definition attribute.
- Specify a default form definition or finishing attribute so all jobs have the same finishing option.

Using InfoPrint Manager hot folders

An InfoPrint Manager hot folder is a directory that is associated with a logical destination. When you copy or move a file to the hot folder, InfoPrint Manager automatically submits the file to that logical destination. The default job and document finishing attributes that are set for the logical destination are applied to jobs submitted through the hot folder. You can also specify other finishing attributes or use different values for existing attributes.

Using InfoPrint Manager LPD and an LPR client

The InfoPrint Manager Line Printer Daemon (LPD) is a utility that lets you submit print jobs from other operating systems using a Line Printer (LPR) client. To submit print jobs with finishing to the InfoPrint Manager LPD, you must use an LPR client, such as **lprafp**, that lets you use –o options to specify advanced printing options (for example, form definitions and page definitions) when you submit jobs to InfoPrint Manager. You can either specify finishing attributes on the submission command or use a default job.

Selecting the correct finishing options

In InfoPrint Manager, the finishing options have very descriptive names so that it is easy to select the one you want for your job. However, the names of those options are based on print jobs that use the "portrait" orientation, and as a result are misleading for jobs that printed in the "landscape" orientation. In addition, some of the options are valid if the paper is fed through the printer long edge first (LEF) and not valid if the paper is fed short edge first (SEF), and vice versa. As a result, choosing the correct finishing option can be confusing.

Setting up finishing options for PS/PCL printers

Depending on the printer model, you can set up several finishing options. InfoPrint Manager can be set up to support many of these possibilities, including edge stitching (stapling one edge of the print job to create a booklet), saddle stitch (stapling in the middle of the page to create a folded booklet), and corner stapling. These sections describe how to set up finishing options for printers that can be sent PostScript and PCL.

Telling InfoPrint Manager what options the finisher supports

The first step in enabling the finishing options is to let InfoPrint Manager know what options the finisher supports. Follow the steps below to set the appropriate values.

1. Start the InfoPrint Manager Administration GUI.
2. Find or create the PSF printer (actual destination) that sends jobs to the printer you are setting up.
3. Right-click the printer and select **Properties** from the pop-up menu.
4. In the Printer Properties notebook, click the **Job** tab.
5. Find the **Finishing options allowed** field.

   Note

   If you do not see the **Finishing options allowed** field, click **Show More**.
6. Select all of the finishing options that your finisher supports in the **Possible values** list and add them to the **Values** list.
7. Click **OK**.

Setting up InfoPrint Manager and clients to use finishing options

How you set up to use finishing options varies depending upon what kind of print jobs you create and how you submit them. There are several categories; find the one below that most resembles your situation.

Using the pdpr command

When you submit print jobs from the command line with the **pdpr** command, you can use the **job-finishing** attribute on the **–x** flag to specify finishing options. For example:

```
pdpr -d printername -x"job-finishing=staple-top-left" printfilename
```

You can also use the **form-definition** attribute on the **–x** flag to specify finishing options. For example:

```
pdpr -d printername -x"form-definition=F1name" printfilename
```

For a list of allowed values, proper syntax, and usage guidelines for the **job-finishing** and **form-definition** attributes, see *RICOH InfoPrint Manager: Reference*.

Using a form definition

If you submit your print jobs using a form definition (either by specifying a certain one or by using a default form definition), you can identify the finishing options that you want to apply in the form definition. When you specify printing and finishing options this way, no InfoPrint finishing attributes are necessary to request the finishing options. The print job is sent to the printer and the options are read out of the formdef. If the finisher supports the options chosen, the job should print correctly without any more configuration on your part.

Note

InfoPrint Manager does not check the finishing options you specify in the formdef against the options you set in Telling InfoPrint Manager what options the finisher supports, p. 386. As a result, if the finisher does not support the options, the job will fail.

Using InfoPrint Submit Express

When you submit a job using InfoPrint Submit, you can select a variety of options, including finishing. Build your job ticket as you usually would, and specify the finishing options that you want to use in the **Finishing** field on the appropriate tab.

Using InfoPrint Select

Your enhanced InfoPrint Select client lets you specify job-finishing attribute values or a form definition when submitting a job to the server. For more information about using the job-finishing attribute, see *RICOH InfoPrint Manager: Reference*. For instructions on setting attribute values in Select, consult the help system provided with InfoPrint Select.

The automatic data stream transforms that PSF printers use do not support finishing (stapling), so in general, if you plan to use InfoPrint Select to submit PCL or PostScript jobs to a PSF printer (destination), you cannot specify finishing options when you submit a job, even if the print driver you use supports those options. However, there are other submission methods that will allow you to submit jobs with finishing options using Select.

Using the InfoPrint AFP driver

If you are interested in using the powerful features of AFP in a wide range of environments, you might want to install and use the InfoPrint AFP driver with InfoPrint Select. To install and set up the InfoPrint AFP driver, see "Setting up the AFP driver" in *RICOH InfoPrint Manager for Linux: Planning Guide*.

To specify the AFP driver for your printer, follow the directions in "Installing and configuring the AFP driver" from *RICOH InfoPrint Manager for Linux: Getting Started*. Once installed, you can click either the **Printing Preferences** button on Windows systems or the **Setup** button on Windows systems from the **Print** window, and then use the AFP driver to specify inline form definitions with finishing options, or make use of other AFP features.

Using a PSF other-driver or DFE printer

You can use the **document-formats-ripped-at-destination** actual destination attribute to prevent the automatic data stream transform from occurring. See *RICOH InfoPrint Manager: Reference* for details.

Using a CUPS printer

A CUPS DSS printer does not transform the data stream using InfoPrint Manager available transforms. This type of destination uses fewer processing resources but provides fewer functions.

Any other destination type

If you want to use InfoPrint Select to submit print jobs to a different destination type (such as a BSD or IPP printer), the only way to request finishing options on PCL and PostScript jobs is to use default jobs. A *default job* is a list of attributes that is applied to every job that a particular logical destination receives. You can use the InfoPrint Manager Administration GUI to modify the default job for the logical destination that InfoPrint Select sends jobs to so that it requests the appropriate finishing options.

If you use default jobs, you will need a different logical destination for each finishing option. Every job that you send to a particular logical destination will be finished the same way. For example, if you set the **Finishing options** attribute of the default job to **staple-top-left**, every job that is submitted to that printer will have a staple in the top left corner. In addition, a logical destination can only have one default job, and a default job can only have one finishing option set.

Using Download for z/OS

Download for z/OS is an optional feature of PSF for OS/390 that uses TCP/IP to automatically send data sets from the JES spool, without formatting them, directly to InfoPrint Manager for AIX, InfoPrint Manager for Linux or InfoPrint Manager for Windows. InfoPrint Manager receives the data sets into files, formats them, and sends them to the printer. When you submit a print job with Download for z/OS, do one of these:

- Specify a mapping from a JCL keyword to a finishing attribute in the Linux shell script. The selected JCL parameter you specify in OS/390 maps to the InfoPrint Manager **pdpr** finishing attribute.
- Use the existing form definition JCL keyword to form definition attribute mapping. The JCL FORMDEF parameter you specify in OS/390 maps to the InfoPrint Manager **pdpr** form definition attribute.
- Specify a default form definition or finishing attribute so all jobs have the same finishing option.

Using InfoPrint Manager hot folders

An InfoPrint Manager hot folder is a directory that is associated with a logical destination. When you copy or move a file to the hot folder, InfoPrint Manager automatically submits the file to that logical destination. The default job and document finishing attributes that are set for the logical destination are applied to jobs submitted through the hot folder. You can also specify other finishing attributes or use different values for existing attributes. For instructions on using hot folders see "Setting up hot folders to submit jobs" in *RICOH InfoPrint Manager for Linux: Getting Started*.

Using InfoPrint Manager LPD and an LPR client

The InfoPrint Manager Line Printer Daemon (LPD) is a utility that lets you submit print jobs from other operating systems using a Line Printer (LPR) client. To submit print jobs with finishing to the InfoPrint Manager LPD, you must use an LPR client, such as **lprafp**, that lets you use –o options to specify advanced printing options (for example, form definitions and page definitions) when you submit jobs to

InfoPrint Manager. You can either specify finishing attributes on the submission command or use a default job.

Selecting the correct finishing options

In InfoPrint Manager, the finishing options have very descriptive names so that it is easy to select the one you want for your job. However, the names of those options are based on print jobs that use the "portrait" orientation, and as a result are misleading for jobs that printed in the "landscape" orientation. In addition, some of the options are valid if the paper is fed through the printer long edge first (LEF) and not valid if the paper is fed short edge first (SEF), and vice versa. As a result, choosing the correct finishing option can be confusing.

Other procedures

Adding an Internet Printing Protocol (IPP) gateway printer to your desktop

The Internet Printing Protocol (IPP) is a data transport protocol that lets you print your documents to any IPP-enabled printer whose IP address you know. You can use IPP to send print data over both LANs and the Internet, so you can send print jobs to a network printer from your desk, your home office, or anywhere that you find an Internet connection. Not all printers are currently enabled to work with IPP. However, InfoPrint Manager provides an IPP gateway (a mechanism that lets you use IPP) even if your printers are not IPP-enabled.

Note

You must work with your InfoPrint Manager Administrator to have the IPP gateway enabled through the IPMMI.

An **Internet Printing Protocol (IPP) gateway printer** is a printer that you create on a client workstation using IPP client software. Users submit print jobs to that printer, and they are sent to InfoPrint Manager through the InfoPrint IPP gateway. The IPP gateway can then send the job on to any InfoPrint printer, even those that are not IPP-enabled.

To be able to print through an IPP gateway, you must first have IPP client software installed on your workstation or laptop. IPP client software is available from various printer vendors; check with your printing system administrator to see which IPP client your company is using. Once you have the IPP client software installed, to set up to print through the IPP gateway:

1. Ask your printing system administrator for the address of the printer that you want to print to.
2. Follow the instructions provided with your IPP client software for adding a printer to your desktop. When you are asked for the address of the printer, enter the address that you got in step 1.
3. Once the printer is added, you can print to it just as you would any other printer.

Submitting a job to a DocuTech Model 135 printer

To send jobs to your networked DocuTech system, you must first create a BSD destination on your Linux server. Once this is done, you can submit the job through InfoPrint Manager and release it for printing using your local DocuTech procedures.

Create a BSD destination for the DocuTech 135 printer

Using the InfoPrint basic administrator's GUI, do these steps:

1. Drag the BSD template from the Types column in the template area, and drop it on the Actual Destinations pane. You will be prompted to select the server that will contain the actual destination.
2. Fill in the required information fields. Physical printer name is a name you determine, and the command will be as follows: **/usr/lpd/rembak -S docu1 -P docutech135** where **-S** is the name of the remote print server where the job will be sent, and where **-P** is the name of the queue on the remote print server where the job will be sent.
3. Select the **OK** button. InfoPrint creates a new actual destination using the information you supplied. It also creates a new logical destination and new default job. The new default job is associated with the new logical destination and requests the actual destination.
4. Enable the actual destination and logical printer.

Submit a job to the Docutech 135 printer

Use a familiar print submision mechanism such as InfoPrint Select or the **pdpr** command to submit the job to the logical destination you just created.

For example, **pdpr -pyuma-lp /bin/test.ps** could be used to route a test job to a logical printer named yuma-lp.

Note

The job will appear briefly in the jobs window and will be spooled to the Docutech server.

Release the job from the Docutech 135 server interface and printer console

To release the job from the Docutech 135 server interface and printer console, follow these instructions:

1. Access the list of queued jobs, and the file you submitted should display in the refreshed list. From the server window, release the job for printing. Job status for each released job will display as the file is released.
2. Now on the DocuTech console, open the printer queue by clicking on its icon, and open the Job File list. The new job appears in the Job File list.
3. Click on the job and move or copy the file to the printer queue. Your InfoPrint Manager job will then print on the DocuTech printer.

Supporting status Information

This section provides details on how to support status information for several types of printers.

Network printers

If you have InfoPrint network printers in your InfoPrint Manager environment, be sure to install the current version of the InfoPrint network printer drivers to take advantage of InfoPrint Manager's messaging support.

HP printers

If you have HP printers that you have already configured in InfoPrint Manager, you will need to change the **destination-command** attribute to use the **pioinfo** backend program rather than the **piojetd** backend program.

The necessary command is:

```
pdset -cde -x "destination command='/usr/lpd/pio/etc/pioinfo ipaddr'"
```

where *ipaddr* is the ip address or host name of the printer.

Stopping, restarting, or pausing a job that is currently printing

Use this procedure to stop printing a job and start printing it again or to stop printing a job, move it to a different printer, and start printing again. If you are printing an AFP print job, you can also use this procedure to stop printing a job and start it again from any page in the job, not just from the beginning.

This procedure does NOT stop the printer; it only stops the job you select.

Once the job is stopped, the next job in the queue starts to print. If you need to stop the printer (for example, to replace toner), use the procedure Stopping a printer to carry out scheduled maintenance, p. 400.

Note

- If you need instructions on completing any of these steps, see the online help in the InfoPrint Manager GUI.

When you use the Stopping the job, p. 392 procedure, InfoPrint Manager considers the job cancelled and does not collect any checkpoint information about the job. Unless the operator changes the page range before releasing the job to print again, the job restarts printing at page 1. Be sure to remember any n-upping associated with the job when selecting the page range so that the pages are placed into the correct partition in relation to each other. You might want to use this procedure when you realize that a job has started to print on the wrong form.

Stopping the job

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. In the GUI, find the job that you want to stop and select it.
3. Click **Job** → **Hold**.

   Note

   If the **Hold** item does not appear in the **Job** menu, use **Add/Remove Menu Items** to add it.
4. Click **OK**.

   The job stops printing, the icon changes to black, and the job is listed in the job window of the main InfoPrint Manager GUI window.

