RICOH

RICOH InfoPrint Manager™
for AIX and Linux

Configuring and Tuning Guide

Version 4.13.1

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

TABLE OF CONTENTS

9 Tuning for GUI Performance

10 Best practices recommendations

Useful AIX commands and tools

Useful Linux commands and tools

Temporary files created by InfoPrint Manager

Accessibility

Glossary

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 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 4.13 (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 R/3 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 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

AFP

Advanced Function Presentation

IP

Internet Protocol

PDF

Portable Document Format

PCL

Printer Command Language

GIF

Graphical Interchange Format

JPEG

Joint Photographic Experts Group

TIFF

Tagged Image File Format

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:

- AIX
- DFS
- IBM
- MVS
- OS/390
- POWER
- Print Services Facility
- pSeries
- S/390
- z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

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

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

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

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

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

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

Thales Group code: The license management portion of this Licensee Application is based upon one or more of the following copyrights:

```
Sentinel RMS
Copyright 1989-2024 Thales Group
All rights reserved.

Sentinel EMS
Copyright 2008-2024 Thales Group
All rights reserved.
```

Sentinel is a registered trademark of Thales Group.

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

1. Overview

InfoPrint Manager is a scalable, distributed, object oriented, client/server output management system. The basic parts of an InfoPrint Manager system include:

- Operator graphic user interface

- Administrator graphic user interface

- Job submission clients

- Command line clients

- InfoPrint Manager server
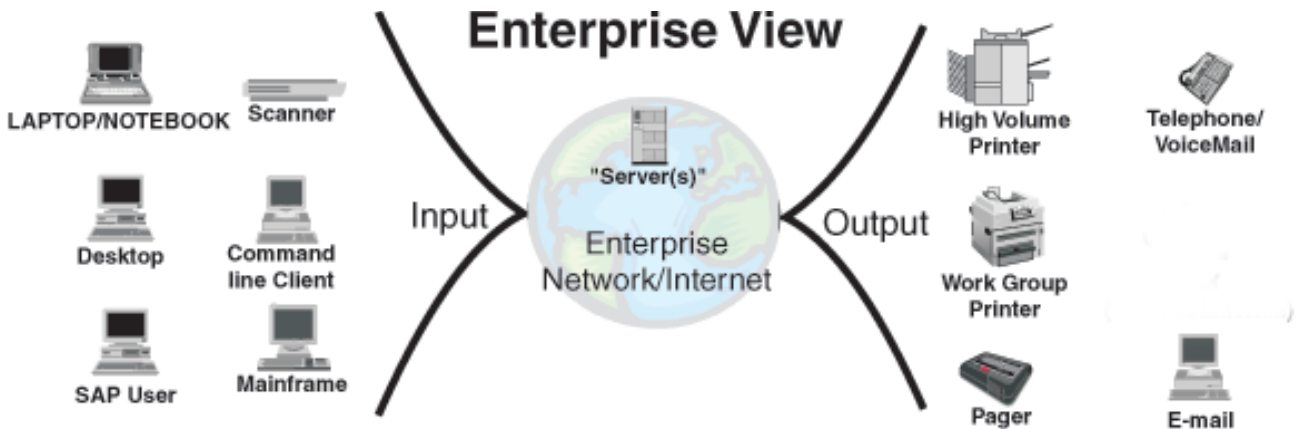
InfoPrint Manager provides a full set of printing objects including documents, jobs, logical destinations, queues, and actual destinations (printers, email).
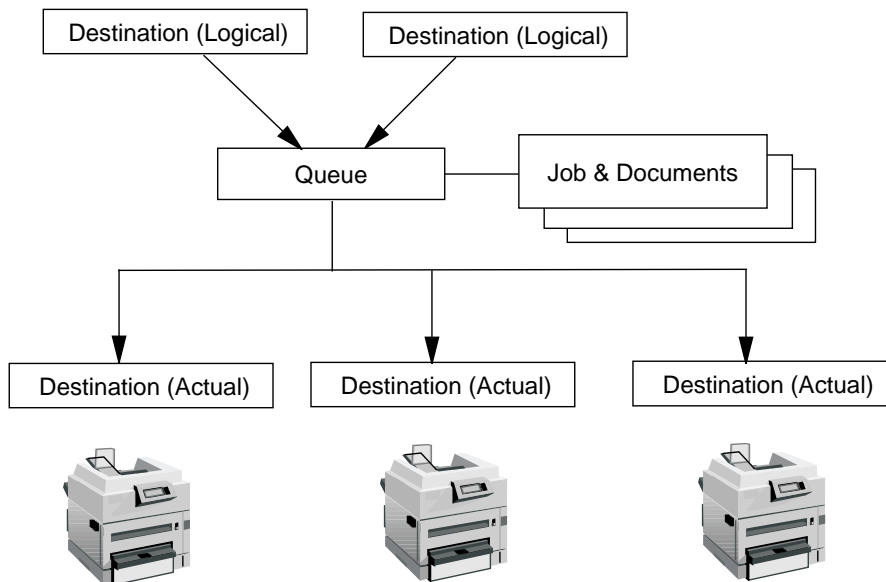
InfoPrint Manager provides two types of commands that you can use to access and manipulate objects: *end user* commands and *management commands*. End user commands allow users to print, query, cancel, and modify jobs. Management commands allow authorized users to create, delete, pause, resume, and change objects.
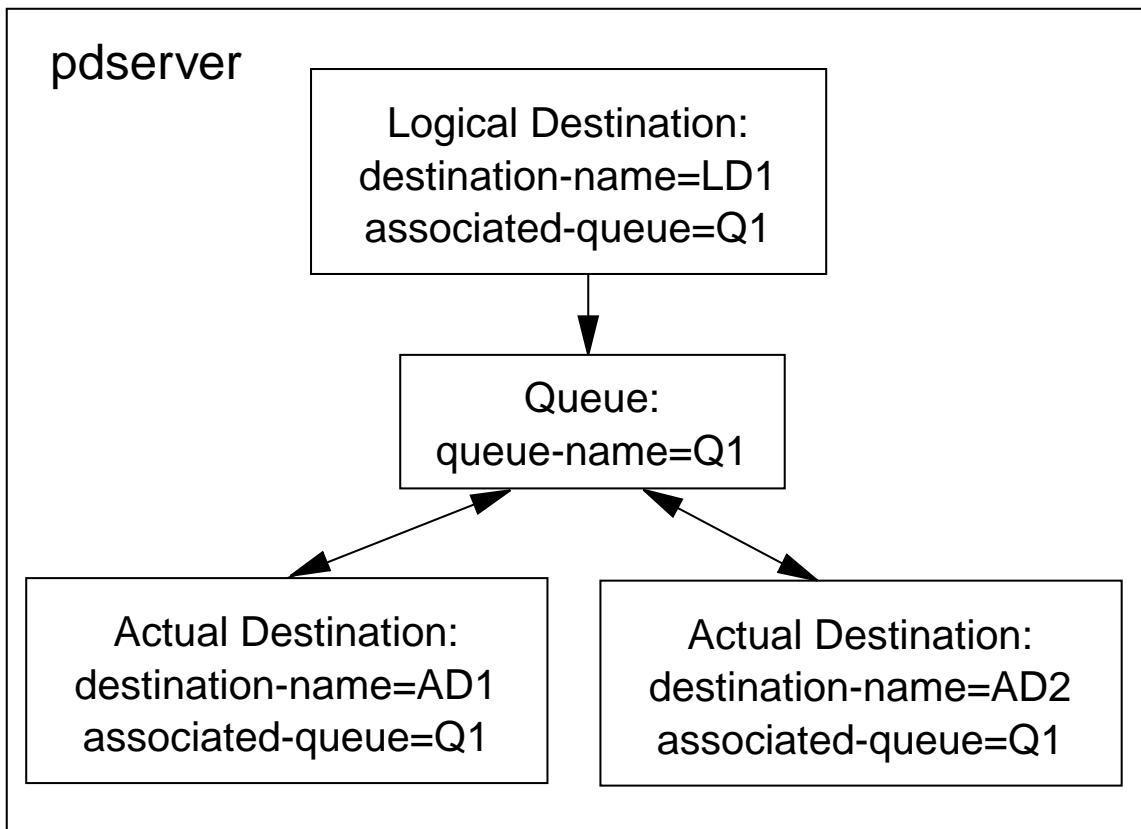
InfoPrint Manager objects have attributes and attributes have values. Object attributes and values allow for a precise definition of each object that includes defining the relationship of the objects to each other. How you define the relationship between InfoPrint Manager objects is part of your configuration. There are many ways to configure InfoPrint Manager objects.

InfoPrint Manager objects are contained in an InfoPrint Manager server. The InfoPrint Manager server is a multi-threaded process (pdserver) capable of performing many tasks concurrently.

InfoPrint Manager is scalable, which means that you can add more pdservers as they are needed to grow your system (on the same machine if necessary). InfoPrint Manager is also distributable, which means that you can add these pdservers on different machines located on your network. Since multiple pdservers on different machines can work together, you can distribute your InfoPrint Manager system, which gives you the ability to reduce single-points of failure, add remote servers, and centralize print management in a large enterprise. You can also view a "single system image" of the overall print system through the Administrator and Operator interfaces, the command line, or other job submission clients. Part of your configuration is designated by the number of pdservers that you define and how you distribute them among your machines.

The key element that allows multiple servers to work together is the InfoPrint Manager namespace. If you install InfoPrint Manager on two machines, each with its own namespace, there can be no communication or interoperation between the two InfoPrint Manager pdservers. You have two different InfoPrint Manager systems. In Two InfoPrint Manager Systems: independent pdservers using different namespaces with **no** communication between the pdservers., p. 11, the objects in the different systems can even have the same names. There is no single system view; clients must make a separate connection to each system to access the objects in each system.

Two InfoPrint Manager Systems: independent pdservers using different namespaces with no communication between the pdservers.

System 1

pdserver A
with
namespace

LD1
↓
Q1
↓
AD1

System 2

pdserver A
with
namespace

LD1
↓
Q1
↓
AD1
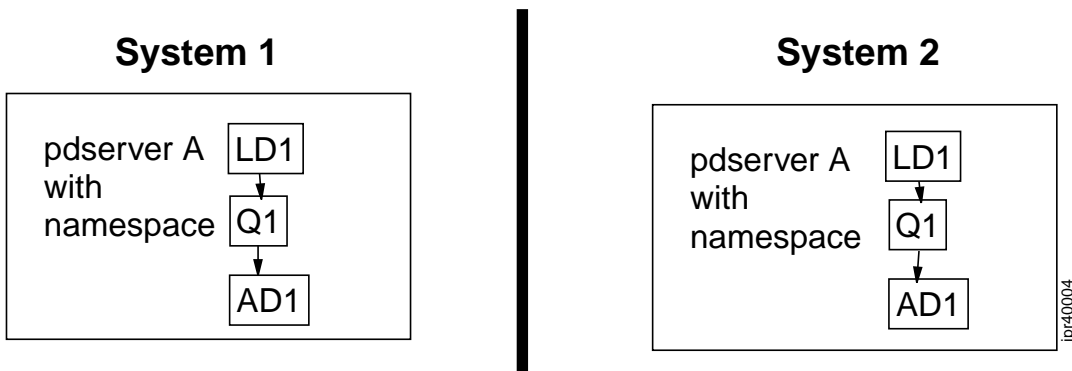
If you have multiple InfoPrint Manager servers that each use the same namespace, you have complete communication and interoperability between servers because you have one InfoPrint Manager System. All of the objects in the system are visible to all of the other servers and all of the clients connected to the system. In , notice that each object has a unique name, and the queue in pdserver A is feeding an actual destination in pdserver B (interoperating). Clients can access the objects in both pdservers (single image view) at the same time.

One InfoPrint Manager System: multiple pdservers using the same namespace

There is communication and interoperability between the pdservers.

pdserver A   LD1
             ↓
             Q1 ⟶ AD1

pdserver B   LD2
             ↓
             Q2
             ↓
             AD2
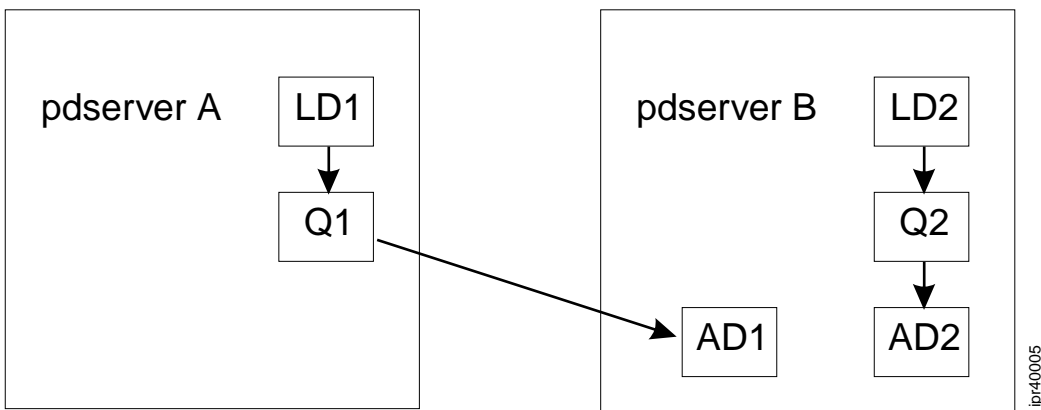
Daemons that forward jobs to InfoPrint Manager and run on the InfoPrint Manager server include:

* MVS Download Receiver daemon (**mvsprsd**)
* The **lpd** daemon that is shipped with the AIX operating system
* The **ipmlpd** daemon that is shipped with InfoPrint Manager for Linux
* IPP Gateway daemon

lists the InfoPrint Manager job submission clients and management clients. The command line client provides both job submission and management functions. Unlike InfoPrint Manager servers that can only work with the other InfoPrint Manager servers in the same InfoPrint Manager system, clients can be configured to work with any InfoPrint Manager System, and clients can switch back and forth between InfoPrint Manager systems.

InfoPrint Manager job submission clients and management clients

Clients used to submit jobs to InfoPrint Manager:	Clients used to manage InfoPrint Manager
• Select ((Linux, Mac OS X, Windows) • Submit Express • Download for z/OS Feature of PSF for z/OS • **lp**, **qprt**, and **enq** (AIX only) • SAP clients • IPP clients • **lpr** • Command line (AIX, HP-UX, Sun, Linux, Mac OS X, and Windows)	• Command line (AIX, HP-UX, Sun, Linux, Mac OS X, and Windows) • Operator interface • Administrator interface • SMIT Production Operator interface (AIX only) • IPMMI (InfoPrint Manager Management Interface - Linux Only) • MMC (Manager Management Console - Windows only)

The pdservers are the center of the InfoPrint Manager system. There are two types of InfoPrint Manager servers: the notification daemon, and the SAP Callback daemon (an optional feature). The other types of servers inter-operate with the pdserver to provide specific functions.

The notification daemon is used when InfoPrint Manager notifications are requested to be delivered by a "message". One notification daemon in an InfoPrint Manager system collects all of the end user "message" notifications from all of the InfoPrint Manager servers in the system and distributes them to the appropriate InfoPrint Manager end user notification client.
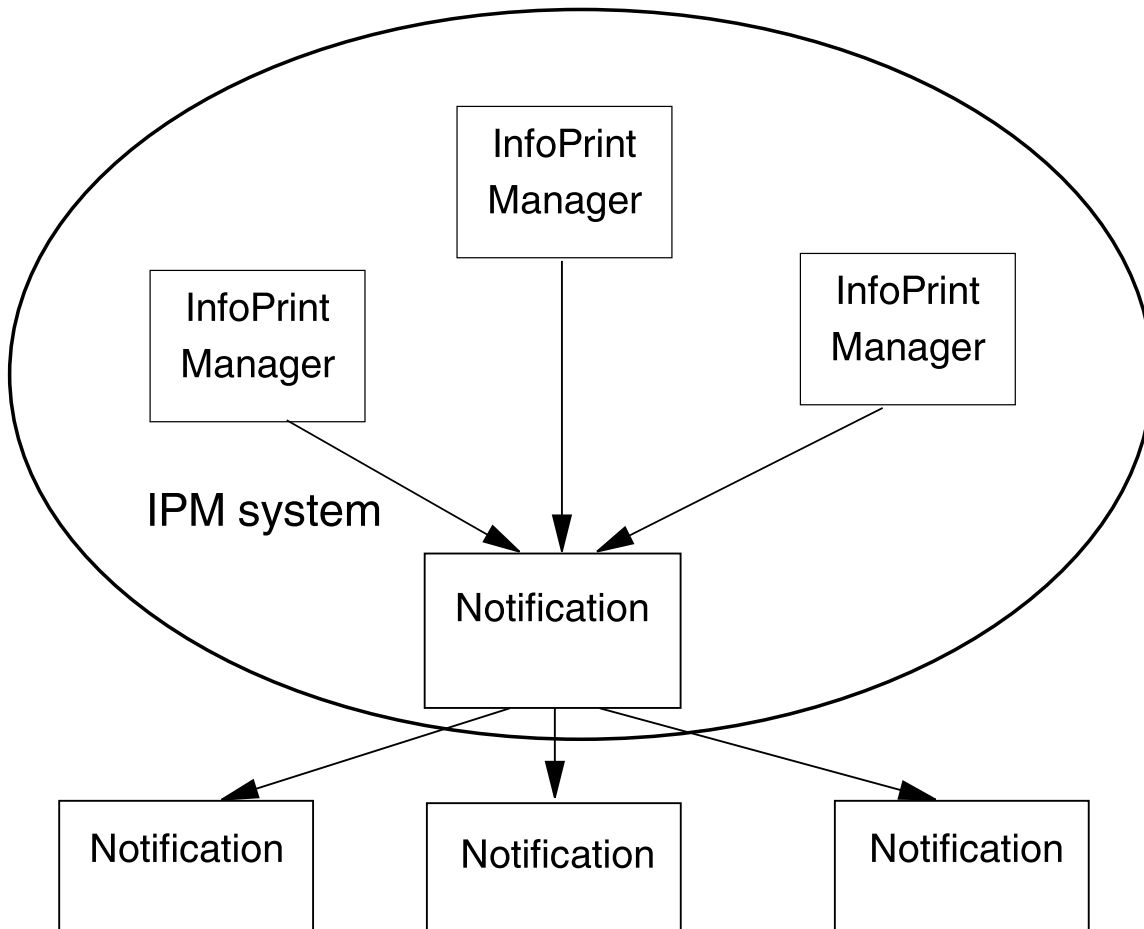
The SAP Callback daemon serves a similar function. It is used when notifications need to be delivered to an SAP System. One SAP Callback daemon for each SAP system sending jobs to an InfoPrint Manager system collects all the notifications for that SAP system from all the InfoPrint Manager servers in the system and delivers them to the SAP system.

InfoPrint Manager has a large set of data stream transforms and methods for driving printers and other output destinations, such as email. The methods include simple and basic passthrough methods (BSD DSS), standard AIX printing methods (AIX DSS), standard Linux printing methods (CUPS DSS), and the very powerful PSF printing methods (PSF DSS).

2. InfoPrint Manager System - interoperating InfoPrint Manager servers

- Parts of the server
- Server to server points of interoperation
- Types of server communication
- InfoPrint Manager namespace
- InfoPrint Manager commands

The InfoPrint Manager server is a multithreaded process (pdserver) capable of performing many tasks concurrently. Some of these tasks are related to the InfoPrint Manager objects that the server contains. Others have nothing to do with the contained objects. An InfoPrint Manager server with no objects can still serve a vital role in an InfoPrint Manager configuration.