Restarting the job

Use this procedure to do one of the following:

- Print the entire job on the same printer it was printing on before.
- Print the entire job on a different printer.
- Print part of the job (such as page one to page fifty, or page twelve to the end).

  Important

  – When you use **Job** → **Hold**, the job information resets to the beginning of the job. Checkpoint information is not stored.

  – You can only start printing in the middle of the job if the job is transformed into AFP before it gets printed. You cannot print part of an ASCII, KGL, or metacode; you have to print the whole job.

  – You can only start printing in the middle of the job if the job has only one printable document. If it has more than one printable document, you must either print the whole job or use spacing to skip a section of the job. See the Moving backward or forward in a job that is currently printing, p. 398 for instructions on spacing.

    If you submitted this job using DPF from a host system, you can start printing from any page in the job, but InfoPrint Manager ignores any value you enter for the end page. You will always print to the end of the job.

    Note

    Spacing only works if you send the job to a PSF DSS printer.

To restart the print job:

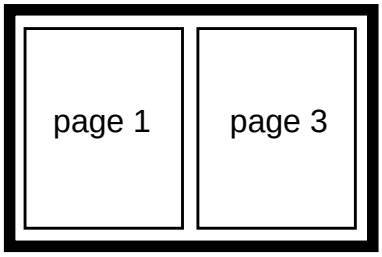
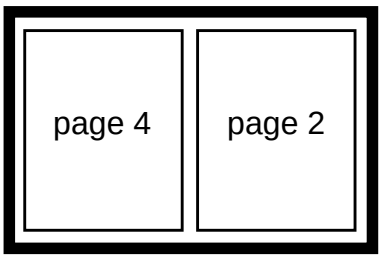
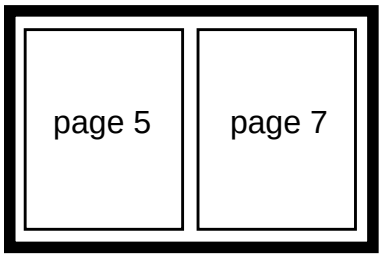
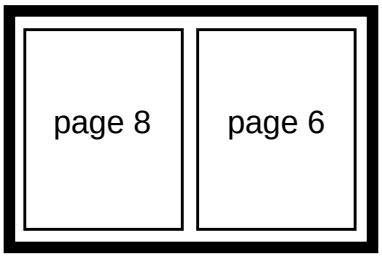
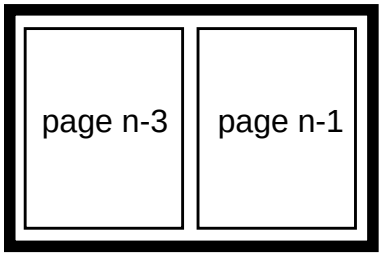
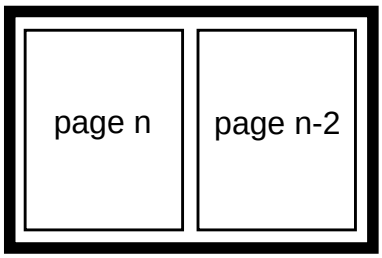
1. To print the entire job on the same printer it was printing on before, select the job in the job window and click **Job** → **Release**. The job will start printing from the first page.
2. To print the entire job on a different printer, skip to **Before you Continue** between steps 8 and 9.
3. To print part of the job, figure out what page you want to start printing from.

Important

If you do not want to start printing from the first page of the job, be careful when you select which page to start with. Duplex and n-up jobs can be confusing because you have to start printing with the **first** page on the **front** side of the sheet of paper. If you do not select the correct start page, the job will print, but the sequence will probably be wrong.

For example, if you are printing a 2–up duplex job, each sheet of paper actually has four pages printed on it, like this:

2–up duplex print job: front and back views

	Front of sheet	Back of sheet
Sheet 1		
Sheet 2		
Last sheet		

Since you have to start printing this job at the first page on the front side, you can select page one or page five (or any other page that would fall in that position). If you select a different page, the sequence of pages will be wrong, and the job won't print correctly.

4. Decide what page you want to stop printing on (if you do not want to print all the way to the end of the job).

   You do not have to be as precise on the last page to print; the last page can be in any location on the sheet of paper.

5. Select the job that you want to print.

6. With the job selected, click **Job** → **Change Page Range to Process**.

   If you do not see the **Change Page Range to Process** item in the **Job** menu, use **Add/Remove Menu Items** to add it.

7. In the **Change Page Range to Process** dialog, select **Range of Pages** and type the start page and end page you figured out earlier.

   Note

   If you submitted this job using Distributed Print Facility (DPF), InfoPrint Manager ignores the end page that you specify. DPF jobs always print from the start page that you specify to the end of the job.

8. Click **OK**.

   Before you continue:

   You should only move a job to an InfoPrint printer that uses the same destination support system (DSS) as the one the job was originally submitted to. If you move the job to a printer that uses a different DSS and the new printer does not support all the attributes of the job, the job fails to print. To check which DSS your InfoPrint printer uses:

   1) Select the printer in the tree view of the Main window of the InfoPrint Manager GUI.
   2) See if the **Type** column appears in the details view. The **Type** column displays the DSS that the InfoPrint printer uses.
   3) If you do not see the **Type** column, use **Add/Remove menu items** to add it.

   Continue with step 9.

9. Select the job again and click **Job** → **Move job**.
10. In the **Move Jobs** dialog, select the printer to print the job on. You can select the same printer that the job was printing on before or a different one.

    When you release the job in step 12, if the job is accepted, it will go back into the queue and wait to be printed. Depending on how many jobs are ahead of it in the queue, it might take a while for the job to print. To print the job right away, you can either move it to the top of the queue (so it prints next) or you can change its priority (so it prints sooner, but not necessarily next).

11. To move the job to the top of the queue, select the job in the InfoPrint Manager GUI and use the **Make Job First** task. To change the priority of the job, use the **Change Priority** task.
12. Select the job in the job window and click **Job** → **Release**.

    Note

    If the task you want to use does not appear in the menu, use **Add/Remove menu items** to add it.

Pausing a job that is printing

When you use this procedure, InfoPrint Manager stops the job, verifies if the paused job is currently printing, and collects checkpoint information about the job if it is. The job shows on the GUI as **Paused**. Assuming that the operator does not take actions that invalidate the checkpoint information when the job is resumed, it should restart at the checkpoint boundary. If the checkpoint information is invalidated, the job restarts with page 1 unless the operator changes the page range before resuming the job. An example of an action that would invalidate the checkpoint information is changing job attributes, so that the job would have to be re-ripped before printing.

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. Find the job that you want to pause and select it.

3. Click either **Job** → **Pause** → **Now** or **Job** → **Pause** → **After current copy**.

   Note

   If the **Pause** item does not appear in the **Job** menu, use **Add/Remove Menu Items** to add it.

4. Click **OK**.

   The job stops printing, the icon changes to black, and the job is listed in the job window of the main InfoPrint Manager GUI window.

5. To resume printing this job, click **Job** → **Resume**.

Important

- If the printer has a functional problem such as a paper jam, the print data in the IPDS printer buffers is no longer printed. The printer does not process the **Pause** command until the data already sent has been processed. This situation might be viewed as a Web GUI freeze.

  To solve this issue, you can either:

  – Resolve the functional problem and then the printer resumes printing.

  or:

  – Send a **Hold** command instead of **Pause**, because the **Hold** command discards the IPDS buffers and puts the job in a hold state in InfoPrint Manager. For more information, see Stopping the job, p. 392.

Resubmitting a retained job for reprint

Use this procedure to resubmit retained jobs for reprinting.

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. In the **Retained Jobs** area, select the job you want to resubmit for reprinting.

   Note

   To prevent the job from printing immediately after it gets in the queue, see Stopping the job, p. 392.

3. Click **Job** → **Move Job** to open the **Move Job** dialog.

   Note

   If the **Move Job** item does not appear in the **Job** menu, go to **Add/Remove Menu Items** to add it.

4. In the **Move Job** dialog, select a destination from the list. You can select the same printer the job was printing on before or a different printer.
5. Click **OK** to resubmit the job to the selected destination. If the job is accepted, it gets in the queue and waits to be printed.

   Note

   If the job was previously stopped, see Restarting the job, p. 392.

Pausing and restarting an actual destination

Use these procedures to pause and restart an actual destination. You might want to pause an actual destination when you are printing a long job, so that you can perform maintenance on the printer.

Pausing an actual destination

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. In the GUI, find the actual destination you want to pause.
3. Right-click the actual destination, and select **Stop** or left-click the actual destination and then left-click the icon on the toolbar.

   Note

   If the **Stop** action does not appear on the menu or toolbar, use the **Customize** dialog to add the action.
4. Choose from these options:
   - Select **Now** to pause the actual destination.
   - Select **After current copy** to pause the actual destination after the current copy has finished printing.
   - Select **After current job** to pause the actual destination after the current job has finished printing.
5. Click **OK**.

   The actual destination stops printing, and the jobs processing on the actual destination stay in the same state until the actual destination starts again.

Resuming the actual destination

To resume the actual destination:

- In the GUI, right-click the actual destination and select **Resume**.
- Select the actual destination and left-click the icon on the toolbar.

Interrupting and restarting a job that is currently printing on an InfoPrint 4000/4100

Use this procedure:

- To interrupt the job that is currently printing and start printing it again

- To interrupt the job that is currently printing, print another job, and then start the job that you interrupted.
- To interrupt the job that is currently printing, move the job to a different printer, and start printing the job again.

This procedure does NOT stop the printer; it only stops the job that is currently printing. Other jobs continue to print on this printer.

Once the job is stopped, the next job in the queue will start to print. If you need to stop the printer (for example, to replace toner), use the procedure Stopping a printer to carry out scheduled maintenance, p. 400.

Note

- For instructions on completing any of these steps, see the InfoPrint Manager GUI help.

When you use the Interrupting a job that is currently printing, p. 397 procedure, the job fills a physical sheet, stops, and prints a message after the last page indicating the number of the last (AFP) page printed. The job shows on the GUI as **Paused**. When the job restarts, it starts on the **next** page of the job in the next logical partition. InfoPrint Manager prints a message at the end of the restarted section of the job indicating the page on which the printing started. You might want to use this procedure if you are printing a very long job, and you need to print another, more urgent job immediately.

Interrupting a job that is currently printing

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. In the GUI, find the printer where the currently printing job is, and select it.
3. Right-click the printer, and select **Interrupt Printing Job** or left-click the printer and then left-click the icon on the toolbar.

   Note

   If the **Interrupt Printing Job** action does not appear on the Printer menu or toolbar, use the **Customize** dialog to add the action.
4. Choose from these options:
   - Select **Now** to interrupt the job as soon as possible
   - Select **After current copy** to interrupt the job after the current copy has finished printing.
5. Click **OK**.

   The job that is currently printing stops printing, its icon changes to black, and the job is listed in the job window of the main InfoPrint Manager GUI window. Other jobs continue to print on this printer.

Restarting the interrupted job

To restart the interrupted print job:

- In the GUI, right-click the interrupted job and select the **Resume** action.

- Select the job and left-click the

  icon on the toolbar.

Moving backward or forward in a job that is currently printing

In InfoPrint Manager, moving from page to page in a job is called *spacing the printer*:

- *Forward spacing* is moving to a point later in the job (such as skipping from page 10 to page 15).
- *Backspacing* is moving to a point earlier in the job (such as backing up from page 40 to page 20).

Spacing a printer

Things to remember:

- **You can only space a printer that uses the PSF Destination Support System (DSS) and one of these attachment types:**
  - PSF TCP/IP
  - PSF BSD
  - PSF Other

  To see what DSS your printer uses, follow this procedure:

  1. Select the printer in the tree view of the Main window of the InfoPrint Manager GUI.
  2. See if the **Type** column appears in the details view. The **Type** column displays the DSS that the InfoPrint printer uses.
  3. If you do not see the **Type** column, use **Add/Remove menu items** to add it.
- **You can only space a printer after a job has already started to print.**
  To only print a section of a job (such as page 11 to page 20), or to start printing from a page other than the first page of the job, use the **Change Page Range to Process** task in the InfoPrint Manager GUI.
- **Spacing works better on high-speed printers than on low speed printers.**
  While spacing is supported on all printers that you print to using the PSF DSS, it is very difficult to space printers that print 40 pages per minute (ppm) or less. Spacing works adequately on printers that print between 60 and 100 ppm, and works very well with high-speed InfoPrint printers that are configured correctly.
- **The InfoPrint 4000 and InfoPrint 4100 printers require additional configuration for spacing to work efficiently.**
  See the **Setting up high speed printers for spacing** topic for instructions on configuring your printer.
- **Page numbers refer to pages of the print job, not necessarily to any page numbers in your document.**

For example, in a book, the first page of chapter one is usually "page 1." However, it might be page 15 of the print job, because of the title page, credits, and table of contents.

- **You cannot space to the back side of a sheet of paper.**
  If you are printing a duplex job, you will always print both the front and back sides of the sheet of paper that you space to.

To space a printer, follow this procedure:

1. Start the InfoPrint Manager GUI.
2. In the GUI, find the InfoPrint printer that you need to space.
3. Select the InfoPrint printer by clicking on it, then click **Printer** → **Stop**.

   Note

   If you do not see **Stop** in the menu, use**Add/Remove menu items** to add it.
4. In the **Stop *printer_name*** dialog, select **Pause** and **Now**, and click **OK**.

   Even though you selected **Now**, the printer will keep printing for a while because it has to get to a point in the job where it can stop. For InfoPrint 4000 printers that are configured correctly, you should not have to wait more than 10 seconds. For other printers, you might have to wait longer.
5. Once the printer has stopped printing, look at the last sheet that printed. Write down the number of that sheet of the job.
6. Figure out what sheet of paper in the job you want to space to. Write down the number of that sheet of the job.
7. Figure out the number of sheets of paper you need to space.
   - To **forward space** the printer:
     Subtract the number of the sheet that printed last (from step 5) from the number of the sheet that you want to skip to (from step 6), and then subtract one. Like this:

     (number from step 6) – (number from step 5) – 1 = number of sheets to skip
   - To **backspace** the printer:
     Subtract the number of the sheet that you want to go back to (from step 6) from the page that printed last (from step 5), and add one (for the current page). Like this:

     (number from step 5) – (number from step 6) + 1 = number of sheets to back up
8. Determine the number of "sides" to space the printer.
   - If you are printing a simplex job, the number of sides is the same as the number of sheet from step 7.
   - If you are printing a duplex job, multiply the number from step 7 by two to get the number of sides. (number from step 7 x 2 = number of sides)
9. In the InfoPrint Manager GUI, find the printer you want to space and select it.
10. With the printer selected, click **Printer** → **Space**.
11. In the **Space *printer_name*** dialog, select **Backspace** or **Forward space** and type the number of sides you found in step 8.
12. Click **OK**.
13. Make sure the printer is still selected and click **Printer** → **Resume** to start the printer again.

Stopping a printer to carry out scheduled maintenance

Use this procedure to stop a printer so you can carry out routine maintenance, such as updating the microcode, changing paper, or cleaning the printer. While you have the printer stopped, InfoPrint Manager will not be able to send or schedule jobs to it. This procedure works for printers that use any DSS.

1. Open the InfoPrint Manager Operations GUI or the InfoPrint Manager Administration GUI.
2. In the GUI, find the printer that you want to stop and select it.
3. Use **Printer** → **Disable** to disable the printer.

   The InfoPrint printer will stop accepting jobs, but the printer will not stop printing immediately. All the jobs that were in the **processing** state before you disabled the printer will print before it stops.
4. After the printer stops printing, you can carry out whatever maintenance you need to.
5. When you are finished and the printer is ready to receive jobs again, go back to the InfoPrint Manager GUI.
6. Find the printer in the GUI and select it.
7. Use the GUI to enable the printer again.

   The printer will be available to accept and print jobs again.

Fixing a printer problem and starting to print again

Use these procedures when the printer stops printing because of a paper jam, "out of paper" error, or for some other reason.

Note

- See the online help in the InfoPrint Manager GUI if you need more instructions on completing any of these procedures.

Try this first

"Out of paper" errors and some paper jams are easy to recover from. Most of the time, you do not have to do anything to InfoPrint Manager to fix them; InfoPrint Manager just waits until the printer is ready to print again. While it is waiting, InfoPrint Manager continues to accept jobs and schedule them to the printer.

If the printer stops printing because of a problem, first try this.

1. Check the printer.
   - If a lot of pages of the job are ruined and need to be reprinted, stop here. Do not finish this procedure. Use the Fixing more complicated problems, p. 401 procedure below instead.
   - Otherwise, load paper if the bins are empty or clear paper jams as the printer console indicates. Once everything is clear, the printer should reset itself and continue sending jobs. InfoPrint Manager should pick up where it left off.

2. Find the printer in the InfoPrint Manager GUI. Make sure that its status is **ready**. If it is not, select the printer and use the GUI to enable it.
3. If just a few pages of the print job need to be printed again, the easiest thing to do is to resubmit the job and print only those pages through **page-select** attribute. You can only print part of the job if you are using the PSF DSS and if the job will be transformed to AFP before it is printed.
4. If just a few sheets of the print job need to be printed again, this can be done by resubmitting the job and reprinting only certain sheet ranges. You can only print sheet ranges of the job if you are using the PSF DSS. The sheet range option does not apply to the ASCII or DPF IPDS datastreams.

Fixing more complicated problems

Use this procedure when the problem is more difficult to fix or if you have to reprint numerous pages of a job.

1. Open the InfoPrint Manager GUI.
2. Find the InfoPrint Manager printer (also called an actual destination) that sends jobs to that printer and select it.
3. Use the GUI to disable the printer.
   The printer is not able to accept any print jobs until you enable it again.
4. If the printer stopped printing in the middle of a job, find that job in the GUI and select it.
5. With the job selected, click **Job** → **Hold**.

   Note

   If the **Hold** item does not appear in the **Job** menu, use **Add/Remove Menu Items** to add it.
6. Click **OK**.
   The job stops printing, the icon changes to black, and the job is listed in the job window of the main InfoPrint Manager GUI window.
7. Decide what you want to do with any other jobs that are in the queue waiting for this printer. Some of the options are:
   - Leave them in the queue, so they print when you enable the printer again. If you can fix the problem quickly, this option is the easiest one. In addition, if your printers are pooled, the other printers will take care of all the jobs until this one is ready.
   - Move them to a different printer so they can be scheduled to print while you are working on this one.
   - Delete them. You should only consider this option if it is easier to resubmit the jobs than to use other methods to recover them.
8. Fix the problem at the printer.
   Once the problem is fixed, the printer might start to print again, just to clear the pages that were in the buffer before you disabled it and deleted the job. If any other jobs were in the **processing** state when you disabled the printer, they will print as well.
9. Go back to the InfoPrint Manager GUI, find the InfoPrint Manager printer again, and select it.
10. Use the GUI to enable the printer.

The printer will be available to accept jobs again. If you left jobs in the queue, the printer should start accepting and printing them.

Restarting the job that was printing when the problem occurred

You can either:

- Start the job from the beginning.
- Start the job from a point before the problem occurred (so the damaged pages get printed).

  Important

  – You can only start printing in the middle of the job if the job is transformed to AFP before it gets printed. You cannot print part of an ASCII, KGL, metacode, or LCDS job; you have to print the whole job.

  – You can start printing in the middle of the job, even if the job has more than one printable document. You can do this by specifying a sheet range (or multiple sheet ranges) to be reprinted.

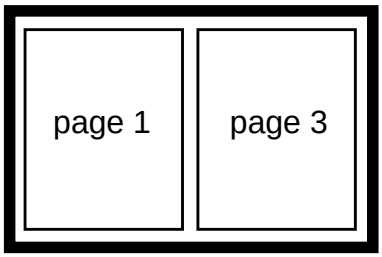
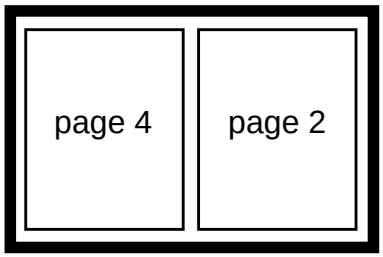
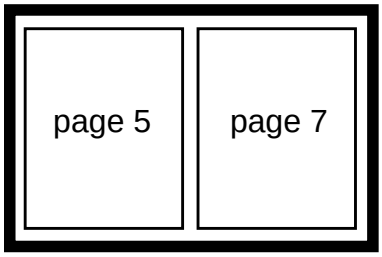
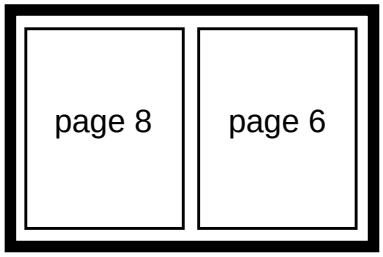
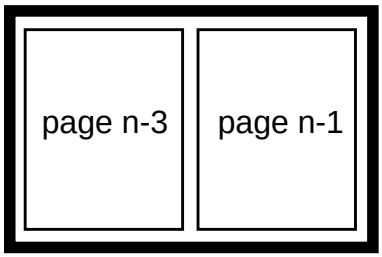
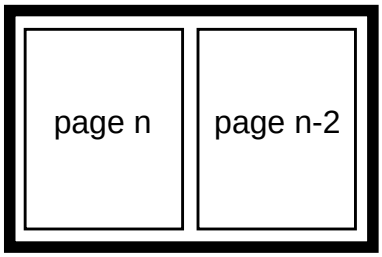
1. To print the entire job on the same printer it was printing on before, select the job in the job window and click **Job** → **Release**. The job will start printing from the first page.
2. To print the entire job on a different printer, skip to **Before you Continue** between steps 8 and 9.
3. To print only part of the job, figure out what page you want to start printing from.

Important

If you do not want to start printing from the first page of the job, be careful when you select which page to start with. Duplex and n-up jobs can be confusing because you have to start printing with the **first** page on the **front** side of the sheet of paper. If you do not choose the correct start page, the job will print, but the sequence will probably be wrong. The easiest way to avoid this situation is to choose a sheet range specifying the start and the end of the sheet that you want to print.

For example, if you are printing a 2-up duplex job, each sheet of paper has four pages printed on it, like this:

2-up duplex print job: front and back views

	Front of sheet	Back of sheet
Sheet 1		
Sheet 2		
Last sheet		

Since you have to start printing this job at the first page on the front side, you can choose page one or page five (or any other page that would fall in that position). If you choose a different page, the sequence of pages will be wrong, and the job will not print correctly.

However, there is a way to avoid this situation. You can choose a sheet range specifying the first and the last sheet that you want to print.

To define the sheet range:

1) Go to **Job** → **Properties** → **Show more**.
2) Click the **Job Other** tab.
3) Enter a number range in the fields provided under the **Sheet range** section.

4) Click the **Add** button to insert new ranges in the values list field on the right.

Note

You can specify up to 10 sheet-ranges to be printed. If you specify more than 10 sheet-ranges, no sheets print and InfoPrint Manager issues an error.

4. Decide what page you want to stop printing on (if you do not want to print all the way to the end of the job).

   You do not have to be as precise on the last page to print; the last page can be in any location on the sheet of paper.

5. Select the job that you want to print.
6. With the job selected, click **Job** → **Change Page Range to Process**.

   If you do not see the **Change Page Range to Process** item in the **Job** menu, use **Add/Remove Menu Items** to add it.

7. In the **Change Page Range to Process** dialog, select **Range of Pages** and type the start page and end page you have chosen.

   Note

   If you submitted this job using Distributed Print Facility (DPF), InfoPrint Manager ignores the end page that you specify. DPF jobs always print from the start page that you specify to the end of the job.

8. Click **OK**.

   Before you continue:

   You should only move a job to an InfoPrint printer that uses the same destination support system (DSS) as the one the job was originally submitted to. If you move the job to a printer that uses a different DSS and the new printer does not support all the attributes of the job, the job fails to print. To check which DSS your InfoPrint printer uses:

   1) Select the printer in the tree view of the Main window of the InfoPrint Manager GUI.
   2) See if the **Type** column appears in the details view. The **Type** column displays the DSS that the InfoPrint printer uses.
   3) If you do not see the **Type** column, use **Add/Remove menu items** to add it.

   Continue with step 9.

9. Select the job again and click **Job** → **Move job**.
10. In the **Move Jobs** dialog, select the printer that you want to print the job on. You can select the same printer it was printing on before or a different one.

    When you release the job in step 12, if the job is accepted, it goes back into the queue and wait to be printed. Depending on how many jobs are ahead of it in the queue, it might take a while for the job to print. If you want the job to print right away, you can either move it to the top of the queue (so it prints next) or you can change its priority (so it prints sooner, but not necessarily next).

11. To move the job to the top of the queue, select the job in the InfoPrint Manager GUI and use the **Make Job First** task. To change the priority of the job, use the **Change Priority** task.
12. Select the job in the job window and click **Job** → **Release**.

Note

If the task you want to use does not appear in the menu, use **Add/Remove menu items** to add it.

Using the correct halftones for the installed toner version

Multiple versions of toner are available for the InfoPrint 4000 and InfoPrint 4100 printers. For some printer models, InfoPrint Manager ships different halftone curves for different toners, and provides scripts to replace or install the correct ones. Make sure you have the correct halftone installed for your toner to achieve the best print quality.

The toner characteristics can require changes in the halftone resources. When the toner in the printer is replaced with a different version of toner, you must change the version of the printer toner that you want to use and set the **toner-version** actual destination attribute.

5. Reference Information

- **IPDS print operator commands**
- **IPDS error recovery**

IPDS print operator commands

InfoPrint Manager offers a comprehensive set of operator commands that can be used to manage printing on IPDS printers. These commands can be used to help manage printing priorities and also help by providing the operator with functions that are useful when dealing with problems that can occur in the printing system. This table describes several of the most commonly used operator tasks, and their corresponding commands. For more information, see the *RICOH InfoPrint Manager: Reference*.

	Task	Description	InfoPrint Manager Command Syntax
1	Disable	Stop a printer from receiving new jobs.	***pddisable -cd*** *dest* where *dest* specifies either a logical or an actual destination (printer) object.
2	Enable	Allow a printer to receive new jobs.	***pdenable -cd*** *dest* where *dest* specifies either a logical or an actual destination (printer) object.
3	Query active job	Query which job is actively printing on a printer Note The query response is based on the last set of station counters obtained from the printer, which is printing pages continuously. To get an absolutely accurate query you must: • Pause the printer (7) • Issue the query • Resume the printer (11).	***pdls -f "destination-name-requested==ldest && current-job-state==printing"*** *server:* where *server* specifies the InfoPrintLinux server to which the job was submitted, and *dest*specifies the actual destination (physical printer).
4	Query page number	Query the page number of the currently printing page. See **Note** after (3).	***pdls -cj -r current-page-printing*** *jobid* where *jobid* specifies the **job-identifier** attribute. we recommend using the global ID, which follows the convention *server*:0123456789 where *server* specifies the name of the InfoPrintLinux server to which this job was submitted and 0123456789 indicates a 10-digit numeric value assigned by the InfoPrint spooler. You can determine the **job-identifier** attribute value by specifying: ***pdls -cj*** *server:*
5	Query job details	Query the number of copies, sheets and pages stacked for a job. See **Note** after (3).	***pdls -cj -r "job-copies-completed media-sheets-completed pages-completed"*** *jobid* where *jobid* specifies the **job-identifier** attribute.