InfoPrint Manager servers can interoperate with each other at many points.
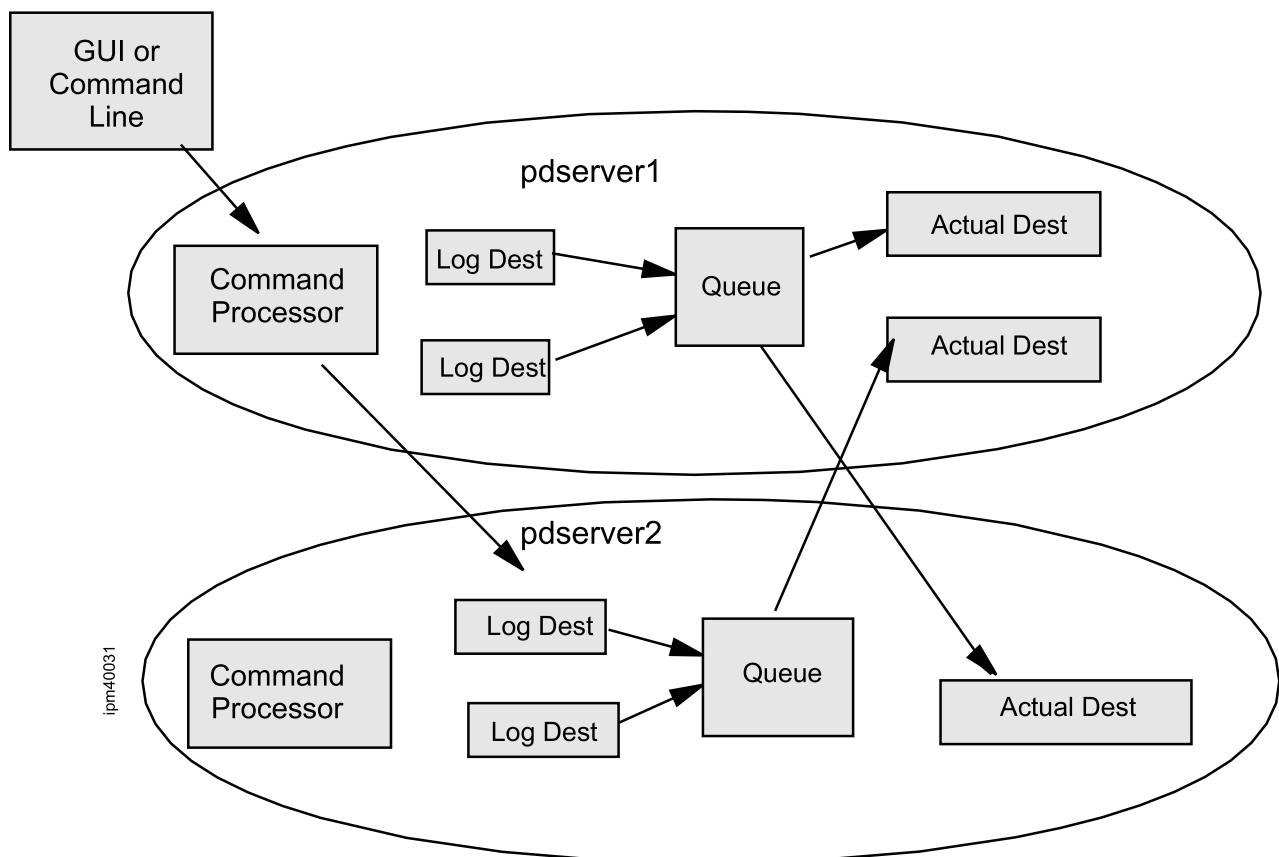
Not only can you resubmit jobs from one server to another, but you can also configure your system to have a queue in one server feed jobs to an actual destination in another server. And since you can access any object in the system through any pdserver, your clients can connect to the system through any server. This works because each server has three functional parts: command processor, spooler, and supervisor. Any given pdserver can act as a command processor, a spooler, a supervisor, or any combination of these.
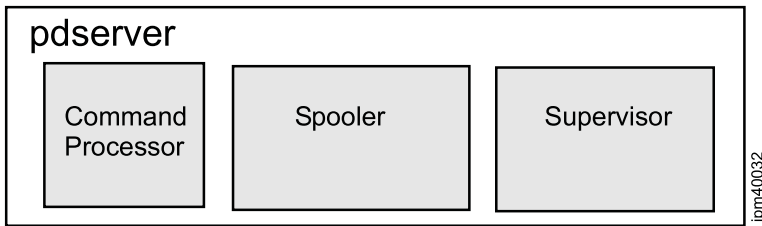
2 Parts of the server

Command processor

The command processor is the interface to all the clients and the submission daemons. Clients and submission daemons consist of the command line or InfoPrint Manager clients and submission daemons (such as the GUI, Select, Submit Express, or IPP Gateway), which use an internal, proprietary, object-oriented API (OOAPI). The command processor parses and analyzes requests from clients and daemons in a specific language, locates the pdserver owning the target object of the request, and sends the request (using an internal platform-independent protocol) to the pdserver containing the object. It then waits for the results and returns formatted results to the requester in their language. The command processor owns the copy of all print files for jobs submitted to it for any logical or actual destination in the system. If the actual destination processing the job cannot directly access the command processor's copy of the print file, the command processor provides the print file upon request to the pdserver containing the actual destination. It also does automatic data stream determination and owns the local to global ID mapping table. The local to global job id mapping tables allow users to refer to their jobs by 1, 2, or 3 instead of `serverName:123450000`. Each user can have a job 1. The command processor also manages and dispatches GUI events to all operator and administrator GUIs connected to it.

Spooler

The spooler provides the job management function. It owns and manages print jobs (not print files) including their creation and validation. It makes sure jobs are scheduled for printing to an appropriate actual destination based on the actual destination's scheduling algorithm and attributes. It holds retained jobs for possible reprint. And it sends notification for events that occur in the spooler to end users and to the command processor for dispatch to GUIs.
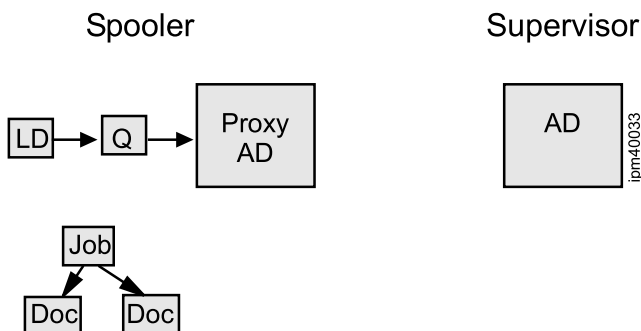
The spooler part of the pdserver contains jobs and document objects that represent the jobs to be printed; logical destination objects that are the front end to a queue; initial value job and initial value document objects that provide the defaulting for the job and document attributes; queue objects that handle the scheduling of jobs to actual destinations; and proxy actual destination objects that represent an actual destination object in some supervisor part of a pdserver. The actual destination could be in either the same pdserver or a different pdserver than the pdserver containing the proxy actual destination.

Supervisor

The supervisor provides printer management. It manages the actual destinations that handle the actual printing of the print job, and it sends notification for events that occur in the supervisor to end users and GUIs.
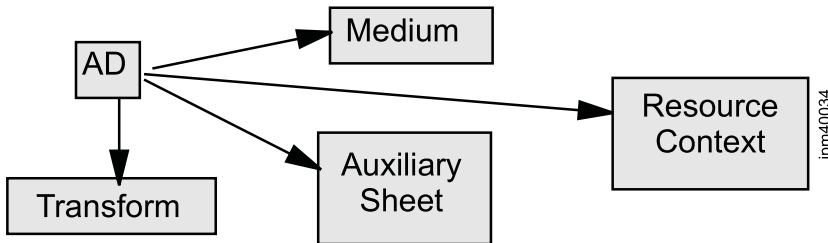
The supervisor part of the pdserver contains actual destination objects that represent printing, or email devices, auxiliary sheet objects that represent start, separator, and end sheets, medium objects that represent the material on which printing occurs, resource context objects that represent the location of some printing resources, and transform objects that represent transforms (and other programs that act on the print data). It also contains proxy job and document objects that represent the job and documents that have been scheduled to it to be processed and printed.

Note

1. Not all of these objects are used by every type of DSS.
2. The transform objects should not be confused with the automatic transforms used by the PSF DSS. See Print jobs and print files, p. 31 for more information.

A pdserver

In summary, it is important to understand that these are logical distinctions and remember that any server can play any combination of these roles.
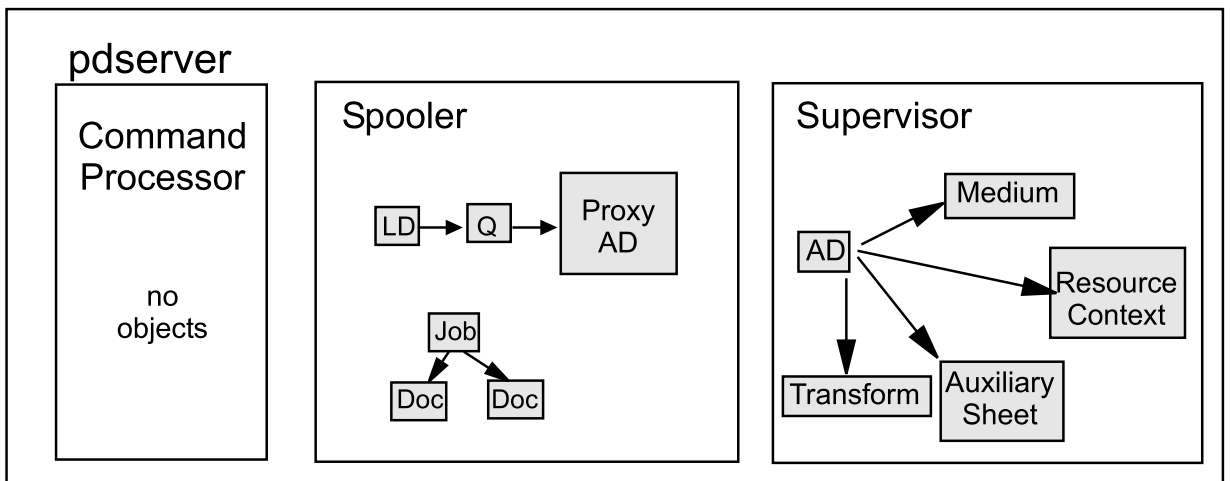
Server to server points of interoperation

Each part of a pdserver can interoperate with the other parts of the same or different pdservers. Command processors can talk to spoolers and supervisors. Spoolers can talk to supervisors, command processors, and other spoolers. Supervisors can talk to spoolers and command processors.

Let's look at some examples.

Here is an example of a command processor part communicating to a supervisor part in the same pdserver.
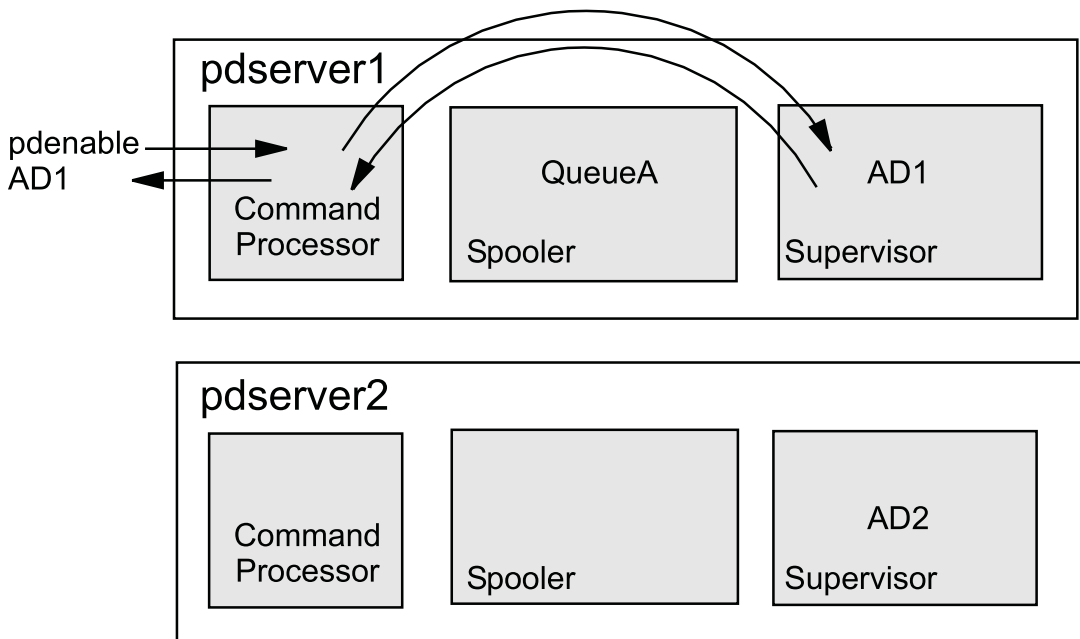
This example shows a command processor part communicating with a supervisor part in a different pdserver.
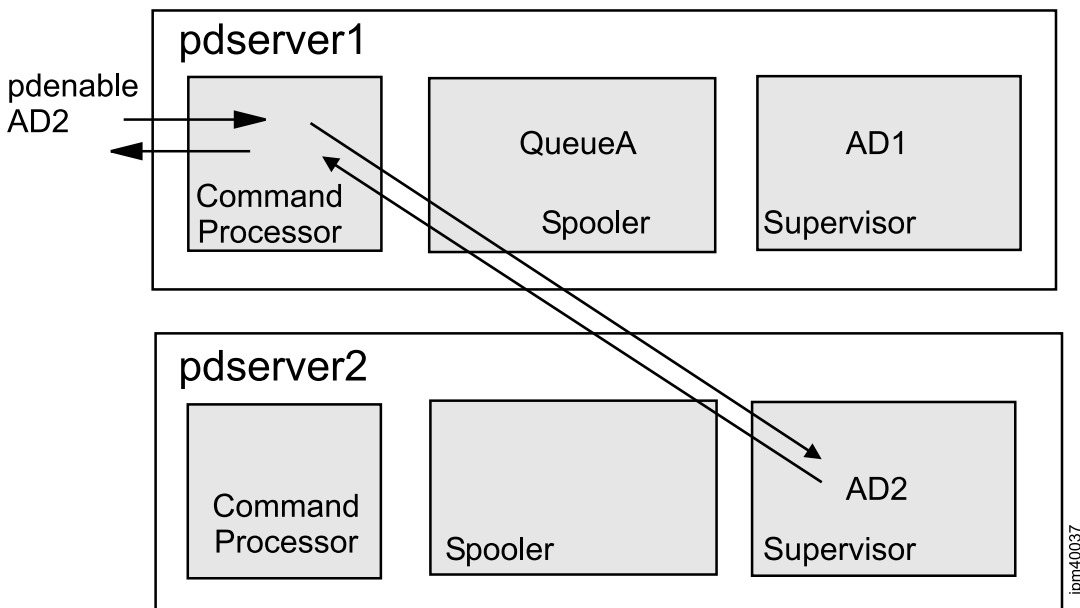
Here is an example of a command processor part communicating with a spooler part in the same pdserver.
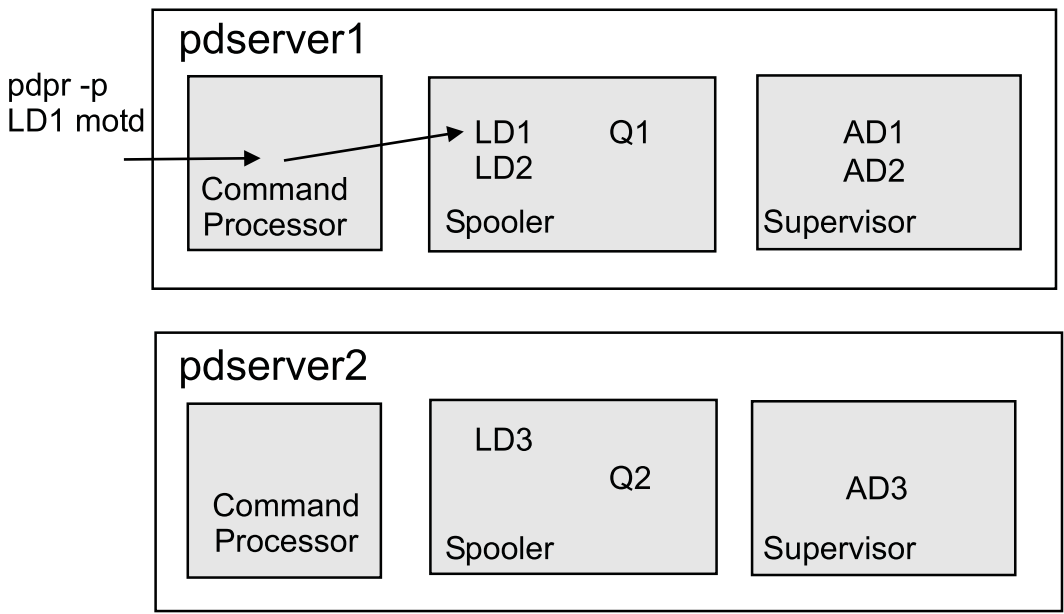
This example shows a command processor part communicating with a spooler part in a different pdserver.
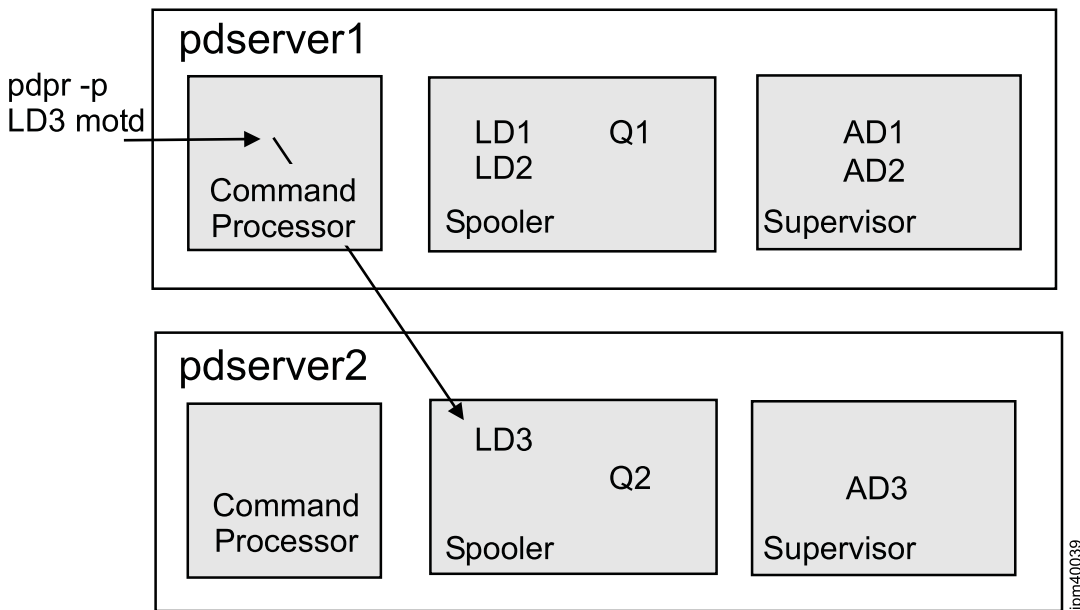
Here is an example of a supervisor part communicating with a spooler part in a different pdserver. In this case, an actual destination requests that its associated queue send it a job.

2
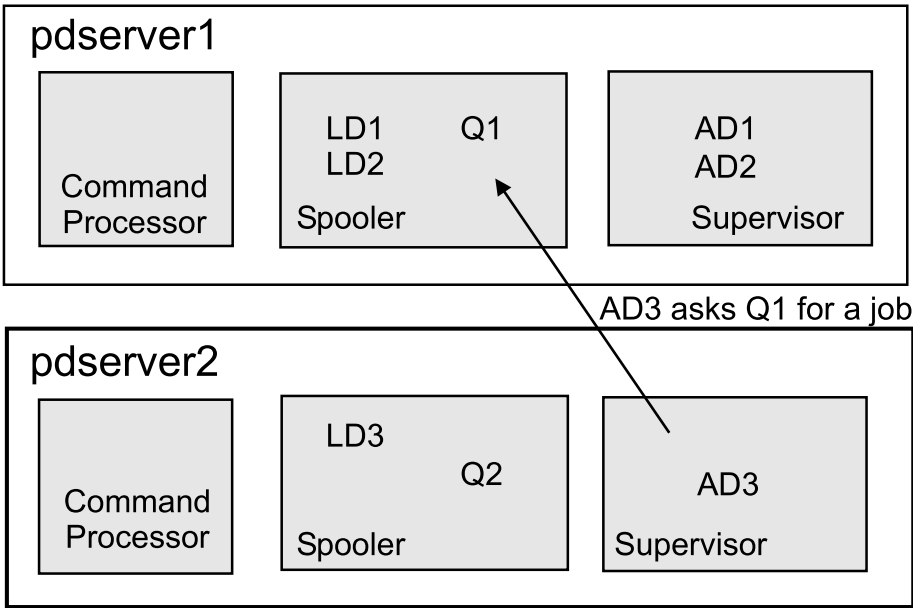
This is an example of a spooler part communicating with a supervisor part in a different pdserver. In this case, the queue associated with the actual destination sends a job to the actual destination for processing.
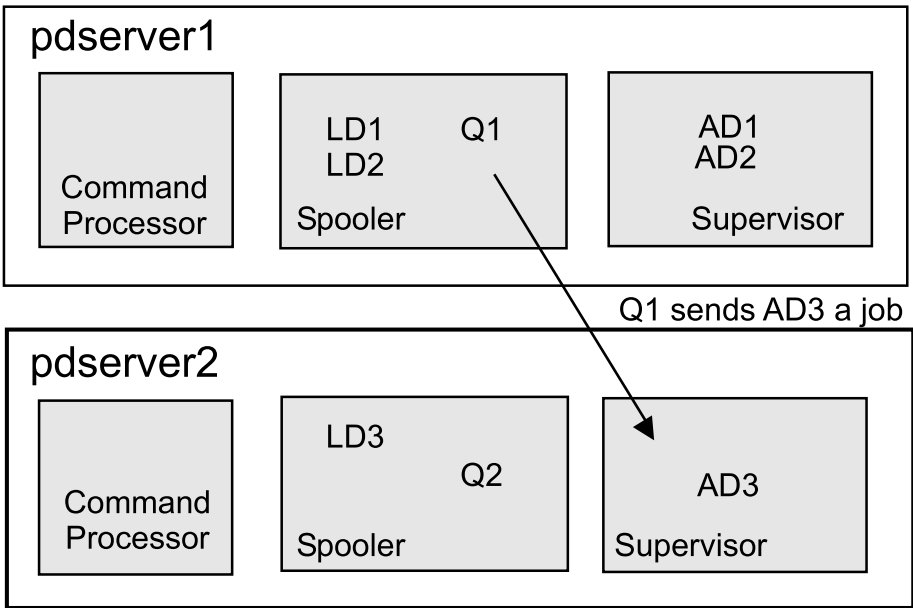
This next example adds print file handling to the example of a job submission. The command processor part of the pdserver copies the print file to its directory while the spooler part of the pdserver creates a document object (named 1234500000 in this example) that just contains information on where the print file is located. No file transfer occurs. If the job is scheduled to an actual destination on the same machine as the command processor, the file is accessed directly by the actual destination. If the pdserver containing the actual destination is on a different machine than the command processor pdserver, the actual destination's pdserver communicates with the command processor pdserver to "pull" the file over to its machine.

pdserver1

pdpr -p
LD1 motd

Command
Processor

LD1   Q1
LD2

Spooler

AD1
AD2

Supervisor

/var/pd/pserver1/pdb/document/1234500000.1
document-content=/var/pd/pserver1/pdprxxxxx
/var/pd/pdserver1/pdprxxxxx (copy of motd)

pdserver2

Command
Processor

LD3
        Q2

Spooler

AD3

Supervisor

ipm40070
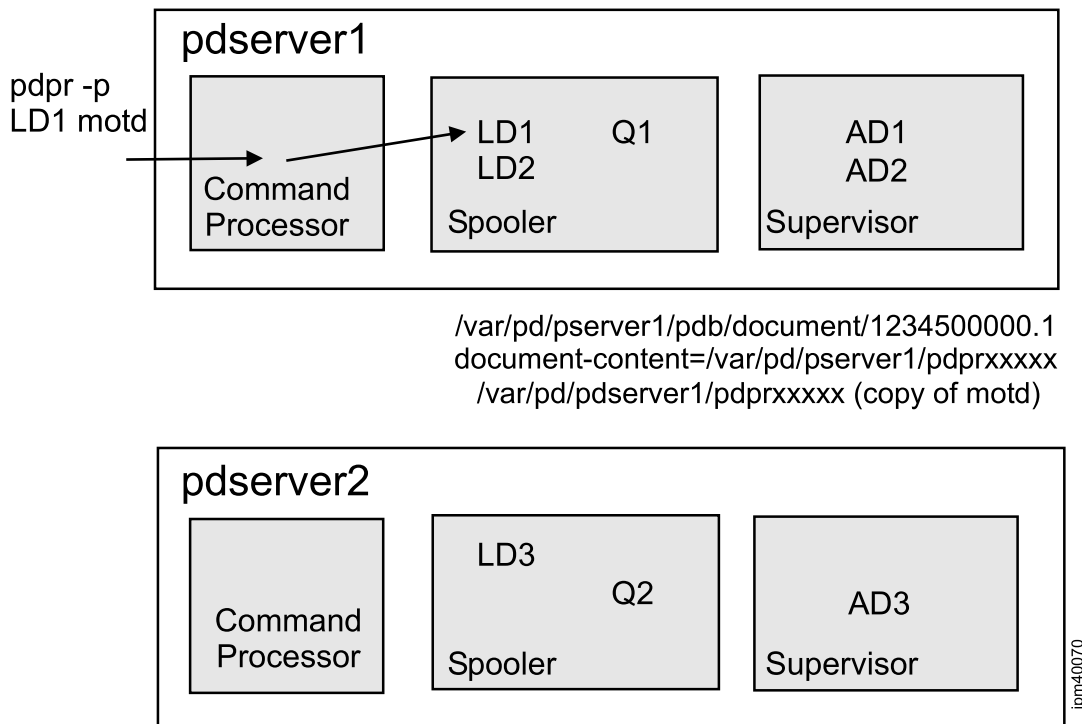
Here is an example of a supervisor part communicating with a command processor part in a different pdserver. In this case, the actual destination requests the print file for the job from the command processor that processed the original **pdpr** request.

pdserver1

Command
Processor

LD1   Q1
LD2

Spooler

AD1
AD2

Supervisor

AD3 requests print file for job

pdserver2

Command
Processor

LD3
        Q2

Spooler

AD3

Supervisor

ipm40042
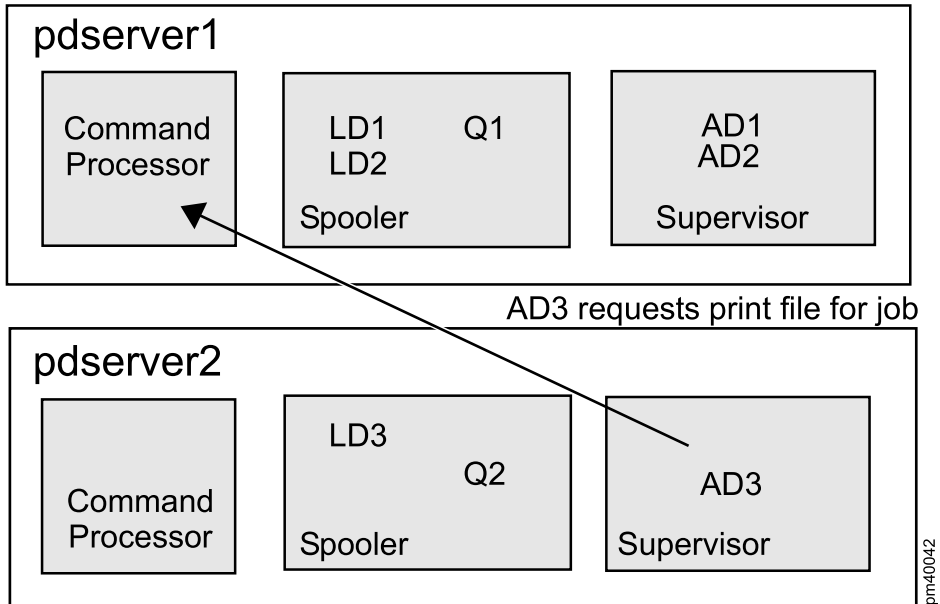
For more information about how InfoPrint Manager handles print files, see .

Types of server communication

Now that we understand how pdservers work together, we can talk about the two types of server communications. InfoPrint Manager clients always communicate with the command processor part of the pdserver, and this is called *client to server* communication. When any of the three parts of the pdserver communicate with any other part of a pdserver this is called *server to server* communication.

⬇ Note

When different parts of the same pdserver communicate, the communication is optimized.

For clients to work with a pdserver, they need:

1. A way to locate a pdserver
2. A way to communicate with a pdserver

For pdservers to work together, whether they are on the same machine or different machines, they need:

1. A way to locate each other
2. A way to communicate with each other

Ways to communicate

Sockets and ports are the key to both client-to-server communication and server-to-server communication. A socket is a TCP/IP connection between two ports. The command processor part of each pdserver listens for client requests on a unique port. The client or submission daemon uses a special command line protocol or an internal object-oriented API (OOAPI) protocol used by InfoPrint Manager non-command line clients. The next numbered port (client-to-server port + 1) is used by the command processor to send events to the client.

Each pdserver also listens for server-to-server requests on third unique port. These server-to-server requests use an internal platform-independent protocol that only InfoPrint Manager servers understand. This protocol not only lets InfoPrint Manager servers communicate with each other on the same or different machines, but also on the same or different platforms: AIX, Linux, and Windows.
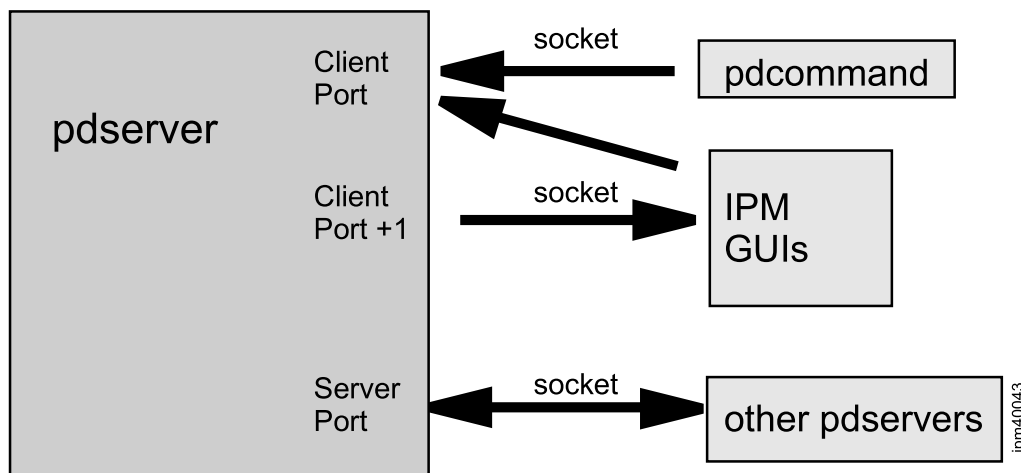
Ways to locate

InfoPrint Manager clients and submission daemons locate a pdserver by using the host name of the server and the client port number that the server is listening on. They use their own stored information, the `/var/lib/pd/pdserver.conf` file, or two environment variables:

1. PDHOST defines which machine the server is on

2. PD_SOCKET defines which port the server is listening for client requests on

InfoPrint Manager servers locate each other by using the InfoPrint Manager namespace. The *location* information in the namespace includes:

1. Host name on which the pdserver is running
2. TCP/IP port number on which the pdserver is listening for requests from other pdservers

pdserver client ports details

A pdserver's client ports are defined by the **-p** flag on the **start_server** or **startsrv** command or by the PD_SOCKET setting when the server is started. The default is **6874** and is defined in `/etc/services`. Since each pdserver uses two consecutive ports if multiple pdservers are started on the same machine, the client port specified for the pdserver to use must be incremented by at least two to avoid conflicts. When a pdserver starts, it adds its client port in `/var/lib/pd/pdserver.conf` along with its locale (its language).

In the normal case where PD_SOCKET is not set, clients on the InfoPrint Manager server machine discover what port to use by looking in `/var/lib/pd/pdserver.conf` for the first entry in their locale (the user's language). If PD_SOCKET is set, it overrides any values in `/var/lib/pd/pdserver.conf`. Since clients on non-InfoPrint Manager server machines don't have access to the `/var/lib/pd/pdserver.conf` file, they use the PDHOST and PD_SOCKET environment variables or other configuration mechanisms to obtain this information. See the information for each client to determine how to configure.

⬇ Note

Clients should be configured to point at a pdserver running in the same language as the user. If this is not possible, they should be set to point to a pdserver running in English.

Some examples might help illustrate the effect of the server selection in the client. All objects in the InfoPrint Manager system are accessible to the client; not just the objects in the pdserver the client connects to the system through. Also, the client can be running anywhere on the network; not just where the InfoPrint Manager server is installed. You can also switch a client to use a different InfoPrint Manager system by changing he IP address or the port settings in the client.

With PD_SOCKET set to port 6874
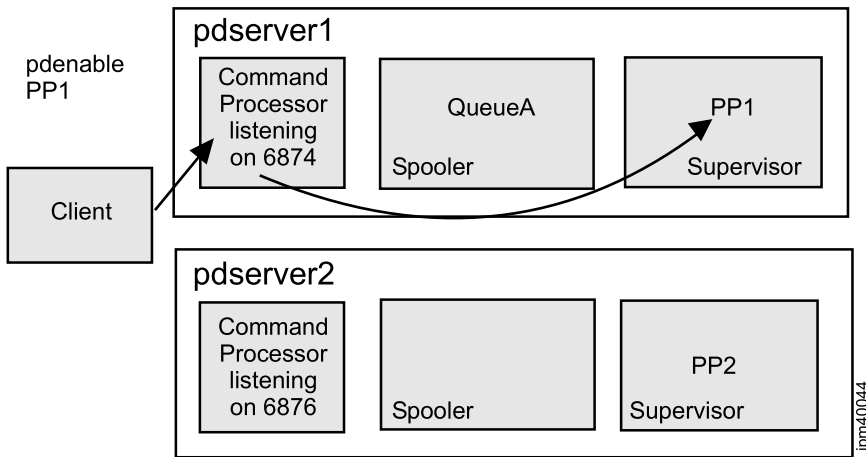
With PD_SOCKET set to port 6876
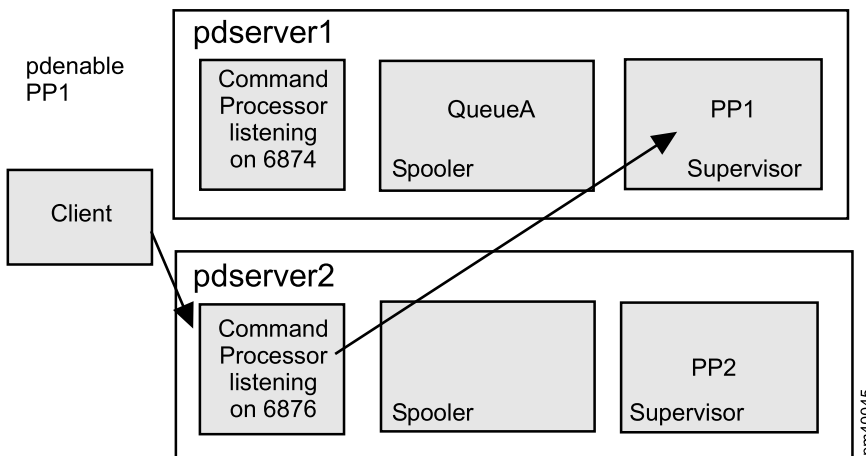
/var/lib/pd/pdserver.conf file details

The mechanism used on an InfoPrint Manager for AIX or InfoPrint Manager for Linux server machine that automatically connects a client or submission daemon on the same machines to a command processor is the file `/var/lib/pd/pdserver.conf`. This file contains a list of the running pdservers on the machine, the port the command processor part of the pdserver is listening on, and the locale in which the pdserver is running. When a command is invoked, the first command processor port in the list in the user's locale is used.

When a pdserver is started, it adds its entry to the list. When a pdserver is shutdown, it removes its entry from the list. If a pdserver didn't have a chance to removes its entry from the list and it is no longer running, you could receive a `5010-452 Cannot communicate with the communication daemon on port ___` error message. The key here is to check the port number returned in the message and make sure that it is the port number you were expecting to be used. Then make sure that a pdserver is running

on that port. A pdserver might have crashed or been killed, and it needs to be restarted or have its entry removed from the `/var/lib/pd/pdserver.conf` file.

You can always override the `/var/lib/pd/pdserver.conf` file by setting the PD_SOCKET environment variable. In fact, if you are running multiple pdservers on the same machine and want to control which pdservers are being used as command processors, this is a very good way to do it. Otherwise, you must always make sure the pdservers are started in the right order so the command processor pdserver is the first entry in `/var/lib/pd/pdserver.conf`.

2

pdserver server port details

InfoPrint Manager sets aside and assigns pdserver server-to-server ports automatically. Just like the default client port, these ports are assigned on each machine and not per InfoPrint Manager system. This means that a pdserver on two different machines can be listening on the same server port number just like they can be listening on the same client port number (6874). By default, ten ports are selected starting with the lowest block of ten available above 2000 in `/etc/services`. Each InfoPrint Manager server is assigned a port from the available ports when it first starts. If possible, it reuses that port on subsequent runs.

The **fstshow -o** command can be used to view the current server ports for a machine running InfoPrint Manager servers. For example, **fstshow -o** might display:

```
$ fstshow -o
IN-USE? PORT# PID    DIR-ID  NAME
------- ----- -----  ------  ----
   Y    2058  23364      2   info1
   Y    2059  15310      0   notifyd
   Y    2060  32724      2   info2
   Y    2061  33950      2   cmdsrv
   N    2062             2   info3   <- Currently not running
   Y    2063   4910      2   info4
   N    2064            99
   N    2065            99
   N    2066            99
   N    2067            99
```

ipm40046

The first column displays whether a server port is in use; the second column displays the port number; and the third column displays the process ID (PID) of the process using the port. Notice how `port 2062` is assigned to `info3`, but it is not currently running. Also the `NAME` column displays the name of the InfoPrint Manager server.

To start more than ten InfoPrint Manager servers (including the notification daemon and any SAP callback daemons) on one machine, you must add more server ports. If you try to start a server when there are not enough server ports available, you receive `5010-760 There are no available ports` error message. The server ports can be redefined (both the quantity and the starting number) by using **pdinitports**. All InfoPrint Manager servers on the machine must be shutdown to run **pdinitports**. See the *RICOH InfoPrint Manager: Reference* guide for more information about the **pdinitports** utility.

Each InfoPrint Manager server advertises its server port in the namespace when it starts. A pdserver discovers what server port to use to communicate with another object through the namespace.

InfoPrint Manager namespace

The namespace for the InfoPrint Manager system is created at install time. It is created in the **/var/ pddir/default_cell** directory by default. The PDNAMESP environment variable can be used to override the default location of the namespace. The namespace file system must be shared among all server machines in an InfoPrint Manager system. See *RICOH InfoPrint Manager for AIX: Getting Started* or InfoPrint Manager for Linux for information about setting up a shared namespace, and for information about setting up to share a namespace between InfoPrint Manager for AIX, InfoPrint Manager for Linux, and InfoPrint Manager for Windows .

Terminology Note

A *primary* server is simply the machine where the namespace file system is physically installed. *Secondary* servers are not less important; they just run on a machine where the namespace is NFS mounted or otherswise remotely accessed.

The main objects in the printing system are stored in the namespace. These include servers, destinations (both logical and actual), and queues. These types of objects must have unique names in the InfoPrint Manager system since only one entry for a given name for a type of object is allowed.

⬇Note

The namespace is also used to store security information.

Namespace entries

There are two kinds of entries in the namespace: profile and server. Server entries contain information about the IP address of the host on which the pdserver is running and the TCP/IP port number on which the pdserver is listening for server requests from other pdservers. Queue and destination entries are profile entries. Profile entries refer to the server entry for the pdserver in which they reside.

The **fstshow** command can be used to display information about server, queue, and actual destination entries in the namespace. The syntax for the command is:

```
fstshow [ -s | -q | -p ] entry-name
```

Where **s** represents server, **q** represents queue, and **p** represents destinations (formerly known as printers).

For example, if a pdserver named info2 is running in the InfoPrint Manager system, `fstshow -s info2` displays information like:

```
Type: Server
Dir Type: DIR_ID_SPOOLER_ENTRY
Name: info2
Listen Port: 2051
IP Addr: server.local
Client Port: 6874
```

The `Type:` field identifies this as a server entry. The `Dir Type` field identifies this as an entry for a server (formerly known as a spooler). The `IP Addr` field lists info2's host machine's IP address. The `Listen Port` field lists info2's server port. The `Client Port` field lists info2's client port but is only provided for information and is not used dynamically by any part of InfoPrint Manager. You can also determine the client port for a particular pdserver by entering a `ps -ef | grep pdserver` command on the machine

where the pdserver is running. The client port appears after the name of the pdserver. `Default` means it is running on port 6874.

Running the same command for a queue or actual destination would display a profile entry which would look like:

```
Type: Profile
Dir Type: DIR_ID_QUEUE_ENTRY
Name: info2-q
Server Path: servers/info2
```

The `Type:` field identifies this as a profile entry. The `Dir Type:` field identifies this as a queue. The `Name:` field identifies this as `info2-q`. And the `Server Path:` field identifies the server entry for the pdserver that contains this queue.

You must have read access to the namespace files to use the **fstshow** utility.

Lifecycle of the namespace entries

The queue and actual destination object places their name in the namespace when they are created and only remove them when they are deleted. The names remain in the namespace when their server is not running to help avoid another object with the same name being created. However, when a pdserver is restarted, it makes sure its objects' names are in the namespace, re-adding them if they have been removed.

Always delete objects using **pddelete**, so that their entries are removed from the namespace.

To function, a pdserver must have access to the namespace when it is started. If you receive `5010-845 Server cannot run with namespace at version __` when you attempt to start a pdserver, you should make sure that **/var/pddir/default_cell** is mounted or accessible.

Namespace caching

Every pdserver caches the namespace entries in memory to provide better performance and higher availability. The pdserver only refreshes its cache if an object is not found in its cache, it has a communication failure trying to access an object, or it is sent a "reset server" request from an InfoPrint Utility, such as the InfoPrint Manager SMIT interface. Once started, the pdserver is designed to continue functioning if it loses contact with the namespace. If the namespace becomes inaccessible (temporarily), the pdserver runs in cache-only mode. While in this mode, queue and destination objects cannot be created in the pdserver since they require the pdserver to write an entry in the namespace. The pdserver will also not attempt to refresh namespace entries. The pdserver periodically polls to see if the namespace is available so it can go back to its normal operating mode.

Namespace example

command
processor
pdserver

>pdls -c d ad05
(1)

(2a)
serv1

(2b)

Shared Filesystem
/var/pddir/default_cell/printers/
...
ad04
ad05
ad06
...

(3)   (4)

pdserver
containing
ad05

serv2

ipm40047

ad05

(1) Command received by command processor pdserver
(2a) Looks for object in local cache and doesn't find it (rare!)
(2b) Object found in namespace and "location info" read
(3) Command processor pdserver establishes communication
with pdserver containing the target object, ad05
(4) Data/messages returned to requester via command
processor pdserver

Since pdservers make sure their server, queue, and destination entries are in the name space when they are restarted, it is possible to clear a namespace for an InfoPrint Manager system if all pdservers are shutdown. The **clrfstns** command clears all dynamic information from the namespace. This command should only be used if all pdservers in the system are shutdown.