	Task	Description	InfoPrint Manager Command Syntax
6	Clean	Remove all jobs from a printer.	***pdclean*** *ldest* where *ldest* specifies the logical destination (logical printer).
7	Pause printer	Pause a printer immediately.	***pdpause -cd*** *dest* where *dest* specifies the actual destination (physical printer).
8	Pause printer	Pause a printer after the current copy completes.	***pdpause -cd -w after-current-copy*** *dest* where *dest* specifies the actual destination (physical printer).
9	Pause printer	Pause a printer after the current job completes.	***pdpause -cd -w after-current-job*** *dest* where *dest* specifies the actual destination (physical printer).
10	Resume printer	Resume a printer that had been paused.	***pdresume -cd*** *dest* where *dest* specifies the actual destination (physical printer).
11	Cancel job	Cancel a specific job.	***pdrm*** *jobid* where *jobid* specifies the **job-identifier** attribute.
12	Pause job	Pause a specific job after the current copy completes.	***pdpause -cj -w after-current-copy*** *jobid* where *jobid* specifies the **job-identifier** attribute.
13	Pause job	Pause a specific job immediately.	***pdpause -cj*** *jobid* where *jobid* specifies the **job-identifier** attribute.
14	Pause job	Pause the job that is currently printing immediately.	***pdpause -cd -j*** *dest* where *dest* specifies the actual destination (physical printer).
15	Pause job	Pause the job that is currently printing after the current copy.	***pdpause -cd -j -w after-current-copy*** *dest* where *dest* specifies the actual destination (physical printer).
16	Resume job	Resume a job that was paused.	***pdresume -cj*** *jobid* where *jobid* specifies the **job-identifier** attribute.
17	Back space	Back space (reprint a specific number of sides) the job that is printing.	***pdspace -cd -b sidecount*** *dest* where *dest* specifies the actual destination (physical printer).
18	Reprint	Reprint the job that is printing from the beginning.	***pdspace -cd -b -1*** *dest* where *dest* specifies the actual destination (physical printer).
19	Forward space	Forward space (skip a specific number of sides) the job that is printing.	***pdspace -cd -f sidecoun***t *dest* where *dest* specifies the actual destination (physical printer).

If one of these commands is received and it affects a job that has been sent or is in the process of being sent to the printer, InfoPrint Manager immediately stops sending data to the printer and sends the printer a command to stop printing (this applies to commands 7, 9, 10, 12-15).

For printers that support the SNMP Remote Operator interface, like the InfoPrint 4000, the printer will stop in a few seconds. For printers without this support, some pages will likely print before the printer stops.

The number of pages that print before the printer stops will vary depending on the number, size, and complexity of jobs that are in process as well as the speed and type of printer involved. Once the printer has stopped, InfoPrint Manager does the steps necessary to execute the requested operator command.

IPDS error recovery

When driving an IPDS printer, InfoPrint Manager sends page and resource data to the printer. The printer reports status and errors back to InfoPrint Manager. The printer status information is in the form of station counters representing progress of pages sent to the device through the printer microcode and the physical printing mechanism. Station counters report the number of pages and the number of copies of pages proceeding through each point in the printer.

InfoPrint Manager requests printer status information (an acknowledgement request) after sending each job. The acknowledgement response contains the latest values of all the printer station counters. Once all pending jobs have been processed, InfoPrint Manager polls the printer periodically until all the pages of each job have been stacked.

The actual destination object **ack-interval** attribute is configurable and defines how often InfoPrint Manager requests updated station counters from the printer while processing a particular job. The value for this attribute is specified in terms of pages and can be from 1 to 9999; the default value is 100 pages. For a 360 page job, an **ack-interval** setting of 100 pages means that InfoPrint Manager will send an acknowledgement request to the printer after sending pages 100, 200, 300, and 360.

The printer responds to an acknowledgement request and returns station counters when all of the pages up to that point have been processed. This does not necessarily mean that all of those pages have printed or are stacked. The acknowledgement response from the printer after the 100th page will show that 100 pages have been processed (received). The printer might report, however, that only 10 pages have stacked.

The **ack-interval** value affects printing performance and the currency of job status information. A small **ack-interval** value causes increased communication and processing and can cause a reduction in printer throughput. A small **ack-interval** value will, however, increase the frequency that job attributes such as **current-page-printing**, **job-copies-completed**, **media-sheets-completed**, and **pages-completed** are updated. A small **ack-interval** value will also enable the InfoPrint Manager software to have more recent station counters than it would with a large **ack-interval** value.

Errors reported from the printer can be grouped into these categories:

- IPDS error recovery: Data stream errors, p. 409
- IPDS error recovery: Insufficient memory in the printer, p. 412
- IPDS error recovery: Intervention required conditions, p. 412

IPDS error recovery: Data stream errors

For data stream errors, the job might continue printing or the job might be terminated, depending on the severity of the error. There are hundreds of specific data stream errors. If one data stream error is

defined for the error, InfoPrint Manager directs the printer to continue processing the page and take an alternate exception action.

When a data stream error occurs, messages are produced describing the problem and the recovery actions performed. The messages report details about the error and include the page number where the error occurred. After InfoPrint Manager receives a data stream error from the printer and produces a message, it uses station counters returned from the printer to continue printing without skipping or reprinting any pages.

Reporting of some data stream errors (for example, using undefined code points in a font or printing outside of the valid printable area) can be controlled with job attributes. This allows the job submitter to decide whether error messages or printer error marks should be generated when these types of errors are encountered:

- Data stream error example 1 (job continues printing):, p. 410
- Data stream error example 2 (job is terminated, no pages print):, p. 411
- Data stream error example 3 (job is terminated; some pages print):, p. 411

Interrupt Message Pages

If Interrupt Message Pages are activated, InfoPrint Manager will insert an IMP to mark the location of the error. See Using Interrupt Message Pages, p. 91 for more information.

Data stream error example 1 (job continues printing):

If page 3 of a 10–page document tries to print outside the valid printable area (an error) and the submitter specified (data-fidelity-problem-reported=all), this occurs:

1. Pages 1-2 and 4-10 in the document print normally.
2. For page 3, the printer prints only the data that is inside the valid printable area. In addition, a printer error marker (PEM) is printed on the page where the print data extends outside the valid printable area.
3. After all 10 pages have printed, a set of messages is printed:

0420-094:	The following messages were generated for file /info/afp/baddata with Job ID 123. This file printed on the InfoPrint destination ip60, which is a 3160 destination.
0420-484:	The destination reported that an attempt was made to print outside of the valid printable area.
0420-249:	InfoPrint received IPDS exception X'08C1..00', action code X'01' from the destination.
0420-098:	The error occurred while printing page 3 of copy 1 of this print job.
0420-254:	The destination reported an error that caused InfoPrint to stop processing the page.

Data stream error example 2 (job is terminated, no pages print):

If a document is submitted and it requests a specific form definition which does not exist, this occurs:

1. The document does not print. However, auxiliary sheets such as the start sheet (header) and end sheet (trailer) page, if requested, do print.
2. A message page will print with these messages:

0420-094:	The following messages were generated for file /info/afp/testfdef with Job ID 456. This file printed on the InfoPrint destination ip60, which is a 3160 destination.
0420-128:	ERROR: InfoPrint cannot find or cannot access the FORMDEF resource named F1BAD.
0420-060:	The error occurred with the object type page with token name PAGE0001.
0420-060:	The error occurred with the object type document with token name ASCII.
0420-098:	The error occurred while printing page 1 of copy 1 of this print job.

Data stream error example 3 (job is terminated; some pages print):

If page 5 of a 10–page document requests a font (C0D0GT15) which cannot be located, this occurs:

1. Pages one through four print; page five does not print. Printing of the document ends; pages 6 through 10 also do not print.
2. A message page will print with these messages:

0420-094:	The following messages were generated for file /info/afp/long.list3820 with Job ID 167. This file printed on the InfoPrint destination psf4317, which is a 4317 destination.
0423-284:	InfoPrint mapped the FGID, GCSGID, width, vertical size combination: 230, 0, 96, 160 to the FCS name C0D0GT15. The FGID.GRD mapping file specified that this mapping be performed.
0423-291:	Map Coded Font (MCF) structured field could not be processed because InfoPrint could not locate the font character set C0D0GT15. Attempted resolution/type(s): 300-pel Font Fidelity for this job was CONTINUE. Destination supported resolution/type(s): 300-pel.
0420-060:	The error occurred with the object type Active Environment Group with token name 1.
0420-060:	The error occurred with the object type page with token name 1.
0420-060:	The error occurred with the object type document with token name LONG.
0420-098:	The error occurred while printing page 1 of copy 1 of this print job.

0423-302:	The font reference in the MCF1 structured field, repeating group number 1, with sequence number 1, at file offset 118, could not be processed successfully. Additional error and diagnostic messages follow.
0420-729:	The error occurred in file /info/afp/long.list3820. The structured field in error was structured field EAG with code X'D3A9C9' and sequence number 1 located at file offset 250.

IPDS error recovery: Insufficient memory in the printer

In some cases, a printer might not have sufficient storage to process the data or resources required for a page. When this occurs, InfoPrint Manager responds by deleting resources in the printer which are not required for the page and then tries to print the page again. Frequently, this recovery will allow the page to print successfully. If the printer still reports that there is insufficient storage to process the page, the job is terminated and a message is produced indicating that the page and its associated resources are too complex.

5

Insufficient memory in the printer exmple:

A 30 page document is submitted. Each page of the document references several complex page segments.

1. InfoPrint Manager sends pages for the document to the printer.
    - Before each page is sent, InfoPrint Manager downloads page segments that are required by the page.
2. The printer reports an insufficient memory error to InfoPrint Manager (for page 23).
    - InfoPrint Manager deletes all the page segments (and any other resources) previously downloaded to the printer. All of the printer memory available for resources should now be free. InfoPrint Manager then tries to load only the page segments required for page 23. Because unnecessary page segments have been removed from printer memory, all of the page segments for page 23 can be successfully downloaded into the printer. No messages are reported to the server and no message pages are printed.
3. All 30 pages of the job print successfully.

IPDS error recovery: Intervention required conditions

A printer might experience intervention required conditions that are not related to the data stream or complexity of the data. These conditions cause printing to stop. Some intervention required conditions are handled directly by some printers and are not reported back to InfoPrint Manager. In these cases, InfoPrint Manager continues to send data to the printer.

Example:

1. As soon as the paper jam occurs, printing stops and the destination (printer) object changes to red.
   - A message is sent to the operator:

```
0420-466: The destination reported a paper jam; clear the jam.
```

2. The actual destination (printer) object's destination-state changes to "needs key operator."
   - InfoPrint Manager waits until the paper jam has been cleared.
3. The operator then clears the paper jam, according to the printer jam recovery procedure. Some damaged pages are usually discarded as part of this procedure.
4. Once the printer jam is cleared, the destination (printer) object is no longer red. Using station counters from the printer, InfoPrint Manager repositions to the correct page and continues printing the document. If Interrupt Message Pages are activated, InfoPrint Manager will insert an IMP at this point to mark the location of the error. See Using Interrupt Message Pages, p. 91 for more information.
5. At the end of the document, a message page is printed which contains this:
   - 0420-094: The following messages were generated for file /info/paperjam with Job ID 787. This file printed on the InfoPrint destination ip32, which is a 4332 destination.
   - 0420-467: A paper jam occurred in the destination while printing this job. Inspect the output to determine if it is complete and that all printed pages are acceptable.

IPDS error recovery: Unrecoverable problems

Unrecoverable problems cause the InfoPrint Manager software or printer software to stop functioning or prevent them from communicating with each other. Examples of unrecoverable errors include: server system crash, software core dump, printer abort, power failure, or severe network problems.

These errors prevent the InfoPrint Manager software from obtaining current station counter status information from the printer. In these cases, InfoPrint Manager cannot determine how many pages and which jobs printed since the last set of valid station counters were received. InfoPrint Manager considers a job complete only when all pages are reported by the printer as stacked. In most cases, some pages for one or more jobs will print between the time InfoPrint Manager last received a printer acknowledgment and the time the unrecoverable error occurs.

Those jobs, because they are considered incomplete, will normally start printing from the beginning when the problem is resolved and both the InfoPrint Manager software and the printer are operating and communicating again.

However, if the submitter or operator does not want the job to start printing from the beginning, there is a way to modify where printing starts for the documents in these jobs. The document object **page-select** attribute can be modified before the job is processed again. This attribute allows the submitter to specify a page range (first to last) for each document in a job. This way, the submitter or operator can control which pages are printed after the problem is resolved.

Another way to modify where the printing starts is to set the **sheet-range** job attribute before the job is processed again. This attribute allows the submitter to specify a sheet range (first to last) for the whole

job. This way, the submitter or operator can control which sheets are printed after the problem is resolved.

Depending on the type of error and the network configuration, some errors might not be reported to InfoPrint Manager immediately. In these cases, an operator can issue a **pdshutdown** command to the actual destination object. This command works without any attempts to communicate with the printer. InfoPrint Manager will checkpoint the job that was partially stacked at the printer and use the station counters from the last valid acknowledgment to determine which page of which job to checkpoint. A small **ack-interval** value will make it possible to checkpoint using a more recent set of station counters than would be available with a large **ack-interval** value. The job can be resumed later from that checkpoint.

Some unrecoverable errors are reported immediately to the InfoPrint Manager software and the actual destination object (printer) is disabled. When this occurs, the **pdshutdown** command cannot be issued and therefore no checkpoint can be taken. InfoPrint Manager does not perform automatic checkpointing.

Unrecoverable problem example:

1. The destination (printer) object becomes disabled and the icon changes to **black**.
2. The job object changes to the **pending** state.
3. Later, the printer power is restored.
4. The operator must enable the destination (printer) object.
5. Jobs start printing on the printer in the order they have been scheduled.

All 300 pages of the job that were printing during the power failure print again.