InfoPrint Manager servers include pdserver, **notifyd** (notification server for message notifications), **sapcbd** (SAP callback daemons for SAP notifications)

InfoPrint Manager commands

Now that we know we are conversing with an InfoPrint Manager system through a command processor and that this command processor locates certain InfoPrint Manager objects through entries in the namespace, we can better understand how the InfoPrint Manager commands work and how we can optimize use of them.

The InfoPrint Manager commands comprise a syntax that specifies actions, attributes, and objects.

Command syntax

This section describes the command syntax:

Command format

```
{command}{-flags …}{argument …}
```

Where:

- **command** is the name of a pdcommand
- **flags** are dash options
- **arguments** are the target of the command

Example

```
pdls -c queue Q2
```

The arguments are the target of the request or action. Each argument can have two parts: the server to which the request should be forwarded and the object in that server which is the target of the request.

Since the command processor can locate the server containing any object in the name space, you can issue commands directly against any queue, logical destination, actual destination, or server by simply using the objects name as the target for the command. For example:

```
pddisable  -c dest  ld2
pdls -c server spl2
```

The **pdcreate** command is an exception because when you use **pdcreate**, it is essential to identify the pdserver where you want to create the object.

All other InfoPrint Manager objects, such as, medium, initial-value-jobs, transforms, or auxiliary-sheets, do not have entries in the namespace. This means you must identify the pdserver that contains these objects when you run commands on them. To do this, you need to preface the object name with the server name followed by a separating colon (:), for example:

```
pdls -c initial-value-job  server2:myivj
```

The PDPRINTER environment variable that identifies a default destination to use for your command-line commands is also used to identify a default pdserver. If you have the PDPRINTER environment variable set, your commands for objects not in the namespace are sent to the server that contains the destination object defined. If you want your commands to go to another pdserver, you **must** qualify the target of the command with containing pdserver.

Optimizing your commands

Requesting all servers, objects, or attributes is easy, but results in a performance hit to your system.

Precede the object name with an asterisk (*) if you want a command to be sent to all pdservers in your namespace. For example, `pdls -c medium *:letter` sends the query to every server in the namespace. This is easy to do if you don't know the server you want, but results in more overhead, that is, more time, more cpu, and more memory. If you truly don't need the request to be sent to all pdservers, you should avoid the use of this option.

Placing a blank after the colon tells the server receiving the request to return information for all objects it has of this class. This is also easy to do especially since you can grep the results to find what you are looking for. This also results in more overhead on your server. You should learn how to use the filtering options on **pdls** to let the server locate just the information you need.

Requesting all the attributes (**-r all**) of objects is also easy to do. However, unless you want all the attributes, it is more efficient to request exactly what you need. You should learn how to use the **-r** (requested attributes) option of the **pdls** command to list just the attributes you need.

```
pdls -c dest -r media-ready AD1
```

If you need these functions (all servers, all objects, or all attributes), the InfoPrint Manager server will perform the request. However, just keep in mind what you are asking the server to do when you issue:

```
pdls   -c destination  -r all   *:
```

3. Print jobs and print files

To plan your configuration and hardware requirements, you need to understand how InfoPrint Manager handles print files and files generated while transforming and processing print files. What happens depends on how you submitted the job, what attributes you specify, what transform runs, and if you are running InfoPrint Manager on a single machine or multiple machines (sharing the namespace) and where you have defined your objects (logical and actual destinations).

Now that the different parts of the InfoPrint Manager server (pdserver) and how they interoperate are understood, you need to understand where a print job's print files are stored and how they are transferred around the system. This is the general flow of a print job in the system, and it applies to every job that is submitted to InfoPrint Manager, regardless of the destination support system used to process the job. Then you need to consider the specifics of the PSF destination support system that provides many additional functions that create additional files. These functions include automatic data stream transforms, pre-ripping files, and so forth.

A job object in InfoPrint Manager is created to represent each print job (**pdpr**) request. A job object contains information about the print job request. A job contains one or more document objects. A document object is created to represent each file specified on a job submission.

One of the primary design objectives of InfoPrint Manager minimizes the copying and movement of print files to minimize CPU, disk space, and additionally in a multiple server environment, network traffic usage.

These terms are useful in this description:

originating pdserver

> The pdserver the client is connected to that is functioning as the command processor for the print request. The pdserver that *receives* the print submission request. The **originating pdserver** manages the copy of or the link to the print files. This does not imply that the **originating pdserver** must contain the logical destination and queue; it can simply forward the request to the pdserver that does contain the logical destination and queue.

printing pdserver

> The pdserver that contains the actual destination object that prints the job. The **printing pdserver** can, if possible, directly access the **originating pdserver**'s copy of (or link to) the print files, or it can *pull* a local temporary copy of the print files from the **originating pdserver**. See Basic print file handling for all DSSs, p. 31 for more information.

🔵 Note

The information about originating pdservers and printing pdservers, their temporary files, and the general printing flow applies to ALL DSSs. In contrast, the information about transforms and their temporary files (see PSF, DFE, and Email print file handling, p. 36) applies only to the PSF DSS.

Basic print file handling for all DSSs

When a print request is made to InfoPrint Manager, the print file is sent to the pdserver running as a command processor (which then becomes the **originating pdserver**); this might be the pdserver containing the logical destination.

If the link option (`pdpr -1`) **is not** specified, which is the default, a temporary copy of the incoming print job is made by the **originating pdserver** in its `/var/pd/servername` directory with a randomly

generated name that begins with pdpr (for example, `pdpr1pp5nV`). The exception is for jobs submitted with InfoPrint Submit Express clients that use `/ipdata` as their work area. These jobs are treated as if the link option was specified. That temporary copy stays in `/var/pd/servername` on the **originating pdserver** until the print job completely leaves the system, including job printed or cancelled and expiration of any retention periods that have been set through InfoPrint Manager attributes. To put it another way, the temporary copy remains in `/var/pd/servername` on the **originating pdserver** until a `pdls` for that particular job returns with a message that InfoPrint Manager cannot find the job.

If the link option (`pdpr -l`) **is** specified, the `pdprxxxxxx` is a symbolic link to the original file (rather than a copy of). The link is deleted by InfoPrint Manager when the job completely leaves the system.

⬇ Note

The submission daemons keep a transient copy of the print file before it is sent to the InfoPrint Manager server. For example, MVS Download Receiver receives the file downloaded into a specified directory. After it is received, it is submitted to an InfoPrint Manager pdserver. Upon successful receipt of the file by the InfoPrint Manager server, the downloaded file is then deleted.

The job is submitted to the logical destination, and a job object is *stored* in the queue associated with the logical destination. There are no file transfers at this point; the Job object only contains a *reference* or pointer to the location of the print file. Each document in the job has a **document-content** attribute that contains information about the location of the document's print file. There are two pieces of information contained in this attribute:

- The actual location of the print file, for example, `/var/pd/CmdProcServ/pdprxxxxxxxxxxx`.
- The name of the command processor pdserver.
  This attribute will be displayed with a **pdls** command but only if specifically requested:

```
pdls   -r document-content   jobID
```

When the print file is scheduled to the actual destination one of two things can happen. If the **printing pdserver** that contains the actual destination can directly access the copy of the print file in the **originating pdserver's** `/var/pd/servername` directory, it uses the file directly and does not make its own temporary copy. However, if the **printing pdserver cannot** directly access the copy on the **originating pdserver**, the **printing pdserver** contacts the **originating pdserver** and requests a copy of the print file, which is known as a *file pull* or *pipe pull*. Files are *pulled* from one server to another using a socket. The **originating pdserver** must be running for a file to be pulled. The pulled file is stored in the **printing pdserver's** `/var/pd/servername/pdb` directory. The temporary copy on the **printing pdserver** only remains while the job is processing. As soon as the job leaves the actual destination object, the temporary copy on the **printing pdserver** is erased. Meanwhile, the first copy on the **originating pdserver** remains until the job completely leaves the system.

All environments must plan spooling space for one copy of each print file. The command processor's copy of the print files are not deleted until the job leaves the InfoPrint Manager system. In other words, until it is completed or canceled and the retention-period has expired. Multiple machine environments must also plan additional space for pulled files. Since these files only exist while a job is processing, less space is required.

Basic print file handling example

When a job is submitted to InfoPrint Manager:

1. The pdserver command processor copies the print file to its directory:

```
/var/pd/Serv2/pdprxxxxx
```

2. The job is submitted to the logical destination and a job object is stored in the logical destination's queue until scheduled for printing. No file transfer occurs; the job only contains reference information to the location of the print files.

Serv1

| | LD1 --> QueueA
Job1/Doc1 | AD1 |

/var/pd/Serv1/pdb/document/1234500000.1
doc-cont=/var/pd/Serv2/pdprxxxxx

Serv2

pdpr
-p LD1
myfile

| | | AD2 |

/var/pd/Serv2/pdprxxxxx (copy of myfile)

ipm40025
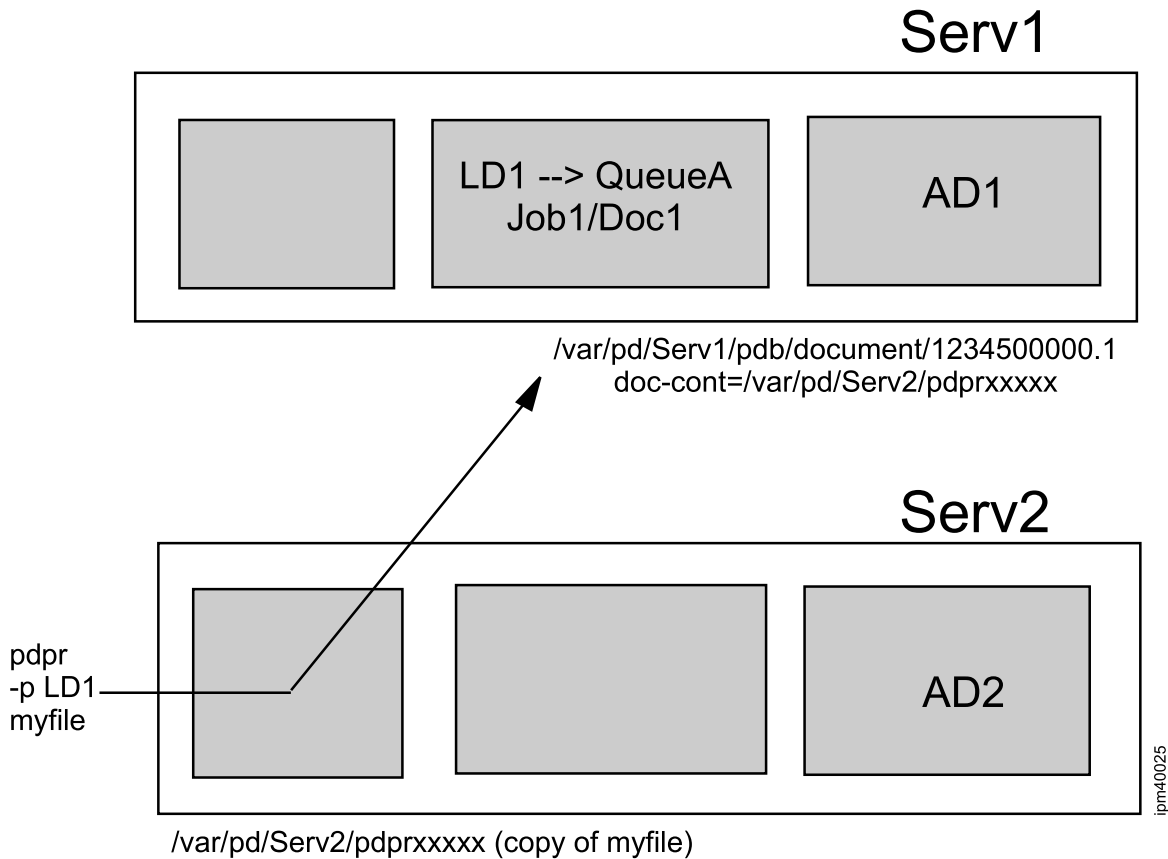
3. The job is scheduled and sent to an actual destination. The actual destination might be in a pdserver on the same machine or a different machine than the command processor pdserver. If the pdserver containing the actual destination is on the same machine as the command processor pdserver, the file is accessed directly by the actual destination.

(1) QueueA schedules job to AD2 in Serv2
(2) File is local to AD2 so it reads it directly
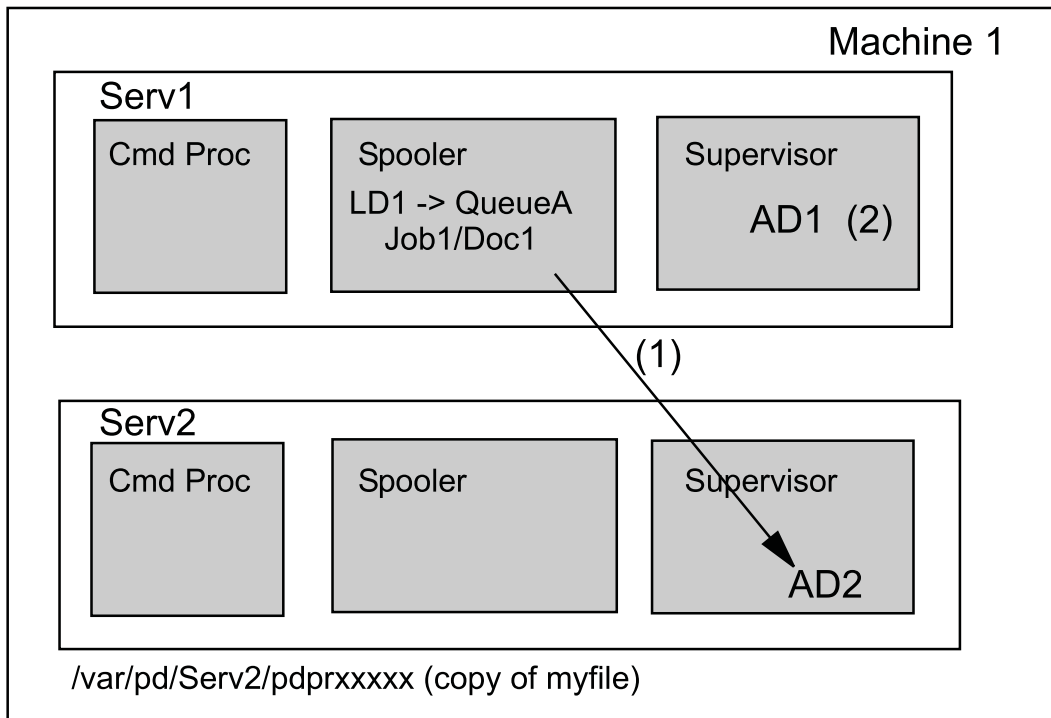
4. If the pdserver containing the actual destination is on a different machine than the command processor pdserver, the actual destination's pdserver communicates with the command processor pdserver to pull the file over to its machine.

(1) QueueA schedules job to AD1 in server on different machine than Serv2
(2) file is not local to AD1 so Serv2 is contacted and asked to send file to AD1
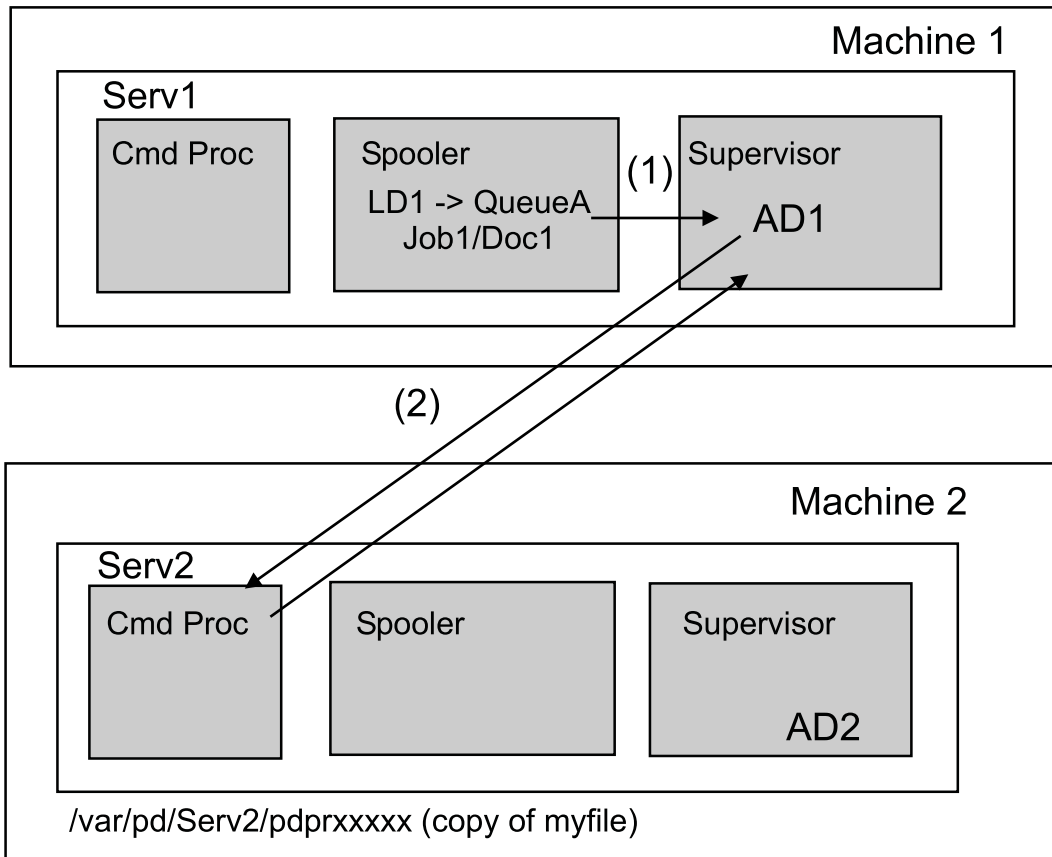
Files are "pulled" from one server to another using a socket. The command processor pdserver must be running for a file to be "pulled". When a file is "pulled", the actual destination stores it in its pdb directory. The "pulled" file is removed from the actual destination's server's pdb directory when the actual destination finishes processing the job, that is, it is completed or canceled.

Destination Support System (DSS) job processing

After the job (and print files) are received by the actual destination, how it is processed depends upon the type of actual destination. InfoPrint Manager supports several types of actual destinations. The files involved vary greatly depending upon the type of actual destination. The types are defined by the **destination-support-system** attribute of the actual destination and include:

BSD

A BSD DSS actual destination can be used to invoke any command you wish as long as it accepts a file as the last argument. The default command is **rembak**.

AIX

An AIX DSS actual destination drives printers like the AIX print-spooling subsystem. The AIX print-spooling subsystem uses backend programs that are integrated into the AIX print-spooling subsystem. Examples of these programs include **piobe**, **piojetd**, **lexbe**, **pioibmdpm**, **pioinfo**, **pioipps**, **piorpdm**, and **rembak**.

CUPS

A Linux CUPS DSS actual destination drives printers like the Linux CUPS print-spooling subsystem. The Linux print spooling system uses backend programs that are integrated into the Linux CUPS print-spooling system. Examples of these programs include **pioinfo** and **rembak**.

PSF

A PSF DSS actual destination uses Print Service Facility (PSF) to process the job. There are many flavors of PSF actual destinations. The flavor is defined by another actual destination attribute, **attachment-type**. The attachment types include:

Command

InfoPrint Manager communicates with this printer device by forwarding print requests to the remote print system or printer with a command.

Other-driver

InfoPrint Manager communicates directly with the printer device through an AIX or Linux printer backend program.

TCP/IP

InfoPrint Manager communicates with this TCP/IP LAN-attached printer device directly through the TCP/IP network.

Email

An Email DSS actual destination sends the data as an email.

DFE

A DFE DSS actual destination uses the Digital Front End (DFE) server to process the job.

The AIX, CUPS and BSD Destination Support Systems simply forward the print file to the defined destination. The PSF DSS, DFE DSS, or Email DSS provide many functions that might result in many more files being generated.

PSF, DFE, and Email print file handling

For actual destinations defined with either the PSF DSS, DFE DSS, or Email DSS, InfoPrint Manager does any transforms that are needed (see Transforms, p. 37). In the cases where the job is converted to AFP, InfoPrint Manager invokes Input Manager to segment the files into smaller pieces (called segments) and to create a segment list, which is a file that lists all the segments.

/var/psf/seglist and /var/psf/segments with Input Manager

For InfoPrint Manager actual destinations defined with the PSF DSS, InfoPrint Manager uses `/var/psf/seglist` and `/var/psf/segments` (by default) for storing all the AFP, including AFP resulting from the transform of all datatypes (or in InfoPrint Manager terminology, *document formats*), except ASCII (meaning Proprinter or Quietwriter ASCII, or flat ASCII that cannot be formatted with a page definition through ACIF).

The Input Manager segments incoming jobs once they have been transformed to AFP, or if they come in as AFP, to create smaller pieces. These pieces can then begin to print before a file is entirely

transformed, leading to faster *first-page-out* time. It is more of an advantage for files that need to be transformed (such as PostScript, PCL, PDF, and so forth) than for files that are already in AFPDS or MO: DCA format.

When Input Manager segments the file, it creates a segment list (a list of all the segments) in `/var/psf/seglist` and the segments themselves, which are by default, 5000K chunks of AFP, in `/var/psf/segments`. The default segment size of 5000K is probably acceptable for most environments. See PSF (/var/psf/segments), p. 41 for more information about segment size.

The first segment of a job contains **all** of the inline resources associated with the job. It is called `xxx.seg0`. If this segment does not exist for a job, the job does not have any inline resources. This first segment does not follow the segment size configured for the segmenter, that is, it is as large as it needs to be to contain all of the inline resources for the job.

All of the segments have the same name, with the final extension increasing numerically until it reaches the last segment with a name like `xxx.segx.end`. This segment tells PSF that it does not need to wait for more segments and that the job is finished.

After a job prints and is stacked, its segments are deleted unless the **delete-segment-list** job attribute is set to **false**. If segments have been saved or space is needed while processing a job, a cleanup routine deletes the segments that are no longer needed.

Segments are created one after the other and sent to the PSF DSS to improve the processing or printing speed. However, there is a limitation when submitting multiple-document jobs such as multiple AFP files in a single `pdpr` command or staging to print jobs via hot folders. In this case, the documents are concatenated and placed in the same stream, being seen as a single AFP print file (job).

If the first AFP file contains an inline resource group, then the first segment, `xxx.seg0` for the entire multi-document job, will contain this inline resource group and the inline resource groups from the rest of the documents (the other AFP files) will be discarded. Moreover, if the first document does not contain an inline resource group, the entire concatenated multi-document job will not have an inline resource group `xxx.seg0`. Therefore, it is strongly recommended to avoid using multi-document jobs having inline resource groups so as not to receive resource error messages.

Transforms

Different document formats use different transforms. Different transforms behave differently when making temporary *work* copies:

AFPDS

Since the file is already AFPDS, there are no transforms needed. When the job is scheduled to the actual destination, the Input Manager segments the AFP file, creating the segment list in `/var/psf/seglist` and the segments themselves in `/var/psf/segments`.

1403 line data

When a line data job to be formatted with a page definition is scheduled to the actual destination, ACIF does **not** make another temporary copy in a work area of its own; rather, ACIF works with Input Manager to build the seglist in `/var/psf/seglist` and the AFP segments directly into `/var/psf/segments`.

PostScript

The Postscript transform pipes the source file to its work area (by default, **/var/psf/ps2afp** or **/var/psf/ps2afp2**, which is user-customizable in the daemon configuration files). This means that

the Postscript transform can process files larger than its work area. The output of the Postscript transform is piped to Input Manager, which segments the AFP data, and creates segments in `/var/psf/segments,` with the segment list file in `/var/psf/seglist.`

PDF

The PDF transform is similar to the Postscript transform, except that the whole PDF file must be copied to the transform's work area (by default, **/var/psf/appe**, **/var/psf/ps2afp**, or **/var/psf/ps2afp2**, which is user-customizable in the daemon configuration files). This means that the PDF transform needs at least as much space in its working directory as the largest PDF file it will process, and that the copy must happen before the transform can begin. Like the Postscript transform, the output of the PDF transform is piped to Input Manager, which segments the AFP data, and creates segments in **/var/psf/segments**, with the segment list file in **/var/psf/seglist**.

Image tranforms (GIF, JPEG, and TIFF)

The Image transforms do not copy the file before transforming them to AFP, and pipe the output to Input Manager, which segments the AFP data, and creates segments in **/var/psf/segments**, with the segment list file in **/var/psf/seglist**.

ASCII

ASCII does not get transformed to AFP; it is converted directly to IPDS and sent to the printer. No segment lists or segments are created. ASCII includes Proprinter II or QuietWriter III ASCII, or flat ASCII that you are not formatting with a pagedef.

In addition, with the PSF other-driver DSS, the data is further transformed into PCL or PPDS and stored in a file in **/var/pd** or `$PDBASE` before being sent to the printer. This file is removed as soon as it prints.

Saving rip files

In some cases, InfoPrint Manager saves the ripped AFP output of a transform to the file **/var/pd/server/*jobidnumber.docnumber.printermodel*.afp**. This happens before the AFP is segmented by Input Manager.

This can happen if any of these are true:

- The **save-rip-files** server attribute is set to **true**.

- The **job-rip-action** document attribute is set to **rip-and-hold**, **rip-and-hold-ignore-ready**, **rip-only**, or **rip-only-ignore-ready**.

- The **transform-output-file-name** document attribute is set to any value.

In other cases, this AFP file is created for processing the job, and deleted after the job prints. These cases are:

- The **output-format** document attribute value is set to **booklet-print**, **booklet-print-with-integrated cover**, or **side-by-side-copies**.

- The **reverse-page-order** actual destination or document attribute is set to **true**.

- The **copy-count** document attribute is set to a value greater than 1 for rip-and-print jobs.

- The **page-range** document attribute is set to any value.

4. File systems

- InfoPrint Manager namespace (/var/pddir/default_cell)
- InfoPrint Manager server (/var/pd)
- PSF (/var/psf)
- PSF (/var/psf/segments)

The main file systems used by InfoPrint Manager include:

- `/var/pddir/default_cell` **(PDNAMESP)**
- `/var/pd` **(PDBASE)**
- `/var/psf`
- `/var/psf/segments`

InfoPrint Manager requires adequate space to store its data in these file systems. Each pdserver automatically monitors these file systems as well as `/var/spool/lpd`, and `/tmp`. When InfoPrint Manager detects a disk space low condition on any of these file systems, it logs a message in the error log and sends a notification if requested. See the chapter entitled "Monitoring memory and space on your system" in the *RICOH InfoPrint Manager for AIX: Procedures* or *RICOH InfoPrint Manager for Linux: Procedures* for more information about the InfoPrint Manager file systems monitoring and notification.

To plan for your disk space, you need to understand what is stored in each of these file systems.

InfoPrint Manager namespace (/var/pddir/default_cell)

InfoPrint Manager server (/var/pd)

Each pdserver has a directory where it stores its data. The PDBASE environment variable defines the parent directory for all the pdserver's directories. It is usually set to `/var/pd`, so you will most likely find a given server's data in the `/var/pd/ServerName` directory. The data includes command processor data, spooler data, supervisor data, and shared data.

Shared general data incudes error and trace logs for the pdserver although the spooler and supervisor parts make most of the error log entries; the command processor part rarely makes error log entries.

Command processor data includes:

- The local to global job id mapping table
- Copies of print files submitted through the command processor

Spooler/Supervisor data includes the print data base (pdb) for the spooler and supervisor parts of the pdserver where information is maintained about the objects in the pdserver, so the pdserver can be shut down and restarted without loss of those objects.

Supervisor only data includes copies of files pulled from command processor on other machines for processing and files generated by the transforms and actual destinations contained in this pdserver.

The main InfoPrint Manager server items to consider for file system space are:

- Print database
- Server logs
- Print files

- Files generated by InfoPrint Manager ripping and printing.

InfoPrint Manager server print database

InfoPrint Manager objects are persistent. Persistent objects are defined as objects whose existence spans the stopping and restarting of the process in which they reside. Persistent objects have the ability to save and restore their data to and from some persistent medium, such as disk. Each InfoPrint Manager pdserver has a Print Database (pdb) where it stores information about its objects.

The pdserver's Print Database also manages controlling the access to objects by different threads, and can swap objects out of memory to handle low-memory situations. When an object is created or changed, it is written to disk. If the object is not currently in memory and some thread wants to access it, it is brought into memory.

The server's Print Database information is stored in a subdirectory of the server's directory, for example, `/var/pd/ServerName/pdb`. The names of the directories in the **pdb** directory denote the type of object, and the names of the files in those subdirectories are the actual object names.

You should never access the server's pdb directory unless absolutely necessary. If you do, you shouldn't do it while a server is running. If a pdserver is killed or it crashes, it is possible for pdb files to get corrupted. If pdb files are corrupted, those objects cannot be restored or brought into memory when the pdserver is restarted. When this happens, you will not be able to access that object any more. The pdserver logs a message at startup for any objects that are corrupted. This alerts you that you have corrupted objects, and they need to be manually removed and recreated. Other than this situation, it is not a good practice to manually remove ojbects from the pdb while the pdserver is shutdown since there could be other associated data (like the namespace entry) that is not removed. You should always remove objects from InfoPrint Manager by using the **pddelete** command if at all possible.

Because information in `/var/pd` is so dynamic, it is not recommended that this file system be NFS mounted or shared across machines.

InfoPrint Manager server logs

The InfoPrint Manager server stores error, trace, and accounting logs in its directory.

The error and trace logs have attributes to control their size. They are backed up every time the pdserver is started. By default, only 10 backup error logs are retained and only two backup trace logs are retained, but you can change the number of logs to retain. See the chapter entitled "Customizing error logs in the InfoPrint Manager for AIX server" in the *RICOH InfoPrint Manager for AIX: Procedures* for more information. See the chapter entitled "Customizing error logs in the InfoPrint Manager for Linux server" in the *RICOH InfoPrint Manager for Linux: Procedures* for more information.

If you are using the InfoPrint Manager accounting logs, these logs are created in a subdirectory of the server's directory. The subdirectory is named `accounting.logs`. The InfoPrint Manager only adds to these logs. You must manually prune or archive the accounting logs. See the chapter entitled "Gathering accounting and audit data about print jobs" in the *RICOH InfoPrint Manager for AIX: Procedures* or *RICOH InfoPrint Manager for Linux: Procedures* for more information.

InfoPrint Manager server print files

As described in Print jobs and print files, p. 31, InfoPrint Manager uses the `/var/pd` filesystem on the "originating server," which is the pdserver used as the command processor, to create and store temporary copies as long as the jobs remain in the InfoPrint Manager system.

The command processor pdserver that is used to process a **pdpr** request makes a copy of each print file in the job and stores it in its directory with a name that starts with `pdpr`. The pdserver containing the logical destination for the job can be a different pdserver. And the actual destination that will process the job can be in a third pdserver. This means that the pdserver with the copy of the print file can be different than the pdserver that contains the job since the job is created on the pdserver containing the target logical destination. The location of the print file and the name of the pdserver that is managing it is stored in the **document-content** attribute of each document object in the job. When the job is scheduled to an actual destination, the actual destination's pdserver will see if it can access the file directly. If not, it will contact the command processor pdserver that was used for the job and request that the print file be sent. When a print file is sent to a supervisor pdserver, the file is stored in the server's **pdb** subdirectory. As soon as the actual destination finishes processing the job, this copy is deleted. The command processor's copy is not deleted until the job is deleted from the spooler pdserver. If a job is retained, the print file is retained.

The printing pdserver or supervisor part of the pdserver also stores saved rip files when requested to save rip files in `/var/pd`.

All environments must plan spooling space for one copy of each print file. The command processor's copy of the print files are not deleted until the job leaves the InfoPrint Manager system. In other words, until it is completed or canceled and the retention-period has expired. Multiple machine environments must also plan additional space for "pulled" files. Since these files only exist while a job is processing, less space is required.

See Temporary files created by InfoPrint Manager, p. 99 for more information.

PSF (/var/psf)

Each machine has a directory where it stores the data for PSF. This data is used by the PSF DSS actual destinations in all the pdservers running on that machine. There are several directories under `/var/psf` when a system is initially installed. Each of the *xxx2afp* transforms has a directory under `/var/psf` (**apped**, **gif2afp**, **jpeg2afp**, **pcl2afp**, **pdf2afp**, **ps2afp**, and **tiff2afp**). In addition to these, there are the **/var/psf/segments** and **/var/psf/seglist** directories. Each time you create a PSF DSS printer, another directory with the name of the actual destination is added under `/var/psf`. The printer directories contain the printer profile and a file identifying the pdserver associated with the printer, the printer error log, the trace log, and several printer status files and printer information files. All of the files, except for the profile and its server file, are created by PSF when it is running. So, if you look in the `/var/psf/printername` directory just after creating a PSF DSS actual destination, you will see only the printer profile and its server file. After PSF has tried to start (successfully or unsuccessfully), you will see many more files in the printer directory.

PSF (/var/psf/segments)

As described in Print jobs and print files, p. 31, PSF Input Manager segments AFP files so printing can start before a job completes ripping. The smaller the segments, the faster PSF starts printing the job.

However, if the pieces get too small, there will be too many files and too much disk thrashing. If PSF has to open a new file for each page that can slow things down significantly.

The default setting of 5000KB for each segment can be raised to reduce the disk I/O 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.

The InfoPrint Manager SMIT **Customize the Input Manager** panel (`InfoPrint Manager→ Miscellaneous→Customize the Input Manager`) lets you change the default segment size. The segment size parameter determines the maximum size of any temporary file created by the Input Manager component of InfoPrint Manager that is stored in `/var/psf/segments`. This parameter should only be adjusted when there is a problem with throughput.

Note

The **segment-file-size** document attribute can be used to change the segment size for a specific print job.

Since there are so many small files stored in `/var/psf/segments`, you might think you should change the fragment size of the file system to allow for better space utilization. However, for better performance, it is best to set the fragmentation size at its highest value of 4096 bytes because a smaller size might increase disk I/O operations.

By default, segments are deleted when they are no longer needed by PSF. You can specify **no** with the **delete-segment-list** job attribute to retain the segments. You might want to do this if you are reprinting the job. When segments are retained, Input Manager automatically manages the space in `/var/psf/segments`. By default, fifty percent of the space will be allotted for retained segments. The **Customize the Input Manager** SMIT panel can also be used to change this value. A higher value will cause Input Manager to retain more print data in the system to extend the range of re-start and rerun job management functions. However, a higher value might cause printer clutching since there will be less room for the printing jobs segments.

5. InfoPrint Manager server memory

- AIX virtual memory
- Linux virtual memory
- What requires memory
- The pdserver's virtual memory management

AIX virtual memory

Whenever a process gets virtual memory from the operating system, it can not return it. It can re-use the memory for other purposes but the size of the memory allocated to that process never decreases. A process is not allowed to exceed its memory allotment.

A process's memory allotment, the actual amount of virtual memory it can be allocated, is defined by the **ulimit -d** (data) value. A machine-wide **ulimit -d** setting is defined in the `/etc/security/limits` file and the default is 256MB. The **ulimit -d** value can be changed either machine-wide by changing the `/etc/security/limits` file or for a specific process by using the **ulimit -d** command. Sufficient authorization is required to change the **ulimit**.

The **ulimit -d** value can be raised to unlimited. A value of unlimited leaves the maximum memory a process can obtain up to the process's memory model.

The InfoPrint Manager pdserver manipulates large amounts of data so it is built to exploit the Large Program support. Each pdserver can be allotted up to four 256 MB segments by the O/S. Therefore, if the **ulimit -d** value is set to unlimited, the pdserver can obtain up to 2 GB of virtual memory.

Linux virtual memory

The amount of memory that an InfoPrint Manager server instance can use is determined by various factors, such as **ulimit -v** (total virtual memory) and **ulimit -d** (data segment size).

A process's memory allotment, the actual amount of virtual memory that can be used, is defined by the **ulimit -v** (total virtual memory) value. A machine-wide **ulimit -v** setting is defined in the `/etc/security/limits.conf` file. The **ulimit -v** value can be changed either machine-wide by changing the `/etc/security/limits.conf` file or or by using the **ulimit -v** command for a specific process. Sufficient authorization is required to change the **ulimit**. The **ulimit -d** setting affects a part of the virtual memory (data segment size) of the InfoPrint Manager server. This is usually unlimited by default.

Unlimited values for these settings leaves the maximum memory a process can obtain from the operating system.

To make sure changes to ulimit parameters persist, record the changes in: `/etc/security/limits.conf`.

What requires memory

The pdserver needs memory for many purposes. It is probably obvious that memory is required for all the objects in the pdserver. The more objects, the more memory required. However, the pdserver needs memory not only for the objects it contains, but also for the functions it does related to those objects. InfoPrint Manager is rich in functions and allows many alternatives on how to configure and use the system. Some of these configuration decisions require more memory than others.

Some examples of functions that require memory include queued notifications for both end users and GUIs, threads for scheduling jobs, threads for processing jobs, threads for communicating with printers, alarms for handling when to delete retained jobs, scheduler verification records for queued jobs, and memory to handle collecting the information to return on queries of objects. This is not intended to be a

complete list; that would be extremely difficult to generate. Suffice it to say, any given function is going to require memory to be performed. The required memory for a given function depends on many variables. Some examples include:

- A scheduler verification record is needed for every actual destination for every job in a queue. So if you have 10 jobs in a queue with 10 actual destinations, there would be 100 verification records. However, if you have the 10 jobs split between 2 queues, each queue having 5 actual destinations, there would be only 50 verification records.

- By default, there is a scheduler thread for every scheduling algorithm used by a queue, but if you change the queue's assign-to-destination attribute value to true, there is a scheduler thread for each actual destination on the queue.

- There is a thread for every processing job. If you increase the **maximum-concurrent-jobs** from 5 to 10, and you have 100 actual destinations, the number of potential processing job threads increases from 500 to 1000.

- If a query for all the jobs in the pdserver is received, the pdserver needs to make a copy of the data to return on the query. The amount of memory required for the data grows as the number of jobs grows.

Many factors affect the memory required for a pdserver. The memory required is not linear to the number of objects in the pdserver.

The pdserver's virtual memory management

Much of the memory used by a pdserver is transient, that is, it is used for a purpose, freed, and available for use again. However, the pdserver tries to maintain all its objects in memory to provide the best performance.

Once a pdserver has allocated 70% of its total virtual memory allotment, it starts to manage its existing memory before requesting more memory. When a pdserver reaches 90% of its total virtual memory, it automatically shutdowns to prevent object corruption that could be caused by a crash from running out of memory.

This mechanism is not enabled by default, because this behavior lowers the performance too early. If you want to enable this mechanism, export the **PD_USEMEMCLEANUP** environment variable with a value of **yes** before starting InfoPrint Manager server.

After reaching 70% of its total virtual memory, a pdserver tries to re-use its existing allocated memory by swapping objects out of memory to disk before requesting more memory. The process of swapping objects out of memory can be time consuming. A least-recently-used algorithm is used to swap objects out of memory. Depending on the situation, a lot of memory can be recovered and your server can run for a long time on that recovered memory. However, any objects swapped out of memory will be brought back into memory if you access them again, for example by issuing a `pdls -c job serverName:`. If the server doesn't have objects to swap out that it doesn't need to access, it is not going to benefit from the memory recovery process.

The ability to recover memory for re-use is related to object management, so the memory is only checked when objects are being accessed in the pdserver. Functions in the pdserver that do not access objects have no ability to recover memory for re-use. These functions do not leak memory, but if they need more memory they request it from the O/S without swapping objects out of memory in an attempt to re-use previously allocated memory.

You are definitely not getting the best performance possible if you have a pdserver in object swapping mode and should avoid this situation. To alert you to this so you can correct the problem, the pdserver

logs messages regarding this situation in its error.log. It also sends notification messages to the GUI for display in the error message window and to those identified on the server's notification-profile attribute. By default, the pdserver's notification-profile is set to send both `low-on-memory` and `memory-exhausted notification` messages. The messages sent for low-on-memory events are `5010-642 InfoPrint Manager server _____ detected a memory shortage and is attempting to recover memory` and `5010-648 The InfoPrint Manager server ___ is using ___ percent of its available memory`. The message sent for memory-exhausted is `5010-647 No memory available for the InfoPrint Manager server ___. Attempting shut down`.

⏬ **Note**

1. If you don't like the server's default delivery-address and delivery-method for these notifications, you can change them. You can also have the notifications sent to multiple delivery-addressses.
2. If your pdserver's notification-profile existed before these events were added to the product, you might have to add them to your server's notification-profile, or you can reset the notification-profile to the default, and they should appear.

If you are encountering a low memory situation, you can correct the problem by increasing the pdservers allotment of memory or by adding more pdservers and moving some of the objects and processing to them. You might also have old objects, such as retained jobs, you do not really need that could be deleted.

The **PD_THRMEM_THREADSHOLD** environment variable can control how much memory is used for permanent objects (queues, actual destinations) and transient objects (jobs). A higher value for the **PD_THRMEM_THREADSHOLD** environment variable is used for permanent objects, while a lower value is used for transient objects. For more information about this environment value, see *RICOH InfoPrint Manager: Reference*.

5

5

6. InfoPrint Manager server performance tuning

- Hardware configuration
- General considerations
- Print throughput performance

Hardware configuration

The InfoPrint Manager performance data depicted in this chapter is based on the following system configuration:

Model	P-series 55A (AIX)	X366 (Linux)
Number of physical CPUs	4	4
CPU Type/Speed	Power 2.5 GHz each	Intel Dual Core 2.5 GHz
RAM Memory	8 GB	8 GB
Disk number and capacity	4 HDD at 36 GB	6 HDD at 34 GB each
Ethernet card	100 BaseT	Gigabit
OS Level	AIX 7.1 TL 09	Red Hat Enterprise Linux 6.3 – 64 bit

General considerations

The current chapter contains the results of InfoPrint Manager performance testing. The tests meant to determine how the system reacts in terms of responsiveness and stability under a particular workload.

You must consider the requirements for an InfoPrint Manager system as a total entity. Remember that the product is dependent on the hardware configuration to process the workload accordingly.