Accessibility

Ricoh strives to provide products with usable access for everyone, regardless of age or ability. For more information about the commitment that we have made to accessibility, see: https://www.ricoh-usa.com/en/about-us/corporate-responsibility-and-environmental-sustainability/accessibility.

Accessibility features

Accessibility features help users who have disabilities, such as restricted mobility or limited vision, use information technology products successfully.

The major accessibility features in this product let you:

- Use screen readers, screen magnifiers, and other assistive technologies.
- Use a keyboard instead of a mouse.
- Change attributes such as volume, color, contrast, and font size.

In addition, the information center and the publications for the product are in an accessible format.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

Glossary

This glossary defines technical terms and abbreviations used in InfoPrint Manager.

Special Characters

.Guidefaults file

A file created by InfoPrint Manager in the home directory of a person using the InfoPrint Manager GUI. InfoPrint Manager uses this file to save and reference information about the servers you are monitoring and the windows you are working in.

A

Access Control List (ACL)

In computer security, a collection of all access rights for one object.

ACL

Access Control List.

acl editor

A command line interface that lets you view, add, update, and delete access authorization records in an ACL.

action

In the InfoPrint Manager GUI, an icon that represents an operation that you can perform on an InfoPrint Manager object. You drag and drop the action onto an object to initiate the operation for that object. For example, using the `Enable` action to enable a disabled actual destination.

actual destination

In InfoPrint Manager, an object that represents the output device that performs the printing or transmission function. See *email destination*; see also *physical printer*, *printer device*; contrast with *logical destination*.

additive color system

A system that reproduces an image by mixing (adding) appropriate quantities of red, green, and blue light (the additive primary colors) to create all other colors of light, either direct or transmitted. When the additive primaries are superimposed on one another, they create white light. Contrast with *subtractive color system*.

administrator

In InfoPrint Manager, the person who creates and manages one or more components of a printing system, such as servers and actual destinations. By default, InfoPrint Manager gives administrators authorization to perform some InfoPrint Manager operations and to access certain information that is not available to the operators or job submitters.

Adobe Acrobat

An Adobe software program that provides instant access to documents in their original format, independent of computer platform. With the Adobe Acrobat Reader, you can view, navigate, print, and present any Portable Document Format (.pdf) file.

Adobe PostScript Raster to Image Transform (RIP)

See *raster image processor (RIP)*.

ADSM/6000

Advanced Distributed Storage Manager.

Advanced Distributed Storage Manager (ADSM/6000)

A program that provides storage management for archived files.

Advanced Function Common Control Unit (AFCCU)

A RISC-based control unit with code common to all printers that use the AFCCU.

Advanced Function Presentation (AFP)

A set of licensed programs, together with user applications, that use the all-points-addressable concept to print data on a wide variety of printers or to display data on a wide variety of display devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information.

Advanced Function Presentation data stream (AFP data stream)

The printer data stream used for printing AFP data. The AFP data stream includes composed text, page segments, electronic overlays, form definitions, and fonts downloaded from the operating system to the printer.

A presentation data stream that is processed in AFP environments. The AFP data stream includes MO:DCA-P-based data streams.

Advanced Interactive Executive (AIX)

An operating system used with pSeries computers. The AIX operating system is IBM's implementation of the UNIX operating system.

AFCCU

Advanced Function Common Control Unit.

AFP

Advanced Function Presentation.

AFP data stream

Advanced Function Presentation data stream.

ainbe

The printer input/output backend program used by the PSF DSS to send jobs to printer devices. Depending on the incoming data stream of the job, the PSF DSS might have transformed the data stream of the job before the backend program sending it to the printer device.

AIX

Advanced Interactive Executive.

AIX-attached printer

A printer device, attached to a pSeries computer through the network or through a serial or parallel port that receives jobs from an AIX print queue.

In InfoPrint Manager, a PSF physical printer that represents an AIX-attached printer device. See also *direct-attached printer*, *TCP/IP-attached printer*, *upload-TCP/IP-attached printer*.

AIX destination support system

In InfoPrint Manager, the destination support system that communicates with the standard AIX print backend (`piobe`), or with certain variations of the `rembak` print backend, to print jobs.

AIX physical printer

In InfoPrint Manager, the object representing a printer device that uses the AIX destination support system.

aliasing

In a digitized image, the creation of diagonal lines by combining short horizontal and vertical line segments that approximate the path of the desired line.

all-points-addressability (APA)

The capability to address, reference, and position text, overlays, and images at any defined point of the printable area of the paper or display medium.

alphameric

Synonym for *alphanumeric*.

alphanumeric

Pertaining to a character set containing letters, digits, and other symbols such as punctuation marks. Synonymous with *alphameric*.

AMPV

Average monthly print volume.

analog

Pertaining to a continuous variable sampling of information between two points that achieves an even, smooth transition of photographic material.

analog color proof

An off-press color proof made from separation films.

anti-aliasing

The rendering of hard-edged objects so that they blend smoothly into the background.

Removing the jagged "stairstep" quality in diagonal lines produced on a computer screen by aliasing. This removal is effected by creating less densely shaded fields parallel to the diagonal line.

APA

All-points-addressability.

API

Application Program Interface.

Application Program Interface (API)

The call interface between a client program and the procedures that implement the printing system, as defined by the specification. Clients use the API to access servers. (P)

architecture

The set of rules and conventions that govern the creation and control of data types such as text, image, graphics, font, color, audio, bar code, and multimedia.

archiving

The transfer of digital information from an online system onto various media for offline storage. The original copy is deleted from the online system. See also *retrieval*.

array inkjet

An ordered collection of multiple print heads used in an inkjet printer.

ASCII

American National Standard Code for Information Exchange. The standard code, using a coded character set consisting of 7–bit coded characters (8 bits including a parity check). The ASCII code is used for information exchange among data processing systems, data communications systems, and associated equipment. The ASCII character set includes control characters and graphic characters.

asynchronous

Pertaining to two or more processes that do not depend upon the occurrence of specific events, such as common timing signals. (T)

In InfoPrint Manager, occurring without a regular or predictable time relationship to a specified action. Contrast with *synchronous*.

attachment type

Defines the method by which a printer device is attached to the AIX system from which it receives data. See *AIX-attached printer*, *direct-attached printer*, *TCP/IP-attached printer*, and *upload-TCP/IP-attached printer*.

attribute

A defined characteristic of an object, such as the number of copies required of a job, or the document formats accepted by an actual destination.

attribute name

A string that identifies an attribute. Typically, in InfoPrint Manager attribute names consist of two or more words separated by hyphens.

attribute value

The element of an attribute that specifies the characteristics relating to the attribute.

authorization

In computer security, verification of the eligibility of a user to access an object.

authorized user

A person with the appropriate permissions to access an object or to issue a command.

automatic recovery

A function of printer logic that permits the printer to reprint a job if the job is misprinted or damaged or if a component has failed.

auxiliary sheet

In InfoPrint Manager, an object that represents a particular sheet of paper, blank or otherwise, that can precede a job, separate documents in a job, or follow a job.

auxiliary-sheet selections

A particular combination of start, separator, and end sheets that print with jobs on a particular printer device.

B

backend

In AIX or Linux, the program that is invoked by the `qdaemon` process (AIX) or CUPS printing system (Linux) to run a print or batch job. Synonymous with *backend program*.

backend program

Synonym for *backend*.

backlog

In InfoPrint Manager, the amount of time calculated by InfoPrint Manager that is required to print all of the jobs currently in a queue.

backspace

In InfoPrint Manager, the action used to back up and reprint pages in a job.

bar code

A code representing characters by sets of parallel bars of varying thickness and separation that are read optically by transverse scanning.

An array of parallel rectangular bars and spaces that together represent data elements or characters in a particular symbology. The bars and spaces are arranged in a predetermined pattern following unambiguous rules defined by the symbology.

BCOCA

Bar Code Object Content Architecture.

Bezier curves

A method of defining curves using anchor points, control handles, and tangent lines. The PostScript path passes through anchor points. Its approach direction is along the tangent lines that are controlled by the control handles. Many personal computer programs use this drawing method. Type 1 PostScript fonts are defined using Bezier curves.

binding

The cover and materials that hold a book together. See *edition binding*, *perfect binding*, *spiral binding*, *wire-o binding*.

The process of applying the binding to a book.

In InfoPrint Manager, assigning a job to an actual destination. See *early binding*, *late binding*.

bitmapped

Pertaining to an image formed by a rectangular grid of pixels. Each pixel is assigned a value to denote its color. One-bit images are black and white; 8–bit images have 256 colors (or grays); 24–bit images have full color. CMYK images have 32–bits per pixel to encode 256 levels in each of four channels. Bitmapped images are also called raster images.

bleed

An extra amount of printed image that extends beyond the trim edge of the sheet. Bleed ensures that no white gap appears at the edge of the sheet.

boot

To prepare a computer for operating by loading an operating system.

BSD

Berkeley Software Distribution.

BSD destination support system

In InfoPrint Manager, the destination support system that generates a print command string that the shell runs to print jobs.

BSD physical printer

In InfoPrint Manager, the object representing a printer device that uses the BSD destination support system.

burn

In platemaking, an exposure. Burn is named because of the extremely bright ultraviolet lamps used to transfer images from film to plate.

In photography, to lengthen the final exposure time to increase the final density of print.

C

CDE

Common Desktop Environment.

child

See *parent/child relationship*.

choke

In color printing, an area whose dimensions have been reduced to avoid overlapping another color. Contrast with *spread*.

class

Synonym for *object class*.

clean

In InfoPrint Manager, an action used to remove all the jobs from specified servers, actual destinations, or queues, or to remove all the jobs submitted to specified logical destinations.

CLI

Command Line Interface.

client

In InfoPrint Manager, the component of the print system that assembles a print request and submits the request to a server. The client generates the local ID numbers for each job, submits the jobs to the servers, and keeps track of where each user's jobs have been submitted.

CMR

Color Management Resource.

CMY

Cyan, magenta, yellow.

CMYK

Cyan, magenta, yellow, black.

collator

In some printers a special-purpose hard drive disk array used to store RIPped files and later transfer them to the LED print heads for imaging.

color balance

The relative levels of cyan, magenta, and yellow channels in an image to produce accurate color rendition of an original image.

color correction

The adjustment of the color balance in an image to produce accurate color rendition of an original image. Color correction is used for two- or three-spot-color jobs.

color key

A color proof similar to Cromalin, but typically not laminated and not necessarily accurate in color. Color keys are used to verify that breaks or color assignments and traps have been made accurately.

Color management resource

Color management resource (CMR) is an architected resource that is used to carry all of the color management information required to render a print file, document, group of pages or sheets, page, or data object with color fidelity.

Command Line Interface (CLI)

A type of user interface where commands are specified on the command line. Contrast with *Graphical User Interface*.

Common Desktop Environment (CDE)

A graphical user interface running on a UNIX operating system.

complex attribute

In InfoPrint Manager, an attribute that can have multiple values with each value containing multiple components.

constant data

Data that does not change between copies of a document and that is merged with variable data to produce personalized documents. For example, a form letter (constant data) can be merged with a customer's name and address (variable data).

contextual help

A type of online help that provides specific information about each selectable object, menu choice, notebook tab, field, control, and pushbutton in a window.

control strip

A strip of information that can be attached to a print job and used to measure registration and density.

Cromalin

The Dupont color proofing system. Four layers (CMYK) of light-sensitive material are exposed through final halftone negative films, then laminated together on an opaque backing to produce a color- and trap-accurate final proof showing actual halftone dot structure. Cromalin is sometimes called an analog proof.

crop

To remove unwanted areas of an image, usually indicated by crop marks on the original.

CTS

Cutter-trimmer-stacker.

CUPS

Common Unix Printing System is a modular printing system for Unix-like computer operating systems which allows a computer to act as a print server. A computer running CUPS is a host that can accept print jobs from client computers, process them, and send them to the appropriate printer.

CUPS destination support system

In InfoPrint Manager, the destination support system that communicates with the standard Linux print system (CUPS) and that uses InfoPrint Manager **pioinfo** backend, InfoPrint Manager **piorpdm** backend for Ricoh printers, or other print backends to print jobs.

cutter-trimmer-stacker (CTS)

A post-processing device attached to a printer device and used for paper handling.

cyan, magenta, yellow (CMY)

The subtractive primary colors.

cyan, magenta, yellow, black (CMYK)

The four basic colors used in offset printing, as opposed to the three basic colors of light: red, green, and blue. Magenta, yellow, and cyan are the subtractive primaries. Black is added to provide better density and to produce a true black.

D

daemon

A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their tasks; others operate periodically. Daemon processes generally provide a service that must be available at all times to more than one task or user, such as sending data to a printer.

data stream

All information (data and control commands) sent over a data link.

A continuous stream of data that has a defined format.

In InfoPrint Manager, pertaining to the incoming format of the job and the output format required by the printer device. InfoPrint Manager transforms the incoming format into the output format, as required. See also *document format*.

deadline

In InfoPrint Manager, a queuing technique where the next job assigned to the first position in the queue is the one whose deadline is closest. Contrast with *FIFO (first-in-first-out)*, *job-priority*, *longest-job-first*, and *shortest-job-first*.

default document

In InfoPrint Manager, an object that represents default attribute values for a document in a job. Synonymous with *initial value document*.

default job

In InfoPrint Manager, an object that represents default attribute values for a job. Synonymous with *initial value job*.

default object

In InfoPrint Manager, a general term that refers either to a *default document* or a *default job*. Synonymous with *initial value object*.

default value

A value stored in the system that is used when no other value is specified.

delete

In InfoPrint Manager, an action used to delete default objects, jobs, logical destinations, actual destinations, or queues from the server.

desktop publishing

Performing print-related activities on a personal computer, including but not limited to layout, design, photo manipulation, creating fonts, editing text, color separation, scanning, and animation.

destination

See *actual destination*, *logical destination*.

destination support system (DSS)

In InfoPrint Manager, the programs that an actual destination uses to communicate with the output device. Synonymous with *device support system*. See *AIX destination support system*, *BSD destination support system*, *email destination support system*, and *PSF destination support system*.

device

An individual piece of equipment (hardware) that attaches to a computer, such as a printer device.

device address

The identification of an input or output device by its channel and unit number.

device driver

A file that communicates with a specific device such as a printer, disk drive, or display. An application that sends output to a device controls the actions of the device through the device driver. See *printer driver*.

device support system (DSS)

Synonym for *destination support system*.