Consider these areas:

- Processor capacity
- Application requirements
- Anticipated growth requirements
- I/O slot requirements (HDD, network, bus, etc.)
- System memory

The application related factors that affect performance are:

- Number of jobs
- Average job size
- Number of pdservers
- Number of queues
- Number of actual destinations that are printing jobs at any given moment
- Actual destinations types (PSF, AIX, CUPS)

The following graphic is valid for any InfoPrint Manager usage type:

The memory usage related to the number of jobs on AIX
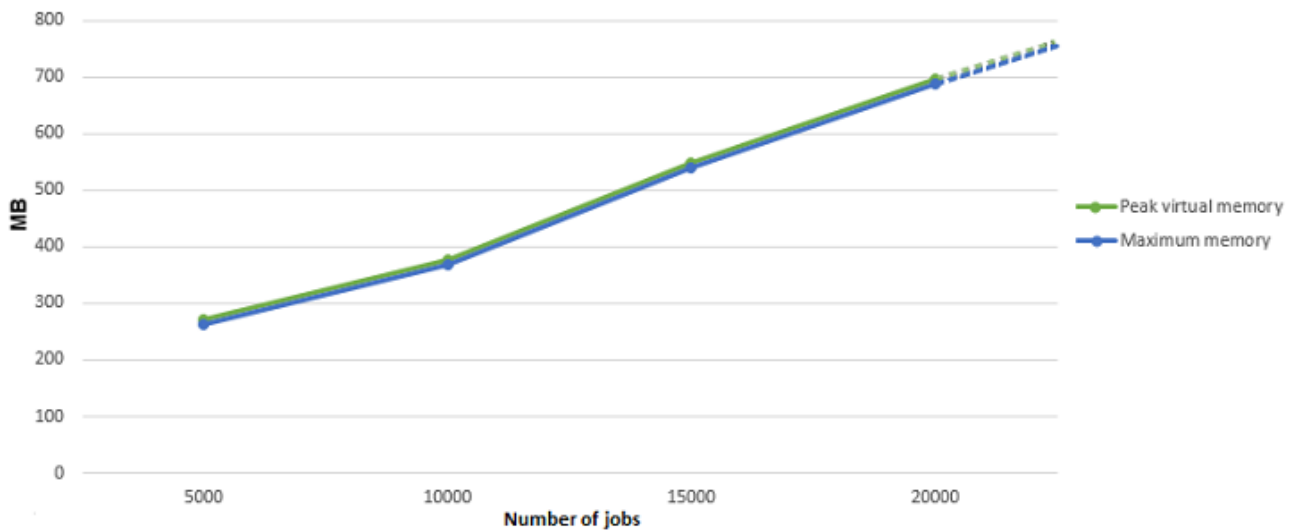
The memory usage related to the number of jobs on Linux
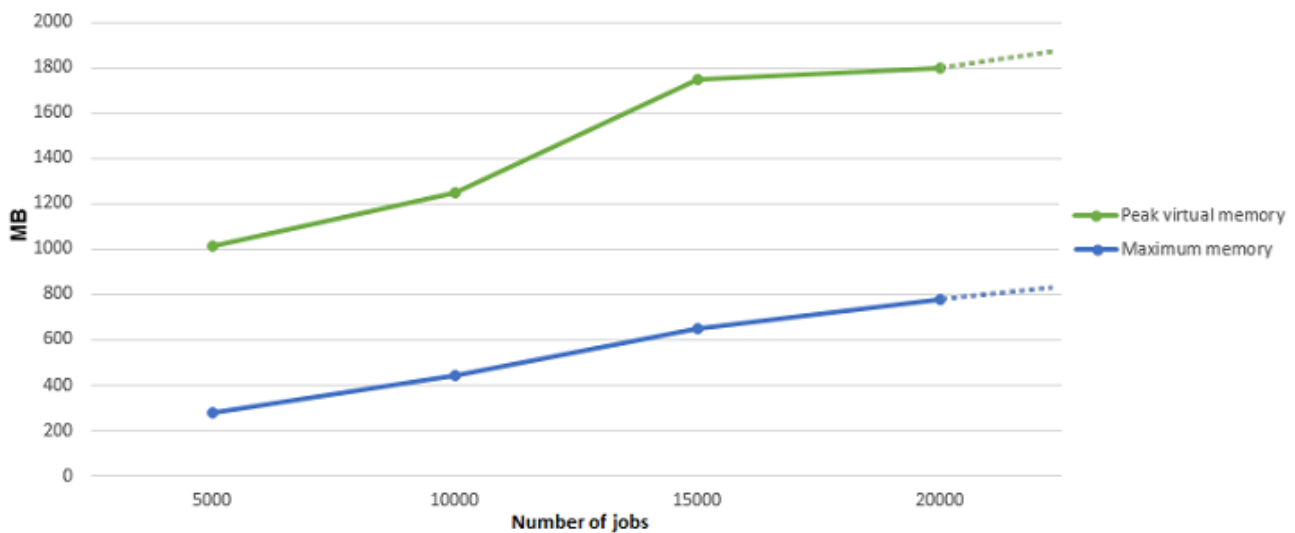
⭐ Important

- For best results on AIX, the maximum recommended memory size for the InfoPrint Manager server must not be higher than 1400 MB. For more then 1400 MB, a new pdserver is needed.

- For best results on Linux, the maximum recommended virtual memory size for the InfoPrint Manager server must not be higher than 3200 MB. For more then 3200 MB, a new pdserver is needed.

- Consider adding a new pdserver on the same system to balance the workload only if the hardware configuration allows. For more informations about the hardware configuration, see AIX operating system and hardware tuning, p. 55 or Linux operating system and hardware tuning, p. 67, depending on the operating system.

- The memory used by the pdserver directly depends to the number of jobs, actual destinations, and queues. For informations about the virtual memory, see InfoPrint Manager server memory, p. 43.

Note

The destination configuration model, referenced in the *Planning for configuration* chapter of *RICOH InfoPrint Manager for AIX: Planning Guide* and *RICOH InfoPrint Manager for Linux: Planning Guide*, does not significantly affect the memory usage and the print throughput.

Print throughput performance

The scope of this section is to help determine an optimum number of objects that the InfoPrint Manager server can handle, to avoid network/disk bottlenecks, large memory increases, or high CPU usage.

A single InfoPrint Manager server instance was tested in two configurations:

1. Using high speed continuous forms printers, optimizing InfoPrint Manager to give a constant high print throughput using AFP documents for printing.
2. Using cut sheet printers, optimizing InfoPrint Manager to support SAP print and PSF-Other/Passthru printing.

High and Ultra High speed printing

The high speed printing assumes the use of high speed printers that usually print large jobs on a low number of devices.

Configuration:

- Only PSF TCPIP Actual Destinations were used.
- Each Actual Destination was assigned to a different printer.
- The same job was sent to disabled Actual Destinations.
- All Actual Destinations were enabled at once.

The print speed related to the number of Actual Destinations for 2000 pages jobs
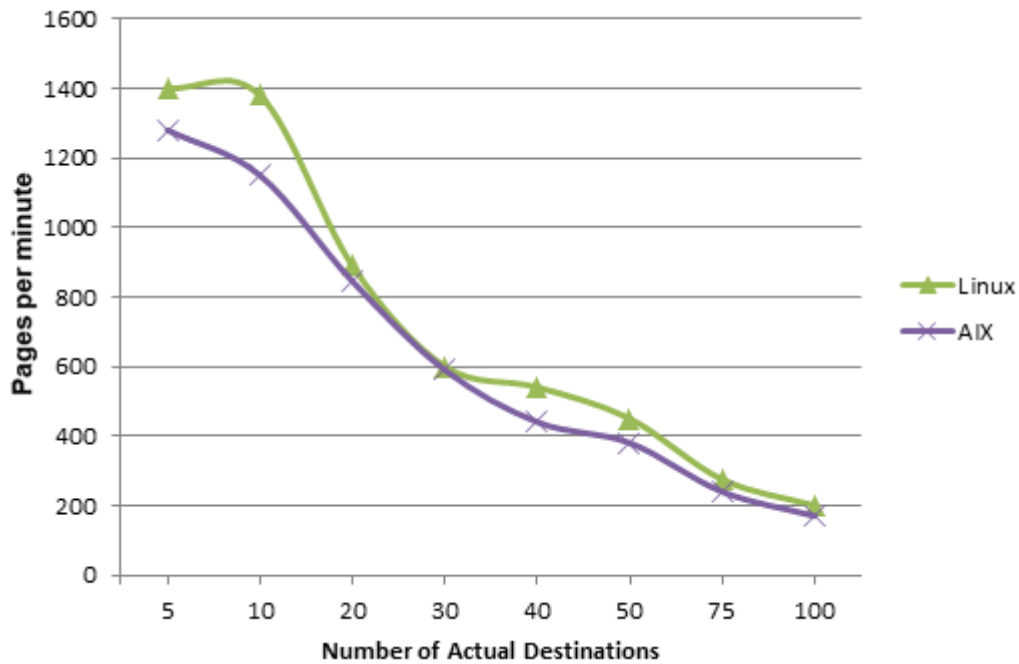
The print speed related to the number of Actual Destination for 11000 pages jobs
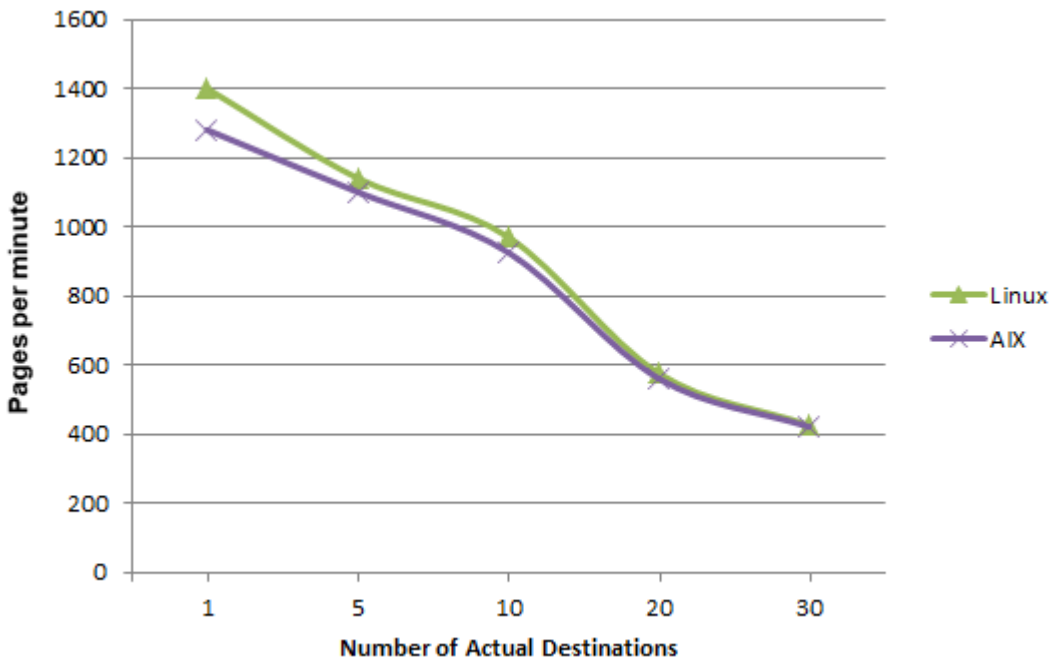
⭐ Important

- If more than 20 printers are printing at the same time and are managed by the same pdserver, the average print throughput decreases on the systems under 1000 pages per minute.
- The optimal results are obtained when `maximum-concurrent-jobs` attribute is set to 1.

⬇Note

The `maximum-concurrent-jobs` attribute value must be set up according to the average job size:

* Low value for large jobs
* High value for small jobs

Low, Workgroup, and Medium speed printing

This type of printing assumes the use of hundreds of cut sheet printers.

Configuration:

* 300 Actual destinations were created.
* Only PSF-Other or AIX/CUPS Actual Destinations were used.
* Each Actual Destination was assigned to a different printer.
* The same job was sent to disabled Actual Destinations.
* All Actual Destinations were enabled at once.
* The destination configuration model used was desktop destination configuration model. For more information about the destination configuration models, see *Planning for configuration* chapter of *RICOH InfoPrint Manager for AIX: Planning Guide* and *RICOH InfoPrint Manager for Linux: Planning Guide*.

In practice, the most used destinations are PSF-Other destinations (SAP->AFP->PCL) and AIX/CUPS destinations (PCL/PS->Printer).

The following charts show the memory usage when a pdserver is managing increasing numbers of Actual Destinations, each printing 18 jobs.

The memory usage of PSF-Other and AIX DSS printing systems related to the number of jobs on AIX
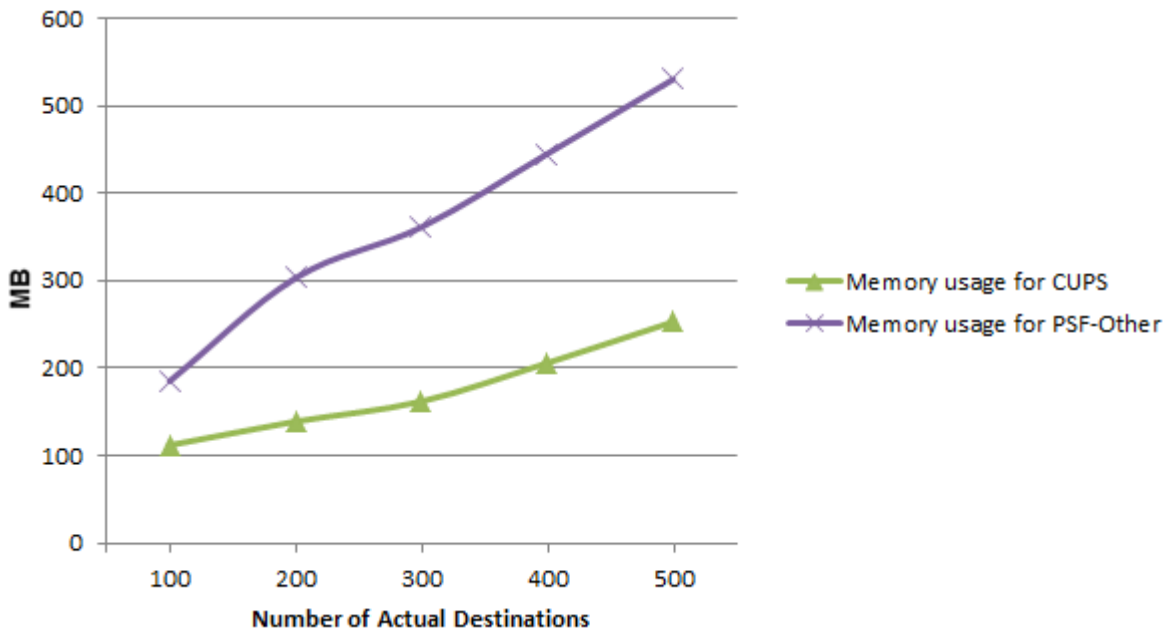
The memory usage of PSF-Other and AIX DSS printing systems related to the number of jobs on Linux
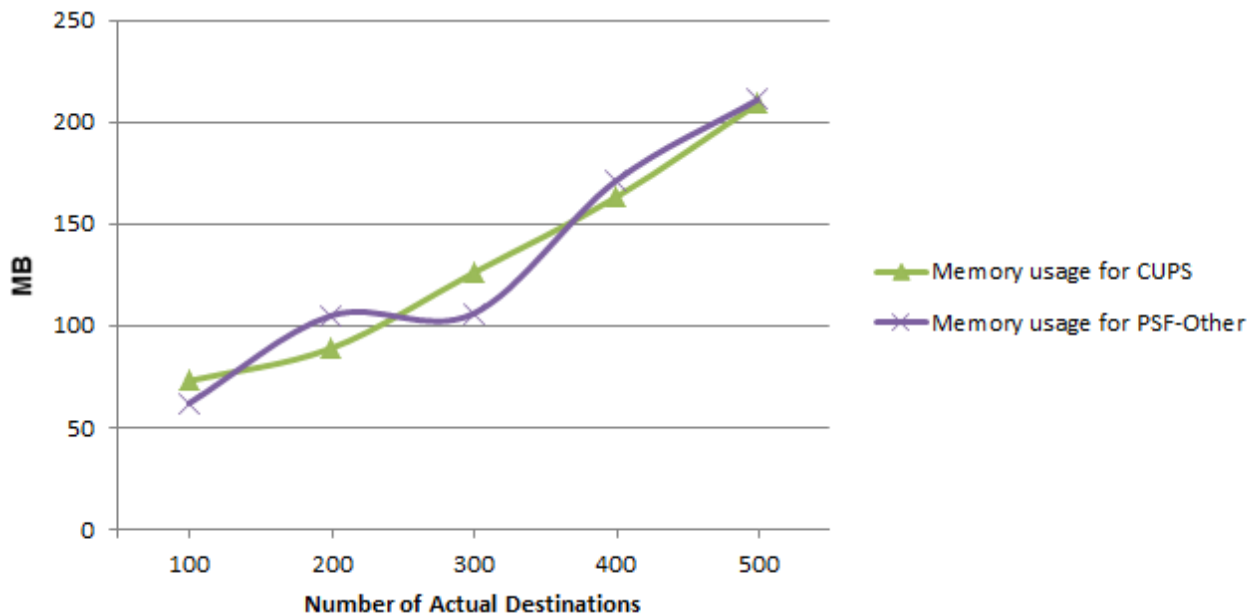
The memory usage in relation to the number of jobs for 300 PSF-Other Actual Destinations
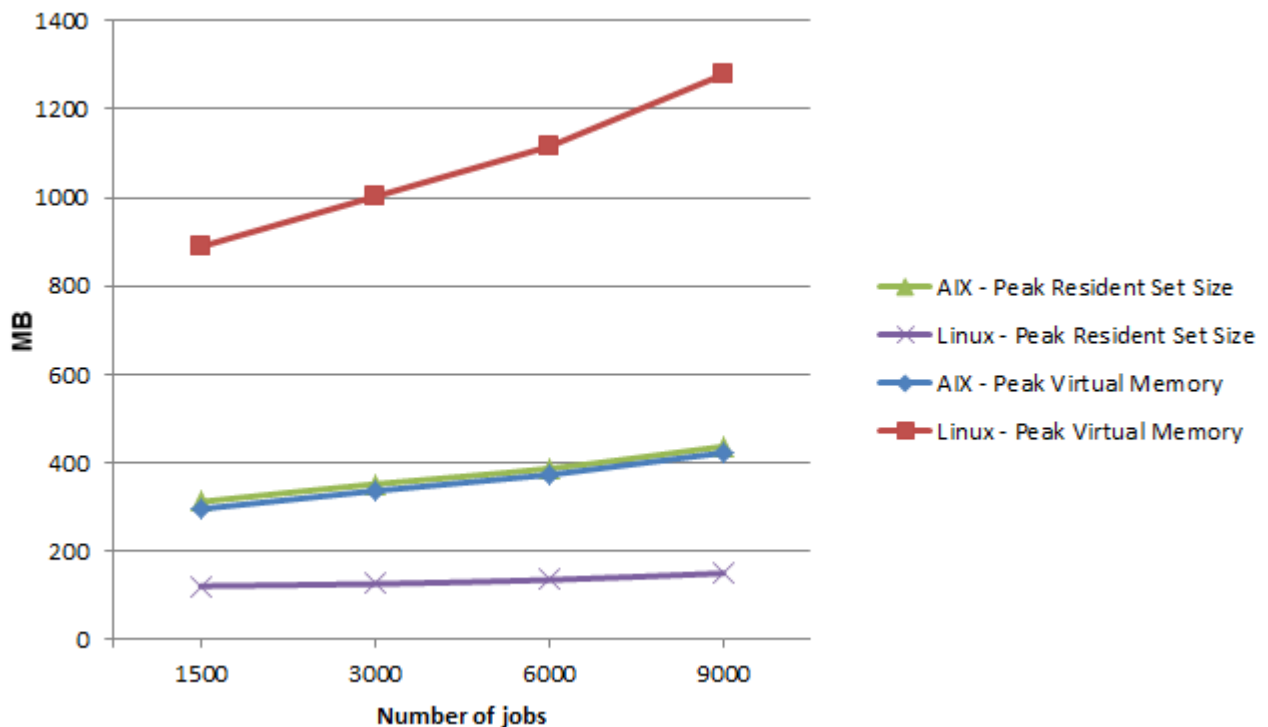
Note

- For optimal results, the maximum recommended number of Actual Destinations must not be higher than 300. For more then 300 Actual Destinations, a new pdserver is needed.

- Consider adding a new pdserver to balance the workload only if the hardware configuration allows. For more informations about the hardware configuration, see AIX operating system and hardware tuning, p. 55 or Linux operating system and hardware tuning, p. 67, depending on the operating system.

- The memory used by the pdserver directly depends to the number of jobs, actual destinations, and queues. The memory used by the pdserver directly depends to the number of jobs, actual destinations, and queues. For informations about the virtual memory, see InfoPrint Manager server memory, p. 43.

- Many factors affect the memory required for a pdserver. The memory required is not linear to the number of objects in the pdserver

7. AIX operating system and hardware tuning

- How much memory is needed?
- How much paging space is needed?
- What affects CPU utilization?
- Network bandwidth
- Network options, AIX settings, and environment variables that affect performance
- Setting process priority
- Setting resource limits for InfoPrint Manager services and processes
- Disk/File systems

When working on the configuration for your operating system and hardware, start with the operating system recommendations.

The **sar** command can be used as a general command to look at system statistics. You can setup the **sar** command to run at specific intervals to collect system statistics for a specific amount of time. You can also use the **sar** command to schedule sytem evaluations at specific intervals, such as every day, every week, every month, and so forth.

How much memory is needed?

No exact number can be given as the recommendation for the amount of memory needed for a system running InfoPrint Manager. The amount of memory depends on how many jobs are going through the system at what frequency, the size, complexity, and data type of the jobs, and what else is running on the system.

With InfoPrint Manager, the more memory the better.InfoPrint Manager performs best when it has enough memory that it does not need to use paging space. That is, the more paging space it must use, the worse the performance.

What defines a memory bound system? If the number of active virtual memory pages exceed the amount of real memory pages, page space is used for those pages that cannot be kept in real memory.

To determine if InfoPrint Manager for AIX is using paging space, use the **vmstat** or **topas** command. With **vmstat**, look at the paging columns, such as **re**, **pi**, **po**, **fr**, **sr**, and **cy**. On a system with no paging, these columns contain zeros. On a machine with memory constraints, you see the free memory frames (**fre**) go toward zero, and then paging begins.

To start the **vmstat** command, enter:

```
vmstat interval  length
```

To tune the memory load control mechanism of AIX, use the **schedo** command. The **schedo** command is part of the `bos.perf.tune` fileset, and it can be found in the `/usr/sbin` directory. The **schedo** command lets you tune the load control mechanism.

For information and recommendations for changing values for your configuration, see:

- *AIX Version 7.1 Performance management*, which is available at https://www.ibm.com/docs/en/ssw_aix_71/performance/performance_pdf.pdf.

- the IBM AIX documentation website, which is available at https://www.ibm.com/docs/en/aix.

How much paging space is needed?

You can use these commands to determine how much paging space is being used on your system:

lsps

Gives a snapshot of the system at the time the command is issued.

topas

Shows how page space changes over the life of a process while a program is running. This command has a column marked `PAGING SPACE`.

svmon

Shows memory and paging space in use when you specify `-G` *`interval`*. The values are given in page frames. Each page frame is 4096 bytes. You can enter the **svmon** command once or with a time interval to monitor how paging and memory are changing.

⬇Note

While the **vmstat** comand shows paging activity, it does not show clearly how much paging space is being used.

To determine how much paging space is needed, it is recommended that the paging space should be double the amount of physical memory. So if a system has 1 gigabyte of memory, the total paging space should be 2 gigabytes (although the 2 gigabytegs can be separated onto more than one disk).

On AIX you should consider the following recommendations when configuring your paging space:

- Use the disk or disks that are least utilized.

- Do not allocate more than one paging page logical volume for each physical disk.

- Avoid sharing the same disk with logical volumes.

- Make all paging spaces the same size if possible.

To make paging space available to your system, you must add and activate the paging space. The total amount of paging space is often determined by trial and error. One commonly used guideline is to double the RAM size and use that figure as a paging space target. To use the Web-based System Manager wizard, click the Volumes container, then click the Paging Space container. From the **Selected** menu, select **Increase Page Space**→**Wizard**.

If you prefer to use `SMIT`, type one of these paths on the command line:

- To list your current paging space, type: `smit lsps`

- To add paging space, type: `smit mkps`

- To activate paging space, type `smit swapon`

When is a system thrashing? When page space becomes depleted, the operating system attempts to release resources by first warning processes to release paging space and then by killing the processes. Use the **vmtune** command to set the thresholds at which this activity occurs. **vmtune** is part of the `bos.adt.smples` fileset set, and it can be found in the `/usr/samples/kernel` directory.

For information and recommendations for changing values for your configuration, see:

- *AIX Version 7.1 Performance management*, which is available at https://www.ibm.com/docs/en/ssw_aix_71/performance/performance_pdf.pdf.

- the IBM AIX documentation website, which is available at https://www.ibm.com/docs/en/aix.

What affects CPU utilization?

You can measure CPU utilization with the **iostat** or the **vmstat** tool. The tools can produce a snapshot of the CPU utilization on a system and can be run at regular time intervals to show usage over time. Both tools separate CPU utilization in these categories:

%user

Percentage of CPU utilization that occurred while executing at the user level.

%system

Percentage of CPU utilization that occurred while executing at the system or kernal level.

%idle

The percentage of time that the CPU was idle and the system was idle, and the system did not have an outstanding I/O request.

%wait

The percentage of time that the CPU was idle during which the system had an outstanding I/O request.

When is a CPU bound? A CPU is bound if the sum of the user and system time exceeds 90% of the CPU resource on a single-user system or 80% of the CPU resource on a multi-user system.

With the **vmstat** tool or by looking at the CPU columns, look at the size of the run queue (`kthr - r` column). In general, as the run queue increases, users will notice an increase in response time, and therefore, a decrease in performance.

A high **%wait** with **vmstat** or **%iowait** with **iostat** indicates potential I/O bottlenecks. It could be that the system is short of memory, so the disks containing paging space might be busy while paging, and you are likely to see a higher run queue as threads are waiting for the CPU.

Network bandwidth

To measure network activity, use the **netstat** command. The **netstat** command can:

- Display active socket connections for each protocol
- Display routing table information
- Display contents of a network data structure
- Display packet counts throughout the communications subsystem
- Display network buffer statistics
- Display data link provider interface statistics
- Reset statistics

For information and recommendations using the **netstat** command, consult your operating system manual.

Network options, AIX settings, and environment variables that affect performance

You use the network options command, **no**, to set the network attributes. To print a list of all configurable attributes and their current values, specify **no -a**. This can be used to tune an AIX machine for different network attachments.

For information about the **no** command, see:

- *AIX Version 7.1 Performance management*, which is available at https://www.ibm.com/docs/en/ssw_aix_71/performance/performance_pdf.pdf.

- the IBM AIX documentation website, which is available at https://www.ibm.com/docs/en/aix.

For information about the semaphores, see *Interprocess communication limits*, which is available at https://www.ibm.com/docs/en/aix/6.1?topic=aix-older-versions.

Use the **maxmbuf** command to limit the amount of memory used by the communication subsystem. If **maxmbuf** is set to **0**, the value of **thewall (no -o thewall)** determines the maximum amount of memory to be used by the communication subsystem.

Setting process priority

Use the **nice** command to change the value of the priority of a thread by changing the nice values of its process that is used to determine the overall priority of the thread. A child process inherits the nice value from the parent process. You can view the nice value with the **ps -l** command.

The **schedo** parameters can also affect the priority of the thread. You use the **schedo** flag to decay the value of a thread, and you use the **schedo -r** flag to set a penalty factor on the thread.

The **schedo** command should only be used by someone who is proficient with the AIX operating system or under the guidance of an AIX professional. By changing some of the values through the **schedo** command, a system can become unusable.

For information about setting threads, see:

- *AIX Version 7.1 Performance management*, which is available at https://www.ibm.com/docs/en/ssw_aix_71/performance/performance_pdf.pdf.

- the IBM AIX documentation website, which is available at https://www.ibm.com/docs/en/aix.

Setting resource limits for InfoPrint Manager services and processes

The **ulimit** command sets or reports process resource limits as defined in the `/etc/security/limits` file. This file contains these default limits:

```
fsize = 2097151
core = 2097151
cpu = -1
data = 262144
rss = 65536
stack = 65536
nofiles = 2000
```

📥 Note

A value of **-1** sets a resource to unlimited.

Where:

fsize

The largest file a process can create or extend.

core

The largest core file a process can create.

cpu

The largest amount of system unit time (in seconds) a process can use.

data

The largest process data segment for a process.

rss

The maximum amount of physical memory (resident set size) a process can allocate.

stack

The maximum size, in bytes, of the stack region for a process. When the stack limit is reached, the process receives a `SIGSEGV` signal.

nofiles

The number of file descriptors a process can have open at one time.

If you are running InfoPrint Manager using a non-root user, for example **ipm1**, update `/etc/security/limits` file with:

```
ipm1:
nofiles = 8192
```

Disk/File systems

The main InfoPrint Manager file systems are:
```
/var/pd
/var/psf
/var/psf/segments
```

I/O performance is usually the limiting factor for an InfoPrint Manager production server. Therefore, spreading the InfoPrint Manager file systems across three or more drives is likely to boost performance for an InfoPrint Manager production server. Even more of a performance gain can be accomplished by putting `/var/pd` on one drive, `/var/psf` on another drive, and `/var/psf/segments` across two more drives, striping the two `/var/psf/segments` drives for better I/O performance. The first file system that should be spread across two or more striped drives is `/var/psf/segments`. You could also spread `/var/pd` across two or more striped drives.

One of the settings that can be made when creating file systems is the fragment size. Changing the fragment size allows for better space utilization. For better performance, it is best to set the fragment size at its highest value of 4096 bytes even if this might cause wasted disk space. With a smaller fragment size, there might be an increase in disk I/O operations.

Manual InfoPrint Manager file system allocation

The InfoPrint Manager for AIX installation software is optimized for a pSeries server with five SSA disks. When the installer senses the presence of the required drives, it creates the InfoPrint Manager file systems to take the best advantage of them.

However, the majority of InfoPrint Manager for AIX installations are done on servers with fewer drives, and non-SSA disks. In these cases, the installation software attempts to load the complete application on a single disk in `rootvg`. Because of InfoPrint Manager's workflow and file duplication characteristics, this can lead to an inefficient print server. To over-ride the installer's default behavior, you should pre-allocate file systems for InfoPrint Manager before starting the install process.

The uses a sample server with four SCSI 73.4GB disk drives. The creation steps and sizes for file systems are based on the assumption that the server is being used only as an InfoPrint Manager print server. The configuration values shown are samples only and will vary based on server type, number and size of disks, and so forth. No considerations have been made for any other applications running in this environment. In the sample, `rootvg` is on `hdisk0`.

File system creation procedure

1. **Create volume groups**

   Two volume groups, `pdvg` and `segvg`, are created to contain the `/var/pd` and `/var/psf/segments` file systems. Creating separate volume groups helps spread I/O across multiple actuators, which can improve performance. The `pdvg` volume group contains the important InfoPrint Manager database objects and should be backed up at regular intervals. The `segvg` volume group contains the segmented print files, and can be considered to be transient data. If you have multiple hard drives available for use with `segvg`, striping across these disks provides multiple heads accessing the data for faster I/O performance, which is useful when driving high speed/high resolution printers.

   ↓Note

   A 128 MB Physical Partition size is recommended for 73.4GB drives. If your server has larger or smaller drives, you should adjust this value proportionally.

   1. **Create `pdvg` volume group**:

```
smitty mkvg
VOLUME GROUP name[pdvg]
Physical partition SIZE in megabytes[128]
PHYSICAL VOLUME names[hdisk1]
Activate volume group AUTOMATICALLY[yes]
 at system restart?
Volume Group MAJOR NUMBER[]
Create VG Concurrent Capable?[no]
Auto-varyon in Concurrent Mode[no]
LTG Size in kbytes[256]
```

   Commandline:

```
mkvg -f -y'pdvg' -s'128' hdisk1
```

   2. **Create `segvg` volume group**

```
smitty mkvg
VOLUME GROUP name[segvg]
Physical partition SIZE in megabytes[128]
PHYSICAL VOLUME names[hdisk2 hdisk3]
Activate volume group AUTOMATICALLY[yes]
 at system restart?
Volume Group MAJOR NUMBER[]
Create VG concurrent capable?[no]
Auto-varyon in Concurrent Mode[no]
LTG size in kbytes[128]
```

Commandline:

```
mkvg -f -y'segvg' -s'128' hdisk2 hdisk3
```

🔽 Note

You need to divide the drives your server has available between your Volume Groups.

2. **Create logical volumes**

The logical volumes are created to contain the file systems. Since **/var/psf/segments** is a third level directory, **/var/psf** must be created first, so that it is mounted before **/var/psf/segments**. If for some reason you create them in the wrong order, you will need to edit **/etc/filesystems** to correct the order. Try to avoid this.

To calculate the number of **LOGICAL PARTITIONS**, issue **lsvg pdvg** to get the number of free physical partitions available on the hard drive. Use a number roughly 30 less than the number of free physical partitions to fill the entire drive with the volume group.

1. **Create lvpd logical volume**

```
smitty mklv
Volume Group name
[pdvg]

Logical volume NAME[lvpd]
VOLUME GROUP name[pdvg]
Number of LOGICAL PARTITIONS[512]
PHYSICAL VOLUME names[hdisk1]
Logical volume TYPE[jfs]
POSITION on physical volume    middle
RANGE of physical volumes      minimum
MAXIMUM NUMBER of PHYSICAL VOLUMES1
to use for allocation
Number of COPIES of each logical1
partition
Mirror Write Consistency?      active
Allocate each logical partition copyyes
on a separate physical volume
RELOCATE the logical volume duringyes
reorganization
Logical Volume label           /var/pd
MAXIMUM NUMBER of LOGICAL PARTITIONS512
Enable BAD BLOCK relocationyes
SCHEDULING POLICY for reading/writingparallel
logical partion copies
Enable WRITE VERIFY?           no
File containing ALLOCATION MAP[]
Stripe Size[Not Striped]
```

Commandline:

```
mklv -y'lvpd' -t'jfs' -u'1' -L'/var/pd' pdvg 512 hdisk1
```

7

2. **Create lv01 logical volume**

```
smitty mklv
Vobumeg Group NAME

Logical Volume NAME[lv01]
VOLUME GROUP name[rootvg]
Number of LOGICAL PARTITIONS[512]
PHYSICAL VOLUME names[hdisk0]
Logical volume TYPE[jfs]
POSITION on physical volumemiddle
RANGE of physical volumesminimum
MAXIMUM NUMBER of PHYSICAL VOLUMES1
to use for allocation
Number of COPIES of each logical1
partition
Mirror Write Consistency?active
Allocate each logical partition copyyes
on a separate physical volume
Relocate the logical partition duringyes
reorganization
Logical Volume label/var/psf
MAXIMUM NUMBER of LOGICAL PARTITIONS512
Enable BAD BLOCK relocationyes
SCHEDULING POLICY for reading/writingparallel
logical partion copies
Enable WRITE VERIFY?no
File containing ALLOCATION MAP[]
Stripe Size[Not Striped]
```

Commandline

```
mklv -y'lv01' -t'jfs' -u'1' -L'/var/psf/' rootvg 512 hdisk0
```

3. **Create lvpsfs logical volume**

```
smitty mklv
Vobumeg]Group name

Logical volume NAME[seglv]
VOLUME GROUP name[segvg]
Number of LOGICAL PARTITIONS[512]
PHYSICAL VOLUME names[hdisk2 hdisk3]
Logical volume TYPE[jfs]
POSITION on physical volumemiddle
RANGE of physical volumesminimum
MAXIMUM NUMBER of PHYSICAL VOLUMES2
to use for allocation
Number of COPIES of each logical1
partition
Mirror Write Consistency?active
Allocate each logical partition copyyes
on a separate physical volume
RELOCATE the logical volume duringyes
reorganization
Logical Volume label/var/psf/segments
MAXIMUM NUMBER of LOGICAL PARTITIONS512
Enable BAD BLOCK relocationyes
SCHEDULING POLICY for reading/writingparallel
logical partion copies
Enable WRITE VERIFY?no
File containing ALLOCATION MAP[]
Stripe Size[Not Striped]
```

↓Note

To stripe **/var/psf/segments**, use 128K for the Stripe Size.

Commandline:

```
mklv -y'lvpsfs' -t'jfs' -u'2' -L'/var/psf/segments'
segvg 512 hdisk2 hdisk3
```

3. **Create file systems in logical volumes**

Three file systems are created for InfoPrint Manager to use during installation. The size of file system information would be determined by the size of the SCSI disk drives. A reasonable file system size to use is:

18.2GB disk

> 33554432

36.4GBdisk

> 67108864

73.4GB disk

> 134217728

146.8GB disk

> 268435456

1. **Create /var/pd file system**

```
smitty crjfslvbf
LOGICAL VOLUME name lvpd
MOUNT POINT /var/pd
Mount AUTOMATICALLY at System restart?yes
PERMISSIONS read/write
Mount OPTIONS []
Start Disk Accounting? no
Number of byutes per inode 4096
Allocation Group Size (Mbytes) 64
```

↓Note

1) The `crjfslvbf` is used because this example assumes the creation of a large file enabled JFS system on a previously defined logical volume.

2) The fragment size defaults to 4096.

Commandline:

```
crfs -v jfs -a bf=true -A''`locale nostr | awk -F: '{print $1}'`''
-p'rw'-t''`locale nostr | awk -F: '{print $1}'`'' -a nbpi='4096'
-a ag='64'
```

2. **Create /var/psf file system**

```
smitty crjfslvstd
LOGICAL VOLUME name lv01
MOUNT POINT /var/psf
Mount AUTOMATICALLY at system restart?yes
PERMISSIONS read/write
Mount OPTIONS []
Start Disk Accounting? no
```

7

```
Fragment Size 4096
Number of bytes per inode 4096
Allocation Group Size (Mbytes) 8
```

↓Note

1) The `crjfslvstd` is used because this example assumes the creation of a standard JFS system on a previously defined logical volume.

2) It is best to set the fragment size to its highest value of 4096 bytes.

Commandline:

```
crfs -v jfs -A''`locale nostr | awk -F: '{print $1}'`'' -p'rw'
-t''`locale nostr | awk -F: '{print $1}'`'' -a frag='4096'
-a nbpi='4096' -a ag='8'
```

3. **Create /var/psf/segments file system**

```
smitty crjfslvbf

LOGICAL VOLUME name seglv
MOUNT POINT /var/psf/segments
Mount AUTOMATICALLY at system restart? yes
PERMISSIONS read/write
Mount OPTIONS []
Start Disk Accounting no
Number of bytes per inode 4096
Allocation Group Size (Mbytes) 64
```

↓Note

The fragment size defaults to 4096.

Commandline:

```
crfs -v jfs -a bf=true -A''`locale nostr | awk -F: '{print $1}'`''
-p'rw' -t''`locale nostr | awk -F: '{print $1}'`'' -a nbpi='4096'
-a ag='64'
```

4. **Mount the file systems**

1. **Mount /dev/lvpd**

```
Smitty mountfs
FILE SYSTEM name /dev/lvpd
DIRECTORY over which to mount /var/pd
TYPE of file system jfs
FORCE the mount? no
REMOTE NODE containing the file system []
to mount?
Mount as a REMOVABLE file system no
Mount as a READ-ONLY system? no
Disallow DEVICE access via this mount? no
Disallow execution of SUID and sgid programs no
in this file system?
```

Commandline:

```
mount -v'jfs' /dev/lvpd /var/pd
```

2. **Mount /dev/lv01**

```
Smitty mountfs
FILE SYSTEM name /dev/lv01
DIRECTORY over which to mount /var/psf
```

```
TYPE of file system jfs
FORCE the mount? no
REMOTE NODE containing the file system[]
to mount?
Mount as a REMOVABLE file systemno
Mount as a READ-ONLY system? no
Disallow DEVICE access via this mount?no
Disallow execution of SUID and sgid programsno
in this file system?
```

Commandline:

```
mount -v'jfs' /dev/lv01 /var/psf
```

3. **Mount /dev/lvpsfs**

First, create the directory for the segments:

```
cd /var/psf
mkdir segments
```

🔽 **Note**

Make sure that the group is **printq** and permissions are **744**.

Next:

```
Smitty mountfs
FILE SYSTEM name /dev/lvpsfs
DIRECTORY over which to mount /var/psf/segments
TYPE of file system jfs
FORCE the mount? no
REMOTE NODE containing the file system[]
to mount?
Mount as a REMOVABLE file systemno
Mount as a READ-ONLY system? no
Disallow DEVICE access via this mount?no
Disallow execution of SUID and sgid programsno
in this file system?
```

Commandline:

```
mount -v'jfs' /dev/lvpsfs /var/psf/segments
```

🔽 **Note**

To find the command line equivalent of any command issued through smitty, use the F6 key.

7

8. Linux operating system and hardware tuning

- How much memory is needed?
- How much paging space is needed?
- What affects CPU utilization?
- Network bandwidth
- Network options, Linux settings, and environment variables that affect performance
- CUPS DSS and PSF-Other tuning
- Semaphores
- Setting process priority
- Determining ulimit segment settings
- Disk/File systems

When working on the configuration for your operating system and hardware, start with the operating system recommendations.

The **sar** command can be used as a general command to look at system statistics. You can setup the **sar** command to run at specific intervals to collect system statistics for a specific amount of time. You can also use the **sar** command to schedule sytem evaluations at specific intervals, such as every day, every week, every month, and so forth.

⬇Note

sar command is available only if you install **sysstat** rpm package.

How much memory is needed?

No exact number can be given as the recommendation for the amount of memory needed for a system running InfoPrint Manager. The amount of memory depends on how many jobs are going through the system at what frequency, the size, complexity, and data type of the jobs, and what else is running on the system. Also, the number of queues and actual destinations impact memory utilization.