DFE

Digital Front End

DFE destination support system

In InfoPrint Manager, the destination support system that communicates with a DFE print server driving a Ricoh printer.

DFE printer

In InfoPrint Manager, the object representing a DFE print server driving a Ricoh printer.

digital

Pertaining to data represented by digits, sometimes with special characters and the space character.

digital color proof

An off-press color proof made from a color printer attached to a computer.

digital printing

A filmless and plateless printing process in which digital data for each page is transferred directly to a light-sensitive drum and then to paper for a final image.

direct-attached printer

A printer device, attached to an pSeries computer through the network or through a serial or parallel port.

In InfoPrint Manager, a PSF physical printer that represents a direct-attached printer device. See also *AIX-attached printer*, *TCP/IP-attached printer*, and *upload-TCP/IP-attached printer*.

disable

In InfoPrint Manager, an action used to stop the acceptance of jobs on destinations, queues, or servers, or to stop writing information to logs.

distributed print system

A computer system with the ability to interchange print data and controls among different computing environments with the intent of printing the data on a different system from the one where the print request was generated. For example, in host-to-LAN distributed printing, data that is located on the host is printed on printers attached to a local area network.

dithering

A technique of filling the gap between two pixels with another pixel having an average value of the two to minimize the difference or to add detail to smooth the result.

document

In InfoPrint Manager, an object representing a grouping of data in a job. A job can contain one or more documents. The documents in a job can differ from each other in some ways. For example, they can contain different data and can have different document formats. A document in a job can contain printable data or a resource that is not printable by itself. See *file-reference document*, *printable document*, and *resource document*.

document element

A portion of a document at least a single page in size.

document format

In InfoPrint Manager, a document format describes the type of the data and control characters in the document, such as line data or PostScript. The format of the data determines which printer devices are capable of printing the document and whether InfoPrint Manager must transform the format.

document identifier

A string that identifies a document in a job. It consists of a job ID followed by a period (.) and a document sequence number. For example, `12.2`. Document sequence numbers are integers starting at `1`.

Document Printing Application (DPA)

An ISO/IEC 10175 standard that addresses those aspects of document processing that enable users in a distributed open systems environment to send electronic documents to shared, possibly geographically-dispersed printers. InfoPrint Manager supports the DPA standard.

document transfer method

In InfoPrint Manager, the transfer method describes how documents are transferred to, or acquired by, servers. See *pipe-pull* and *with-request*.

document type

In InfoPrint Manager, the document type describes the kind of data in the document. A *printable document* can only contain printable data. A *resource document* can only contain data such as fonts or form definitions that are not printable. A *file reference document* can only contain names of files entered on separate lines.

dot

The individual elements of a halftone.

dot gain

An increase in the size of a halftone dot during printing, caused by ink spreading. Generally, this value is known precisely, and the scanning and filming production process is calibrated to compensate for it. The Cromalin proofing system simulates this effect.

dots per inch (dpi)

A measure of data density per unit distance. Typical values for desktop publishing range from 200 to 300 dpi.

DPA

Document Printing Application.

DPF

Distributed Print Facility.

dpi

Dots per inch.

drag and drop

In graphical user interfaces, a procedure by which you perform actions and tasks. Using the mouse, you drag (move) an action or object icon to a new position where you want the action or task to occur.

DSS

Destination support system.

dummy

A rough paste-up or hand-drawn representation of the anticipated finished product. A dummy is used for basic design and pagination.

duplex printing

Printing on both sides of the paper. Contrast with *simplex printing (1)*.

Printing on both sides of the paper, placing output images on the media in a head-to-head format, so that the top of one image is at the same edge as the top of the next image. Contrast with *tumble duplex printing*; see also *simplex printing (2)*.

E

early binding

In InfoPrint Manager, assigning a job to an actual destination as soon as it is accepted. Early binding permits InfoPrint Manager to estimate the time when the job will be completed. Contrast with *late binding*.

edition binding

A type of book binding in which printed sheets are folded into 16- or 32-page signatures. Four-page endleaves are pasted on the outside of the first and last signature. The signatures are then collated by machine and sewn together using special sewing machines. Contrast with *perfect binding*, *spiral binding*, and *wire-o binding*.

electronic document

A document stored in a computer instead of printed on paper.

electronic mail

Correspondence in the form of messages sent between workstations over a network. Synonymous with *email*.

electrophotographic

Pertaining to a type of printer technology that creates an image on paper by uniformly charging the photoconductor, creating an electrostatic image on the photoconductor, attracting negatively charged toner to the discharged areas of the photoconductor, and transferring and fusing the toner to paper.

em

In composition, a unit of measurement exactly as wide and as high as the point size of the font being set. The name is derived from the fact that the letter M in early typefaces was usually cast on a square body.

email

Electronic mail.

email destination

In InfoPrint Manager, an actual destination representing an electronic mailing system.

email destination support system

In InfoPrint Manager, the destination support system that supports email destinations.

embellishments

Variable data added to all copies of assembled pages to make the unit appear like a whole; for example, headers, footers, a table of contents, and chapter separations.

en

In composition, exactly one-half an em.

enable

In InfoPrint Manager, the action that makes a destination, queue, or server able to accept jobs, or a log able to accept information.

end sheet

The sheet of paper, blank or otherwise, that can follow a job. See also *auxiliary sheet*.

Enhanced X-Windows

A tool designed to permit multiple application processes to operate in multiple windows displayed on a virtual terminal. See *X-Windows*.

environment variable

Any one of a number of variables that describe the way an operating system runs and the devices it recognizes.

error log

A data set or file in a product or system where error information is stored for later access.

estimate

The professional cost analysis made by a print shop in response to a customer's request for a price quotation on a print job.

event

In InfoPrint Manager, an occurrence in the printing system during an operation; for example, the completion of a command.

event log

In InfoPrint Manager, a collection of messages about events that have occurred.

event notification

The notification that is sent by InfoPrint Manager for an event.

F

Federated Authentication

A technology that grants users secure access to InfoPrint Manager relying on external identity providers (IdPs). Instead of managing separate user credentials within the InfoPrint Manager system, federated authentication allows users to log in using their existing accounts from trusted third-party services.

FIFO (first-in-first-out)

In InfoPrint Manager, a queuing technique where the next job assigned to the first position in the queue is the job that has been in the queue for the longest time. InfoPrint Manager processes jobs in the order in which they are received. Contrast with *deadline*, *job-priority*, *longest-job-first*, and *shortest-job-first*.

file-reference document

In InfoPrint Manager, a file containing the names of other files, each entered on a separate line. Job submitters can specify this file for printing when they specify a document type of `file-reference`; InfoPrint Manager prints each file listed in the reference document.

File Transfer Protocol (FTP)

In TCP/IP, the application protocol that makes it possible to transfer data to and from host computers and to use foreign hosts indirectly.

finisher

A hardware device attached to a printer that performs such operations as folding or stapling the printed pages.

finishing

In a print shop, the final operations on a printed product, such as stapling, trimming, drilling, folding, embossing, varnishing, gluing, shrink wrapping, perforating, laminating, collating, and so on.

flag

A modifier of a command that specifies the action of the command. A dash usually precedes a flag. Synonymous with *option*. See also *keyword*.

FOCA

Font object content architecture.

folder

In the InfoPrint Manager GUI, an object that represents a container for a collection of similar objects. For example, the Retained Jobs folder contains retained jobs.

font

A family of characters of a given size and style; for example, 9-point Helvetica.

One size and one typeface in a particular type family, including letters, numerals, punctuation marks, special characters, and ligatures.

A paired character set and code page that can be used together for printing a string of text characters. A double-byte font can consist of multiple pairs of character sets and code pages.

form definition

A resource object used by InfoPrint Manager that defines the characteristics of the form or printed media, which include: overlays to be used, the paper source (for cut-sheet printers), duplex printing, text suppression, and the position of composed-text data on the form.

forward space

In InfoPrint Manager, the action used to move ahead and skip the printing of a specified number of pages in a job.

FPO

Low-quality (sometimes photographic) images placed in a dummy to represent final images. Desktop publishing software produces images as display-screen resolution FPOs.

front panel

In the CDE, a workspace area containing controls that represent various tasks you can perform and workspace switches.

FST

Files and Sockets Transport is the local security implementation for InfoPrint Manager. It uses the local namespace for user credentials, and it is a lightweight security protocol.

FTP

File Transfer Protocol.

G

GCR

Gray component replacement.

GIF

Graphics Interchange Format.

global change

In the InfoPrint Manager GUI, an action used to make changes to one or more attributes of several objects at once. You can also perform the same action on several objects of the same object class at the same time; for example, disabling two or more actual destinations at the same time.

global character

A character or set of characters used to specify an unknown number or set of characters in a search string. In InfoPrint Manager, a global character is represented by an asterisk (*).

global ID

Global job identifier.

global job identifier

An unambiguous job identifier. In InfoPrint Manager, it is represented as the name of the server managing the job followed by a colon (:) and a generated integer. This ID uniquely identifies the job in the InfoPrint Manager server.

glyph

An image, usually of a character, in a font.

GOCA

Graphics object content architecture.

graphic character

A visual representation of a character, other than a control character, that is normally produced by writing, printing, or displaying.

Graphical User Interface (GUI)

A type of user interface that takes advantage of a high-resolution monitor and includes a combination of graphics to implement an object-action paradigm, the use of pointing devices, menu bars, overlapping windows, and icons. Contrast with *Command Line Interface*.

Graphics Interchange Format (GIF)

A compressed graphics format widely used on the Internet.

gray component replacement (GCR)

Synonym for *under color removal (UCR)*.

gray scale

A strip of standard gray tones, ranging from white to black, placed at the side of the original copy during photography to measure the tonal range and contrast obtained.

GUI

Graphical User Interface.

gutter

The blank area or inner margin from the printing area to the binding.

H

halftone

A printing method that simulates continuous-tone shading using dots of varying sizes laid out on a rectangular grid. Larger dots simulate darker tones and smaller dots simulate lighter tones. Typical grid spacings are 85 lines per inch (lpi) (newspaper), 133 lpi (low end), 150 lpi (midrange), and 175+ lpi (high quality).

help

In the InfoPrint Manager GUI, an action used to display the online help for a specific template, object, action, button, control, or area in an application window.

The name of a button used to access the online help.

hold

An indication determined by the **job-hold** attribute that is used to keep a job in the queue and prevent InfoPrint Manager from scheduling it.

host name

A unique label assigned to a device (such as a computer, server, or router) on a network, used to identify it. It can be a simple name (such as *laptop-01*) or part of a fully qualified domain name (FQDN) like *server.example.com*, allowing devices to locate and communicate with each other.

hot folder

A workstation directory where users copy jobs to submit them to print.

hypertext

A way of presenting information online with connections between one piece of information and another. These connections are called hypertext links. See also *hypertext link*.

hypertext link

A connection between one piece of information and another. Selecting a link displays the target piece of information.

I

icon

A graphic symbol displayed on a screen that a user can click to invoke a function or software application.

image

Toned and untoned pels arranged in a pattern.

image data

Rectangular arrays of raster information that define an image.

imagesetter

A high resolution (1270–3600+ dpi) printer that uses an Argon (green) laser to write to film using digital input. Imagesetting is the step before Cromalin proofing and platemaking.

imposition

The process of arranging pages on a press sheet to ensure the correct order during final cutting, folding, and binding. Electronic imposition builds press sheets automatically during the RIP and outputs film formatted for immediate use in platemaking.

InfoPrint

A solution of hardware products that can supplement or replace the offset presses and copiers in print shops with high-quality, non-impact, black and white or process color printers.

InfoPrint Manager

A print management product that runs on an AIX, Linux, or Windows operating system. InfoPrint Manager handles the scheduling, archiving, retrieving, and assembly of print jobs and their related resource files. It also tracks the finishing and packaging of the printed product.

InfoPrint Manager Select

Synonym for *InfoPrint Select*.

InfoPrint Manager Submit Express

Synonym for *InfoPrint Submit Express*.

InfoPrint Network

The local area network running TCP/IP protocol that InfoPrint Manager uses to communicate among servers, clients, and output devices.

InfoPrint Select

The component of InfoPrint Manager that lets you submit and track jobs from your system to a destination that is managed by InfoPrint Manager.

InfoPrint Submit Express

The component of InfoPrint Manager that lets you submit and track jobs with a job ticket from a Windows workstation.

InfoPrint 4000

A duplex, black and white, continuous-forms printer with 600-dpi resolution.

initial value document

Synonym for *default document*.

initial value job

Synonym for *default job*.

initial value object

Synonym for *default object*.

initially settable attribute

An attribute whose value can be established when an object is created but cannot be subsequently set or modified. See also *resettable attribute*; contrast with *non-settable attribute*.

input focus

The area of a window where user interaction is possible from either the keyboard or the mouse.

input tray

For a printer device, the container that holds the medium upon which the printer prints its output.

Intelligent Printer Data Stream (IPDS)

An all-points-addressable data stream that enables users to position text, images, and graphics at any defined point on a printed page.

Information that the host sends to IPDS printers. This information generally contains basic formatting, error recovery, and character data and enables the printers to make decisions.

An architected host-to-printer data stream that contains both data (text, image, graphics, and bar codes) and controls defining how the data is to be presented. IPDS provides a device-independent interface for controlling and managing all-points-addressable (APA) printers.

International Organization for Standardization (ISO)

An organization of national standards bodies from various countries established to promote development standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

Internet

A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the protocol for sending information.

Internet Protocol

The set of rules that determines how to route data from its source to its destination in an internet environment.

intervening jobs

In InfoPrint Manager, the number of jobs in a queue that are scheduled to print before the job in question.

IOCA

Image object content architecture.

IP address

The IPv4 or IPv6 address.

IPDS

Intelligent Printer Data Stream.

ISO

International Organization for Standardization.

J

job

In InfoPrint Manager, an object that represents a request to print or send one or more documents together in a single session. A job includes the data to be printed or sent and resources such as fonts, images, and overlays. Depending on how it was submitted, it can also include a job ticket. Synonymous with *job bundle* and *print job*.

job bundle

Synonym for *job*.

job data

The page descriptions, merge data, and embellishments that compose a document in a job, either directly or by reference.

job ID

A local or a global identifier that identifies a job to a job submitter, administrator, operator, or InfoPrint Manager. See *local job identifier*, *global job identifier*.

job-priority

In InfoPrint Manager, a queuing technique where the next job assigned to the first position in the queue is the one with the highest priority. Contrast with *deadline*, *FIFO (first-in-first-out)*, *longest-job-first*, and *shortest-job-first*.

job submitter

In InfoPrint Manager, the person who submits jobs for printing. Often, this person is an application programmer who maintains applications that generate data to be printed.

job ticket

The customer's hardcopy or electronic instructions listing all the variables describing a print job, either directly or by reference. The print shop can add specifications to the job ticket and can print the job ticket.

Joint Photographic Experts Group (JPEG)

A compressed graphics format widely used on the Internet.

JPEG

Joint Photographic Experts Group.

K

kerning

In typesetting, adjusting the relative spacing of two characters to improve their appearance and readability. Kerning pairs are specific sets of characters with built-in relative spacing. Some typefaces have as many as 100 kerning pairs.

keyword

A name or symbol that identifies a parameter.

Part of a command operand that consists of a specific character string, such as `DSNAME=`.

kitting

In a print shop, packaging the completed published work with attendant binders, tabs, diskettes, and other equipment or information, before shipping the finished product.

L

LAN

Local Area Network.

laser (light amplification by stimulated emission of radiation)

In InfoPrint printers, a device that emits a beam of coherent light that forms the image on the photoconductor that is subsequently transferred to the paper.

late binding

In InfoPrint Manager, waiting to assign a job to an actual destination until it is about to be processed. Late binding permits InfoPrint Manager to route a job to the first suitable actual destination that becomes available. Contrast with *early binding*.

LDAP

Lightweight Directory Access Protocol is the network security implementation for InfoPrint Manager. It uses an LDAP or Active Directory server for user credentials, and it offers a unified security implementation in a customer environment.

LED

Light-emitting diode.

light-emitting diode (LED)

The imaging device element for electrophotographic print units.

lines per inch (lpi)

A measure of the density of the grid used to space halftone dots. Typical grid spacings are 85 lpi (newspaper), 133 lpi (low end), 150 lpi (midrange), and 175+ lpi (high quality).

Linux

Linux is an open-source operating system modeled on UNIX. There are multiple distributions available, InfoPrint Manager is supported on AlmaLinux, Red Hat Enterprise Linux (RHEL), Rocky Linux, and SUSE Linux Enterprise Server (SLES).

Local Area Network (LAN)

A computer network at one location that consisting of devices connected to one another and communicating. This network can also be connected to a larger network.

local ID

Local job identifier.

local job identifier

In InfoPrint Manager, a job identifier automatically generated by the server, identifying the job to the person who submitted it. InfoPrint Manager maps a local job ID to a global job ID.

locale

The human language and character set of information presented to a user.

In InfoPrint Manager, the language InfoPrint Manager uses when sending notification and error messages or displaying the InfoPrint Manager graphical user interfaces.

log

A collection of messages or message segments added to a file for accounting or data collection purposes.

To record messages in a file.

logical destination

In InfoPrint Manager, an object to which users submit their jobs. The logical destination routes jobs to one or more actual destinations representing output devices such as printers, or electronic mail systems. See also *logical printer*; contrast with *actual destination*.

logical printer

In InfoPrint Manager, a type of logical destination. The logical printer routes jobs to one or more physical printers representing printing devices.

logical unit (LU)

A type of network accessible unit that enables end users to gain access to network resources and communicate with each other.

logical unit (LU) 6.2

A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

longest-job-first

In InfoPrint Manager, a queuing technique where the next job assigned to the first position in the queue is the longest job in the queue. Contrast with *deadline*, *FIFO (first-in-first-out)*, *job-priority*, and *shortest-job-first*.

lpi

Lines per inch.

LU

Logical unit.

M

magnetic ink character recognition (MICR)

Identification of characters printed with ink that contains particles of magnetic material.

mainframe processor

A functional unit that interprets and executes instructions in a large computer to which other computers are connected so that they can share the facilities the mainframe provides.

makeready

All the setup work involved in preparing a press for a print run.

manage

In the InfoPrint Manager GUI, the action used to put an object into a mode where its icon reflects changes of status.

mechanical

A camera-ready layout. The mechanical can consist of multiple sheets or overlays for each spot or process color used. Final images, if not stripped in later, should be at final size, cropped and screened to the correct line frequency.

medium

In InfoPrint Manager, an object representing the physical material upon which a job is printed.

merge data

Data that is unique to each copy of a document and that customizes the document; for example, serial numbers or mailing information. Merge data is typically a small percentage of the total data in the document.

message catalog

A file of all the possible messages than can display during the processing of an application.

MICR

Magnetic ink character recognition.

Mixed Object Document Content Architecture (MO:DCA)

An architected, device-independent data stream for interchanging documents.

MO:DCA-P

Mixed Object Document Content Architecture Presentation.

modify

In InfoPrint Manager, an action used to modify the values of attributes in the object attributes notebook.

moire

Undesirable interference patterns caused by two overprinting halftone screens with incorrect halftone dot angles.

monospaced

In typesetting, pertaining to a typeface in which all the characters have equal widths. Monospaced typefaces are useful for tabulating figures.

Multiple Virtual Storage (MVS)

An operating system developed by IBM. The design of MVS incorporates an addressing architecture that provides a unique address space to each job in the system.

MVS

Multiple Virtual Storage.

N

N_UP

Pertaining to the number of forms placed together in a layout. Typical layouts are 2_UP, 4_UP, 8_UP, 16_UP, and so on. N_UP printing is done to use the maximum area of the print sheet.

namespace

A global name repository available to all utilities and API procedures. The namespace contains mappings of object names to other related objects. For example, the namespace provides the mapping of a logical destination to the server in which it is located.

Network File System (NFS)

A protocol that uses Internet Protocol to allow a set of cooperating computers to access each other's file systems as if they were local.

newline options

The different ways in which the printer determines how lines are delimited in a document data stream.

NFS

Network File System.

non-process-runout (NPRO)

A printer function that moves the last printed sheet to the stacker of the printer device.

non-settable attribute

An attribute that is neither initially settable nor resettable. The values for these attributes are controlled by InfoPrint Manager. Contrast with *initially settable attribute* and *resettable attribute*.

notification

The act of reporting the occurrence of events.

In InfoPrint Manager, notification of events appears as messages in event logs or messages sent to administrators, operators, and job submitters. In the InfoPrint Manager GUI, notification of events also appears as changes to the appearance of icons.

notification-profile

In InfoPrint Manager, an attribute associated with an object that contains information designating the people to whom InfoPrint Manager sends notification about events for that object, which event information it sends, and how it sends the information.

NPRO

Non-process-runout.

O

object

A collection of attributes that represent a physical or logical entity in the print system. For example, a specific printer device is represented by an actual destination (physical printer) object. An object is identified by its object name. Objects are grouped into classes. See also *object class*.

object class

A group of objects that share a common definition and therefore share common properties, operations, and behavior as defined by their attributes. For example, all InfoPrint Manager queue objects are in the same object class and each queue has the same set of queue attributes. However, the values for those attributes can differ for each queue in the queue object class.

Object Identifier (OID)

In architecture, a notation that assigns a globally unambiguous identifier to an object or a document component. The notation is defined in international standard ISO.IEC 8824(E).

object name

The alphanumeric term that identifies an object.

object state

The state of an object indicates its availability and readiness for performing its functions. An object can be in one of a number of states such as ready, busy, or unknown.

OCR

Optical character recognition.

octet

A byte that consists of eight binary digits (bits).

offset stacking

In certain printer devices, a function that allows the printer to offset the printed output pages for easy separation of the jobs.

OID

Object Identifier.

open destinations window

In the InfoPrint Manager GUI, the action used to open a new application window displaying the logical and actual destinations associated with a queue.

Open Prepress Interface (OPI)

An industry standard for replacing low-resolution images in review documents with high-resolution images needed for high-quality final output.

Open Software Foundation (OSF)

A nonprofit research and development organization created by a consortium of companies that work together to develop software in the open systems market.

OpenType font (OTF)

An extension of the TrueType font format that adds:

- Support for PostScript outlines
- Better support for international character sets
- Broader support for advanced typographic control

open window

In the InfoPrint Manager GUI, the action used to open a new application window representing one or more objects displayed in the currently open application window.

operation

An action performed on one or more data items.

operator

In InfoPrint Manager, the person responsible for printer devices. Also, this person performs a subset of tasks for InfoPrint Manager queues and actual destinations and performs some job-related tasks.

OPI

Open Prepress Interface.

optical character recognition (OCR)

Conversion of scanned text to editable ASCII characters.

option

A modifier of a command that specifies the action of the command. A dash usually precedes an option. Synonymous with *flag*. See also *keyword*.

orphan logical destination

In the InfoPrint Manager GUI, an object that represents a logical destination that is not associated with an existing queue.

orphan logical printer

In the InfoPrint Manager GUI, an object that represents a logical printer that is not associated with an existing queue.

OSF

Open Software Foundation.

overlay

A collection of constant data, such as lines, shading, text, boxes, or logos, that is electronically composed in the host processor and stored in a library, and that can be merged with variable data during printing.

OTF

OpenType font.

P

PAC

Privilege Attribute Certificate.

page definition

A resource that contains the formatting controls for line data.

In InfoPrint Manager, a resource that defines the rules of transforming line data into composed pages and text controls.

page segment

A resource that contains composed text and images, which are prepared before formatting and included during printing.

pane

In the Work Area of the InfoPrint Manager Administration GUI, an area containing a group of objects of a specific type, such as an actual destinations pane.

parent/child relationship

In InfoPrint Manager, servers, queues, and destinations are related to one another in a parent/child relationship. For example, a server is the parent of all the queues that reside in that server, and these queues are its children.

pass through

In InfoPrint Manager, referring to options submitted with a job that are used by the device driver, but not InfoPrint Manager. InfoPrint Manager does not process or validate this information; it passes it along to the device driver. See *printer driver*.

path

The route used to locate files; the storage location of a file. A fully qualified path lists the drive identifier (if any), directory name, subdirectory name (if any), and file name with the associated extension.

pause

In InfoPrint Manager, the action used to temporarily halt the printing or transmission of jobs on actual destinations or the distribution of jobs from servers or queues.

pdcreate

In InfoPrint Manager, the command used to create a new object and set its attributes to specified values.

PDF

Portable Document Format.

Printer description file.

pdmsg

In InfoPrint Manager, a utility used to display information about a message.

pdpr

In InfoPrint Manager, the command used to create and submit a job, consisting of one or more documents, to a server for printing or transmission.

perfect binding

A type of book binding in which the pages are held together with flexible adhesive. Contrast with *edition binding*, *spiral binding*, and *wire-o binding*.

permissions

In AIX, codes that determine who can access a file and what operations they can perform on the file.

physical printer

In InfoPrint Manager, a type of actual destination that represents a printer device. See also *printer device*.

piobe

The standard printer input/output backend program used by AIX for printing. See also *ainbe*.

pipe-pull

In InfoPrint Manager, a document transfer method where InfoPrint Manager saves the documents in a file and transfers the address of the file to the server. InfoPrint Manager transfers the file to the server upon the request from the server. This is an efficient transfer method for large jobs and is the default transfer method at job submission. Contrast with *with-request*.

plex

A document or actual destination attribute used to define the placement of output images on the page. See the plex values *simplex* and *tumble*.

Portable Document Format (PDF)

An Adobe PostScript data format that can be viewed or printed.

Portable Operating System Interface for Computer Environments (POSIX)

An Institute of Electrical and Electronics Engineers (IEEE) standard for computer operating systems.

POSIX

Portable Operating System Interface for Computer Environments.

PostScript

Adobe's page description language. PostScript has become a standard for graphic design and layout software. PostScript files can contain both vector and raster data. Fonts are described using PostScript coding. Many desktop publishing systems produce PostScript data as their output data stream.

PostScript printer description (PPD)

A file that contains options for printing PostScript data on various printer devices.

PPD

PostScript printer description.

Prefix lengths

Identify a range of IPv6 addresses that are on the same network.

preflight

To assess all resources for a job before the actual print run.

prepress

Work done in the print shop before printing a job, such as preparing data and art, page imposition, color retouching, electronic editing, and page layout.

print database

The set of files on disk that provide a permanent repository for the attributes of all print objects, such as logical destinations, queues, and actual destinations.

print job

Synonym for *job*.

Print Quality Enhancement (PQE)

A printer facility that provides edge smoothing along diagonal lines, fine fidelity protection, and independent boldness control.

Print Services Facility (PSF)

Any of several programs, including PSF for AIX, PSF/MVS, PSF/VM, and PSF/VSE, that produce printer commands from the data sent to them.

print system

A group of one or more print servers and one or more printing devices, which might or might not be located in the same geographical area. The components of a print system are assumed to be interconnected in some manner, providing at least one network interface to print clients, and acting synergistically to supply the defined document print service. (D)

printable document

In InfoPrint Manager, an object that represents text or data to be printed by a job. Contrast with *resource document*.

printer description file (PDF)

A file that contains options for printing PostScript data on a specific printer device.

printer device

The physical output device that performs the printing function. See also *physical printer*.

printer driver

A file that describes the physical characteristics of a printer or other peripheral device. This file is used to convert graphics and text into device-specific data at the time of printing or plotting. Synonymous with *device driver*.

priority

In InfoPrint Manager, a number assigned to a job that determines its precedence in being printed. Jobs with higher priority numbers are handled before jobs with lower priority numbers.

process color

Color made up of CMYK links simulating a specified color. This is the traditional method of reproducing continuous tone color images (color separations). Because of the nature of color inks, certain inks do not reproduce well.

processor

In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

promote

In InfoPrint Manager, the action used to move a job to the beginning of the queue so that it will print on the next available printer that can handle that job.

protocol

A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

pSeries

A family of workstations and servers based on IBM's POWER architecture. They are primarily designed for running multi-user numerical computing applications that use the AIX operating system.

PSF

Print Services Facility.

PSF destination support system

In InfoPrint Manager, the destination support system that communicates with PSF for AIX to print jobs.

PSF physical printer

In InfoPrint Manager, the object representing a printer device that uses the PSF destination support system.

Q

queue

In InfoPrint Manager, an object that manages a collection of jobs that are waiting to print. A queue receives jobs from one or more logical destinations and sends jobs to one or more actual destinations.

A line or list formed by items waiting for processing.

R

raster

A pattern of dots defined at varying data depths. Black and white images are one-bit (on/off); grayscale images are 8-bit (256 levels); RGB images are 24-bit; CMYK images are 32-bit.

raster image processor (RIP)

A processor in which a PostScript data stream is converted into dot patterns that are transferred to the final print medium. InfoPrint Manager uses an Adobe RIP to convert PostScript to IPDS for such printers as the InfoPrint 4000.

rc.pd

In InfoPrint Manager, a utility used to start the InfoPrint Manager servers listed in the **/etc/pd.servers** file.

read-only

In InfoPrint Manager, describes an attribute value that cannot be set by the user but can change based on other activity.

ready

A state in which an object is ready and able to perform tasks.

An available resource, such as a value of the `media-ready` attribute. Contrast with *supported*.

red, green, blue (RGB)

The colors of CRT display phosphors. RGB images are for screen display only. They must be converted to CMYK images before printing.

registration

The accuracy of printing on paper relative to the edges of the paper.

The accuracy of printing a single color (cyan, magenta, or yellow) relative to black.

The accuracy of printing on one side of a page relative to printing on the reverse side of the page.

reprographics

The process of copying or duplicating documents or information.

requested

In InfoPrint Manager, pertaining to the specific values of attributes requested by a job when it is submitted. The requested attribute values of a job are validated against supported attribute values for an actual destination to determine if the printer can handle that job. Contrast with *supported*.

resettable attribute

An attribute whose value can be set or modified after an object has been created, assuming the object is in an appropriate state to be modified. See also *initially settable attribute*; contrast with *non-settable attribute*.

resource

In AFP, a file containing a collection of printing instructions used when printing a job. Resources include fonts, overlays, form definitions, page definitions, and page segments.

resource context

In InfoPrint Manager, an object that contains directory path information that helps the print system locate resources needed for printing a job. The resources include fonts, overlays, form definitions, page definitions, and page segments.

resource document

In InfoPrint Manager, an object that represents a resource, such as graphics or fonts, used by a job to print a printable document. Contrast with *printable document*.

resubmit

In InfoPrint Manager, an action used to reroute pending or retained jobs to a different logical destination than the one through which the job was originally submitted.

resume

In InfoPrint Manager, the action used to resume the printing of paused jobs or resume the distribution of jobs from paused servers or queues.

retained job

In InfoPrint Manager, an object that represents a job that is being stored in the print system for a specified amount of time, normally after the completion of printing the job. A retained job does not reside in a queue.

retention

The process of storing data after completion of a process for a certain length of time.

retrieval

The process of bringing digital archived data out of storage and into online memory for reuse. See also *archiving*.

RGB

Red, green, blue.

RIP

Raster image processor.

To convert data to dot patterns using a raster image processor.

root user

In the AIX environment, the system user with the most authority who can log in and execute restricted commands, shut down the system, and edit or delete protected files. Synonymous with *superuser*.

RPC

Remote Procedure Call.

RPM

Red Hat Package Management is the baseline package format of the Linux Standard Base distribution.

S

scanner

A device that converts hardcopy source data into digital format (halftone dots) to avoid retyping the data.

scheduler

In InfoPrint Manager, the scheduling method that the queue uses when assigning a job to an actual destination.

separator sheet

The sheet of paper, blank or otherwise, that separates documents in a job. See also *auxiliary sheet*.

server

In InfoPrint Manager, the object that accepts configuration, management, and printing requests, performs the requested operations, and sends responses back as a result of the operation.

settable attribute

See *initially settable attribute*, *resettable attribute*.

severity

An indication of how serious an error condition is.

shell

In the AIX operating system, a command interpreter that acts as an interface between the user and the operating system. In InfoPrint Manager documentation, all shell examples use the Korn shell.

shift-out, shift-in code

Control characters used to indicate the beginning and end of a string of double-byte, ideographic characters.

shortest-job-first

In InfoPrint Manager, a queuing technique where the next job assigned to the first position in the queue is the shortest job in the queue. Contrast with *deadline*, *FIFO (first-in-first-out)*, *job-priority*, and *longest-job-first*.

shut down

In InfoPrint Manager, the action used to halt all server or actual destination processes without deleting the server or actual destination.

signature

A group of pages that are printed, folded, cut, and bound together. Manual placement of pages in the signature is determined using a folding dummy.

simplex

In InfoPrint Manager, the value of the document or actual destination `plex` attribute indicating that output images are placed on the media in a head-to-head format, so that the top of one image is at the same edge as the top of the next image. Depending on the value of the document or actual destination `sides` attribute, the document can be printed on one or both sides of the paper. Contrast with *tumble*; see also *simplex printing* and *duplex printing*.

simplex printing

Printing on only one side of the paper. Contrast with *duplex printing (1)*.

Printing on one or both sides of the paper, placing output images on the media in a head-to-head format, so that the top of one image is at the same edge as the top of the next image. Contrast with *tumble duplex printing*; see also *duplex printing (2)*.

SMIT

System Management Interface Tool.

SNA

Systems Network Architecture.

spiral binding

A type of book binding in which wire or plastic coils are threaded through a series of holes or slots in the binding edge. Contrast with *edition binding*, *perfect binding*, and *wire-o binding*.

spot color

Individual colored inks formulated to exactly match a specified color. Spot color is used when CMYK process color cannot produce a reasonable facsimile of the color or when vivid color is needed. Spot color is also used when fluorescent or metallic colors are needed.

spread

In color printing, an area whose dimensions have been enlarged to eliminate white space between it and another color. Contrast with *choke*.

start sheet

The sheet of paper, blank or otherwise, that can precede a job. See also *auxiliary sheet*.

state

Synonym for *object state*.

stripping

The process of mechanically assembling film into plate layouts. Page imposition takes place during stripping.

subnet mask

Identify a range of IPv4 addresses that are on the same network.

subnetwork

Any group of nodes that have a set of common characteristics, such as the same network ID.

In the AIX operating system, one of a group of multiple logical divisions of another network, such as can be created by TCP/IP.

subtractive color system

A system that reproduces an image by mixing (adding) appropriate quantities of cyan, magenta, and yellow paints on white paper. These paints reflect certain colors and absorb—or subtract—others. Contrast with *additive color system*.

superuser

Synonym for *root user*.

supported

In InfoPrint Manager, pertaining to the specific job attribute values that the actual destination can accept during job validation. InfoPrint Manager validates the requested attribute values of the job against the supported attribute values of the actual destination to determine whether the actual destination can handle that job. Contrast with *requested*.

synchronous

Occurring with a regular or predictable time relationship to a specified action. Contrast with *asynchronous*.

system administrator

Synonym for *administrator*.

System Management Interface Tool (SMIT)

In the AIX operating system, an interface tool for installation, maintenance, configuration, and diagnostic tasks. SMIT lets you perform tasks without entering any commands.

Systems Network Architecture (SNA)

The description of IBM's logical structure, formats, protocols, and operational sequences for sending units through, and controlling the configuration and operation of, networks.

T

table reference character (TRC)

An optional control character in a print data set. The TRC identifies the font used to print the record and can be used to select a font during printing.

Tagged Image File Format (TIFF)

A digital format for storing scanned images. TIFF files are also referred to as raster format files (as opposed to vector format files). When used in desktop publishing, TIFF files produce only a low-resolution FPO image; the high-resolution data remains on the hard disk.

task help

A type of online help that provides a list of tasks that can be completed with a selected object. When you select a task, the help provides step-by-step information about how to complete the task.

TCP/IP

Transmission Control Protocol/Internet Protocol.

TCP/IP-attached printer

A printer device attached to an pSeries computer through telecommunication lines using the TCP/IP protocol.

In InfoPrint Manager, a PSF physical printer that represents a TCP/IP-attached printer device. See also *AIX-attached printer*, *direct-attached printer*, and *upload-TCP/IP-attached printer*.

template

In the InfoPrint Manager Administration GUI, an object that represents a set of default attribute values that has been defined for creating a particular type of object, such as an actual destination.

ticket

See *job ticket*.

TIFF

Tagged Image File Format.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A set of communication rules used in the Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet Protocol (IP) is the underlying protocol. See also *Internet Protocol*.

trapping

In desktop publishing, the amount of overlap in overprinting colors. Trapping prevents white paper showing through gaps caused by normal errors in registration. On sheet-fed presses, trapping is usually 0.25 point. See also *choke* and *spread*.

TRC

Table Reference Character.

TrueType font (TTF)

A font format based on scalable outline technology with flexible hinting. Glyph shapes are based on quadratic curves. The font is described with a set of tables contained in a TrueType Font File.

TTF

TrueType font.

tumble

In InfoPrint Manager, the value of the document or actual destination **plex** attribute indicating that output images are placed on the media in a head-to-toe format, so that the top of one image is at the same edge as the bottom of the next image. The document must be printed on both sides of the paper. Contrast with *simplex*.

tumble duplex printing

Printing on both sides of the paper, placing output images on the media in a head-to-toe format, so that the top of one image is at the same edge as the bottom of the next image. Contrast with *simplex printing (2)*, *duplex printing (2)*.

U

UCR

Under color removal.

under color removal (UCR)

Conversion of neutral color areas to black ink that reduces use of CMY inks, improves trapping, and reduces setup time. Generally, UCR is performed during image scanning, but some RIPs perform it during processing. Synonymous with *gray component replacement (GCR)*.

unmanage

In the InfoPrint Manager Administration GUI, the action used to put an object into a mode where its icon does not reflect the changes of status for the object.

upload printer

See *upload-TCP/IP-attached printer*.

upload-TCP/IP-attached printer

In InfoPrint Manager, a PSF physical printer that represents a printer device attached through an MVS system and managed by PSF/MVS. InfoPrint Manager communicates with the MVS system through the TCP/IP network. See also *AIX-attached printer*, *direct-attached printer*, and *TCP/IP-attached printer*.

V

validate

In InfoPrint Manager, to compare the attribute values requested by a job with the supported attribute values of the actual destinations in the system and to determine whether there is an actual destination capable of printing or sending the job.

value

A specific characteristic of an attribute.

variable

A name used to represent a data item whose value can change while the program is running.

variable data

Data that can be changed between copies of a document. For example, a form letter (constant data) can be merged with variable data, such as a customer's name and address to produce personalized documents.

varnish

A protective layer applied to a finished sheet, usually for photos, but sometimes used as a design element because of its reflective qualities. Varnishes can be tinted.

vector

An absolute coordinate point and line in space. PostScript files can contain vector artwork. Vector files are converted to rasters during the RIP process.

velox

A black and white photographic print made from a halftone negative, to be used as a proof copy.

vignette

An image with soft, fade-away edges.

Virtual Machine (VM)

A product that manages the resources of a single computer so that multiple computing systems appear to exist.

A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system. (T)

Virtual Storage Extended (VSE)

A program whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs.

Visual Systems Management (VSM)

In AIX, a type of graphical user interface that allows system management through the direct manipulation of objects.

VM

Virtual Machine.

VSE

Virtual Storage Extended.

VSM

Visual Systems Management.

W

web

A roll of paper used in web or rotary printing.

well

In the InfoPrint Manager Administration GUI, an area in a pane that contains a group of objects related to the objects in the pane; for example, a queues well in a servers pane.

what you see is what you get (WYSIWYG)

Refers to the fact that the composite image displayed on the screen at a computer workstation has the appearance of the final printed image.

window

A rectangular area of the screen that you can move about, place on top of, or pull under other windows, or reduce to an icon.

wire-o binding

A continuous double series of wire loops run through punched slots along the binding side of a booklet. Contrast with *edition binding*, *perfect binding*, and *spiral binding*.

with-request

In InfoPrint Manager, a document transfer method where the client transfers the documents directly to the server. Contrast with *pipe-pull*.

workstation

A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can use applications.

write access

A level of authorization that grants the ability to modify data.

WYSIWYG

What you see is what you get.

X

X-Windows

A network-transparent windowing system developed by MIT. It is the basis for Enhanced X-Windows, which runs on the AIX Operating System.

xerography

A dry printing process using corona-charged photoconductive surfaces to hold latent images that are developed with a dry toner and then transferred to paper and fused with heat.

Xstation

A terminal that is connected through a network to an pSeries computer, from which a user can perform command-line functions and run X-Windows based applications.

RICOH InfoPrint Manager™ for Linux Procedures

EN US EN GB EN AU 4/2026 G550-20264-27