With InfoPrint Manager, the more memory the better. InfoPrint Manager performs best when it has enough memory that it does not need to use paging space. That is, the more paging space it must use, the worse the performance.

What defines a memory bound system? If the number of active virtual memory pages exceed the amount of real memory pages, page space is used for those pages that cannot be kept in real memory.

To determine if InfoPrint Manager for Linux is using paging space, use the **vmstat**, **free** or **swapon -s** command. With **vmstat**, look at the **swpd memory, swap si** (swap in), and **swap so** (swap out) columns. On a system with no paging, these columns should remain on zero as much as possible. On a machine with memory constraints, you will see the free memory column going toward zero, and then swapping begins.

To start the **vmstat** command, enter:

```
vmstat delay count
```

For more information about other **vmstat** options, see the **vmstat** man page.

How much paging space is needed?

You can use these commands to determine how much paging space is being used on your system:

free

Displays the amount of free and used memory in the system.

8

swapon -s

>   Displays swap usage summary by device. Equivalent to `cat /proc/swaps`.

⬇Note

While the **vmstat** comand shows paging activity, it does not show clearly how much paging space is being used.

To determine how much paging space is needed, it is recommended that the paging space should be double the amount of physical memory. So if a system has 1 gigabyte of memory, the total paging space should be 2 gigabytes (although the 2 gigabytes can be separated onto more than one disk).

On Linux you should consider the following recommendations when configuring your paging space:

*   Use the disk or disks that are least utilized.

*   Do not allocate more than one paging page logical volume for each physical disk.

*   Avoid sharing the same disk with logical volumes.

*   Make all paging spaces the same size if possible.

*   Make all paging spaces have the same priority (man swapon on how to do that).

To make paging space available to your system, you must add and activate the paging space. The total amount of paging space is often determined by trial and error. One commonly used guideline is to double the RAM size and use that figure as a paging space target.

*   To list your current paging space, type: `swapon -s`

*   To add paging space, type: `mkswap device`

*   Add *device* to /etc/fstab

*   To activate paging space, type `swapon -a` or `swapon device`

⬇Note

The *device* argument is usually a disk partition (like /dev/sdd4) but it can also be a file. Usually the partition type for paging is **0x82**. Note that Solaris also uses this type. Make sure you do not overwrite a Solaris partition.

When is a system thrashing? When page space becomes depleted, the operating system attempts to release resources by first warning processes to release paging space and then by killing the processes. Use the **oom_adj** and **oom_score** tools in */proc*/**PID** to set the thresholds at which this activity occurs.

For information and recommendations for changing values for your configuration, see the documentation of your Linux distribution.

What affects CPU utilization?

You can measure CPU utilization with the **iostat** or the **vmstat** tool. The tools can produce a snapshot of the CPU utilization on a system and can be run at regular time intervals to show usage over time. Both tools separate CPU utilization in these categories:

%user

>   Percentage of CPU utilization that occurred while executing at the user level.

%system

Percentage of CPU utilization that occurred while executing at the system or kernal level.

%idle

The percentage of time that the CPU was idle and the system was idle, and the system did not have an outstanding I/O request.

%wait

The percentage of time that the CPU was idle during which the system had an outstanding I/O request.

⬇Note

iostat command is available only if you install *sysstat* rpm package.

When is a CPU bound? A CPU is bound if the sum of the user and system time exceeds 90% of the CPU resource on a single-user system or 80% of the CPU resource on a multi-user system.

With the **vmstat** tool or by looking at the CPU columns, look at the size of the run queue (`procs - r` column). In general, as the run queue increases, users will notice an increase in response time, and therefore, a decrease in performance.

A high **%wait** with **vmstat** or **%iowait** with **iostat** indicates potential I/O bottlenecks. It could be that the system is short of memory, so the disks containing paging space might be busy while paging, and you are likely to see a higher run queue as threads are waiting for the CPU.

Network bandwidth

To measure network activity, use the **netstat** command. The **netstat** command can:

- Display active socket connections for each protocol
- Display routing table information
- Display a table of all network interfaces or a specified interface
- Display summary statistics for each protocol

For information and recommendations using the **netstat** command, consult your operating system manual.

Other commands that can be used in addition to **netstat** are: **ss** (utility to investigate sockets), **ip** (show / manipulate routing, devices, policy routing and tunnels).

To monitor various network statistics including TCP info, TCP/UDP package counts, ICMP and OSPF information, Ethernet load info, node statistics, IP checksum errors use **iptraf** command (only available if **iptraf** rpm package is installed).

Network options, Linux settings, and environment variables that affect performance

You can use the **ethtool** command to set some of the network attributes. Others network variables can be found in `/proc/sys/net/ipv4/*` or `/proc/sys/net/ipv6/*`.

For information or for further reading, see:

- man page for **ethtool** command.

8

- **ip-sysctl.txt** file (available in *kernel-doc* rpm).

CUPS DSS and PSF-Other tuning

CUPS DSS and **PSF-Other DSS** are driving **CUPS** printers. **CUPS** scheduler, **cupsd**, can be configured using the **cupsd.conf** file. It is normally located in the **/etc/cups** directory. Refer to the **CUPS** documentation if you think your printing environment needs changes to the default configuration.

Examples of directives understood by **cupsd**:

- **MaxClients**

  The MaxClients directive specifies the maximum number of simultaneous clients that are allowed by the server. The default is 100 clients.

  Example:

```
MaxClients 1024
```

- **MaxJobs**

  The MaxJobs directive specifies the maximum number of simultaneous jobs that are allowed. Set to 0 to allow an unlimited number of jobs. The default is 500 jobs.

  Example:

```
MaxJobs 0
```

- **LogLevel**

  The LogLevel directive specifies the level of logging for the ErrorLog file. The following values are recognized (each level logs everything under the preceding levels):

  - none - Log nothing
  - emerg - Log emergency conditions that prevent the server from running
  - alert - Log alerts that must be handled immediately
  - crit - Log critical errors that don't prevent the server from running
  - error - Log general errors
  - warn - Log errors and warnings
  - notice - Log temporary error conditions
  - info - Log all requests and state changes
  - debug - Log basic debugging information
  - debug2 - Log all debugging information

  The default LogLevel is @CUPS_LOG_LEVEL@.

  Examples:

```
LogLevel none
LogLevel debug
```

- **PreserveJobFiles**

  Specifies if job files are preserved after a job is printed.

  Allowed Values

Yes

No

seconds

If a numeric value is specified, job files are preserved for the indicated number of seconds after printing. The default is 86400.

Example:

```
PreserveJobFiles No
```

- **PreserveJobHistory**

Specifies if the job history is preserved after a job is printed.

Allowed Values

Yes

No

seconds

If a numeric value is specified, the job history is preserved for the indicated number of seconds after printing. If Yes, the job history is preserved until the MaxJobs limit is reached. The default value is Yes.

Example:

```
PreserveJobHistory No
```

🔽 Note

Each directive is listed on a line by itself followed by its value. Comments are introduced using the number sign ("#") character at the beginning of a line.

Semaphores

The semaphores are counters which are used to provide synchronization between processes or between threads within a process for shared resources like shared memories.

To see all semaphore settings, run:

```
# cat /proc/sys/kernel/sem
250      32000   32       128
```

These values represent SEMMSL, SEMMNS, SEMOPM, and SEMMNI.

Also, you can use the ipcs utility:

```
# ipcs -ls
------ Semaphore Limits --------
max number of arrays = 128
max semaphores per array = 250
max semaphores system wide = 32000
max ops per semop call = 32
semaphore max value = 32767
```

To permanently change the values of the four semaphore parameters, add or change the following line in the file: /etc/sysctl.conf. This file is used during the boot process.

```
# echo "kernel.sem=250 32000 100 1024" >> /etc/sysctl.conf
```

⭐**Important**

If you use a large number of PSF actual destinations, we recommend to set SEMMNI on 1024.

Setting process priority

Use the **nice** command to run a program with modified scheduling priority. A child process inherits the nice value from the parent process. You can view the nice value with the **ps -l** command.

⬇**Note**

Your shell may have its own version of **nice**, which usually supersedes the nice command. Please refer to your shell's documentation for details.

Use the **renice** command to alter priority of running processes.

Use **taskset** to set a process's CPU affinity.

Use **chrt** to change real-time attributes of a process.

For information about process priority, see:

• man page for **nice** command.

• man page for **renice** command.

• man page for **taskset** command.

• man page for **chrt** command.

Determining ulimit segment settings

The **ulimit** command sets or reports process resource limits as defined in the `/etc/security/limits.conf` file. This file contains these limits:

core

> Limits the core file size (KB).

data

> The maximum data size (KB).

fsize

> The maximum filesize (KB).

memlock

> The maximum locked-in-memory address space (KB).

nofile

> The maximum number of open file descriptors.

> If you are running InfoPrint Manager using a non-root user, for example **ipm1**, update `/etc/security/limits.conf` file with:

> `ipm1 soft nofile = 8192`

> `ipm1 hard nofile = 8192`

rss

The maximum resident set size (KB).

stack

The maximum stack size (KB).

cpu

The maximum CPU time (MIN).

nproc

The maximum number of processes.

If you are running InfoPrint Manager using a non-root user, for example **ipm1**, update `/etc/security/limits.conf` file with:

`ipm1 soft nproc = 8192`

`ipm1 hard nproc = 8192`

as

The address space limit (KB).

maxlogins

The maximum number of logins for the user.

maxsyslogins

The maximum number of logins on the system.

priority

The priority to run user process with.

locks

The maximum number of file locks the user can hold.

sigpending

The maximum number of pending signals.

msgqueue

The maximum memory used by POSIX message queues (bytes).

nice

The maximum nice priority allowed to raise to values: [-20, 19].

rtprio

The maximum real-time priority.

Since InfoPrint Manager services can be started by **systemd** at boot, but also on command line using the appropriate scripts, resource limits must be set in multiple places.

1. To modify resource limits for services and processes that are started by **systemd** at boot or with the **systemctl** command, edit the `/etc/systemd/system/ipm-<servicename>.service.d/override.conf` file, uncomment or add the appropriate keyword and set the desired value. Restart the service using **systemctl**. Only root and the user running InfoPrint Manager can start, stop, or restart an InfoPrint Manager service using **systemctl**.

8

The defaults are:

```
LimitNOFILE=8192
LimitNPROC=8192
LimitCORE=infinity
```

For more information, see the man page for `systemd system.conf(5)`.

2. To modify resource limits for services and processes that are started outside of **systemd** (for example on command line), edit the `/etc/security/limits.conf` file. For more information, see the man page for `limits.conf(5)`.

Disk/File systems

The main InfoPrint Manager file systems are:

`/var/pd`
`/var/psf`
`/var/psf/segments`

I/O performance is usually the limiting factor for an InfoPrint Manager production server. Therefore, spreading the InfoPrint Manager file systems across three or more drives is likely to boost performance for an InfoPrint Manager production server. Even more of a performance gain can be accomplished by putting `/var/pd` on one drive, `/var/psf` on another drive, and `/var/psf/segments` across two more drives, striping the two `/var/psf/segments` drives for better I/O performance. The first file system that should be spread across two or more striped drives is `/var/psf/segments`. You could also spread `/var/pd` across two or more striped drives.

Manual InfoPrint Manager file system allocation

The creation steps below and sizes for file systems are based on the assumption that the server is being used only as an InfoPrint Manager print server. The configuration values shown are samples only and will vary based on server type, number and size of disks, and so forth. No considerations have been made for any other applications running in this environment.

File system creation procedure

Before creating local volumes there must be a volume group with enought free space to contain the following logical volumes: `lvpd`, `lvpsf` and `lvpsfs`. Those three logical volumes will be mounted on the following directories: `/var/pd, /var/psf` and `/var/psf/segments`.

If you have multiple hard drives available, you can create the required volume group on more than one disk. Striping across those disks provides faster I/O performance which is useful when driving high speed/high resolution printers.

We assume that the volume group is created. The name of the volume group is **vgname**.

1. **Create `lvpd` logical volume**

   Commandline:

```
lvcreate -L 1024M -n lvpd vgname
```

The above command creates **lvpd** logical volume with 1024MB in size, on **vgname** volume group.

To force **lvpd** logical volume on a specific disk/partition use the following command (assuming that /dev/sdf has at least 5 partitions):

```
lvcreate -L 1024M -n lvpd vgname /dev/sdf5
```

To verify that the logical volume has been created execute the following command:

```
lvs /dev/vgname/lvpd
```

To see the physical drives on which **lvpd** resides, run the following command:

```
lvs /dev/vgname/lvpd -o devices
```

2. **Create** `lvpsf` **logical volumes**

   Commandline:

```
lvcreate -L 1024M -n lvpsfs vgname
```

3. **Create** `lvpsfs` **logical volume**

   Commandline:

```
lvcreate -L 1024M -n lvpsfs vgname
```

4. **Create file systems in logical volumes**

   After all logical volumes have been created, you must create filesystems on them. In this example we use *xfs* filesystem, but you can choose one of these supported Linux filesystems: *ext3*, *ext4*, *xfs*, or *btrfs*.

   Run the following commands for *xfs*:

```
mkfs.xfs /dev/vgname/lvpd

mkfs.xfs /dev/vgname/lvpsf

mkfs.xfs /dev/vgname/lvpsfs
```

5. **Mount the file systems**

   Create directories **/var/pd**, **/var/psf** and **/var/psf/segments**.

   Each logical volume should be mounted on his own directory:

```
mount /dev/vgname/lvpd /var/pd

mount /dev/vgname/lvpsf /var/psf

mount /dev/vgname/lvpsfs /var/psf/segments
```

   🔸 **Note**

   /var/psf/segments is mounted after **/var/psf**

   To make mounts persistent after reboot add the following lines in **/etc/fstab** file:

```
/dev/vgname/lvpd      /var/pd      xfs      defaults      1 2

/dev/vgname/lvpsf     /var/psf     xfs      defaults      1 2
```

8

```
/dev/vgname/lvpsfs  /var/psf/segments  xfs defaults 1 2
```

9. Tuning for GUI Performance

- • **Specifying GUI options to improve performance**
- • **Configuration Tuning**

GUI performance is related to the performance of the entire InfoPrint Manager system. Some things that affect performance:

- • The speed at which the server responds to queries. The GUI sends queries to the InfoPrint Manager server to gather the necessary information to display when the GUI is started. When the server responds to these requests faster, the performance of the GUI is faster.

- • The network speed between the server and the client GUI. When there are a large number of objects on the server, a lot of information needs to be sent across the network when the GUI starts up.

- • The speed of the client computer, that is, the chip or processor speed.

- • The amount of memory on the client computer. The number of objects displayed in the GUI directly affects the amount of memory required on the client computer.

Some of the ways to tune the server for faster performance include:

- • Specifying GUI options to improve performance, p. 77

- • When to use a separate command processor, p. 81

- • When to increase the PD_LISTEN_COUNT, p. 81

Specifying GUI options to improve performance

You can change specific configurations in the GUI to improve performance.

Filtering

Large InfoPrint Manager systems with several servers and thousands of objects contain too much information for the GUI to present efficiently. You can use filters to display a smaller subset of the whole system. You can filter on specific servers, queues, or destinations.

You can also use the **Customize** feature to configure the GUI so that only columns that contain needed information are presented. Increasing the number of columns that are displayed causes the GUI to keep track of more information.

Changing the polling interval

When the GUI is started, it queries the server for the information to display. After these queries finish, the server keeps the GUI informed of changes by sending events to the GUI. However, some information changes too dynamically for the server to efficiently send updates to the GUI. To stay current with the server, the GUI periodically polls the server for the latest values of these dynamic attributes.

In a heavily loaded server, these periodic requests for information can cause problems. For best performance, the GUI should be configured to minimize the amount of polling.

There are two ways to tune GUI polling. With the first, you adjust the polling interval, which defaults to 120 seconds. The polling interval should be set to the maximum amount of time that the operator can go without seeing the latest information. To set the polling interval, go to **Options**→**Customize General** tab in the GUI. Set the GUI polling interval by entering the desired values in the **Update dynamic details every** field.

The second way to tune GUI polling is to configure the GUI so that the dynamic information is not displayed. The GUI must poll for the information displayed in the **Copies Complete**, **Pages**, **Sheets**, and **Est Comp Time** job details columns, and in the **Attention** actual destination column. If the information presented in these columns is not needed, you should use **Options**→**Customize General** tab feature to remove these columns.

During the polling interval, retained jobs that are displayed in the GUI are scanned by the GUI to update the **Delete in** column and the retained job's icon. As the number of retained jobs increases, the scan can cause a momentary hang in the GUI. With systems that contain a large number of retained jobs, you should turn off the **Display retained jobs in main window** option to improve GUI performance.

⬇Note

If none of the dynamic details views columns are visible, the GUI will not poll the server.

Configuration Tuning

Understanding GUI events

The server the GUI is connected to is used to process all the requests the GUI makes for any object in the InfoPrint Manager system. It also receives and manages events sent from all the objects in the InfoPrint Manager System the GUI is monitoring.

InfoPrint Manager server GUI event dispatcher

When an InfoPrint Manager GUI is started, it connects to the command processor part of the pdserver identified by its connection host and socket settings. That GUI makes requests and receives all information from the InfoPrint Manager system through that command processor. Each GUI informs its command processor which servers it is interested in. The command processor then contacts the pdserver or pdservers asking them to send events for the GUI to it. This is called registering for GUI events. The command processor manages all the GUIs connected to it and makes sure events sent from pdservers are delivered to each GUI that has an interest.

Registering servers to send GUI events.
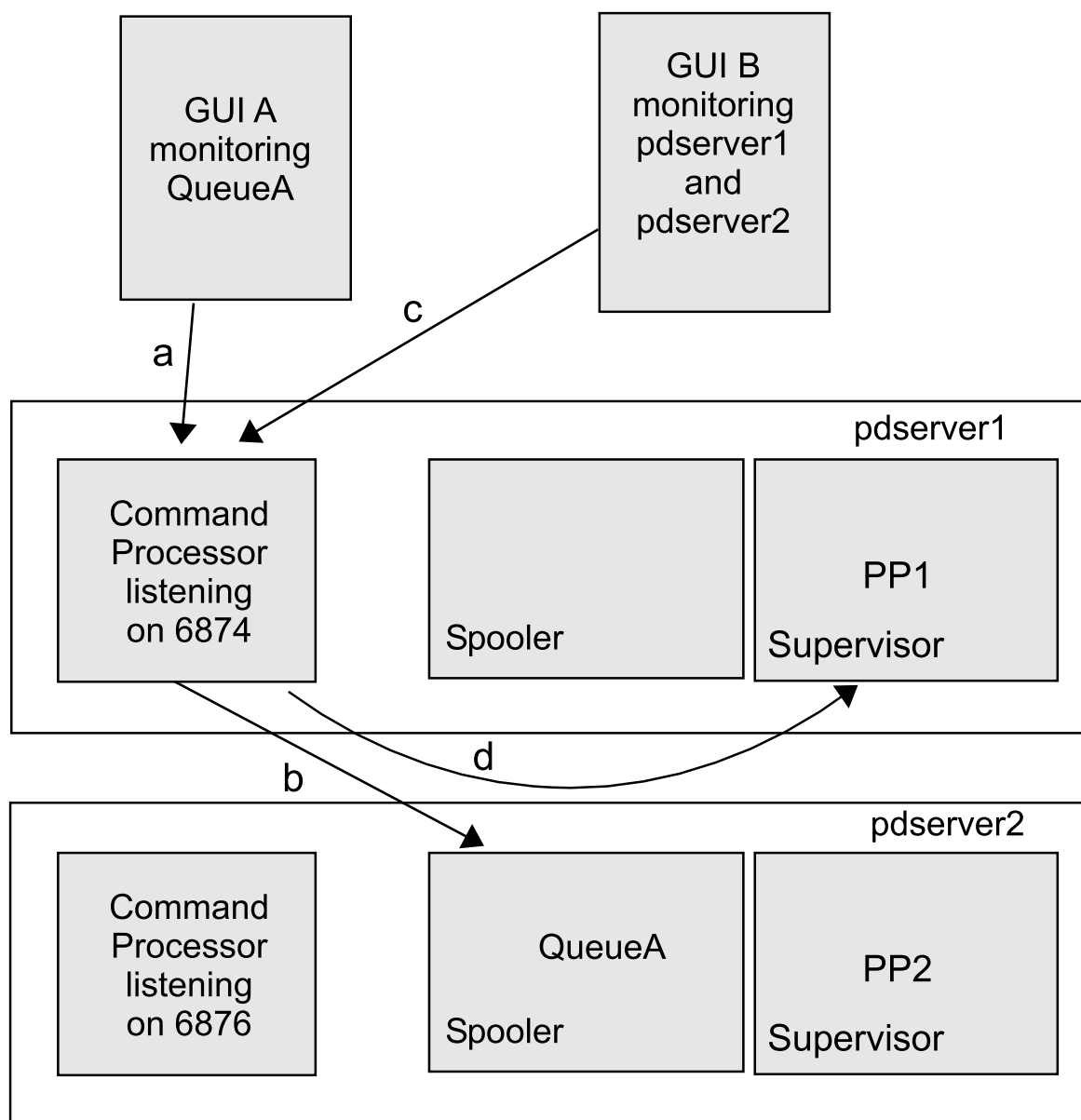
In the previous Figure, GUI A connects (a) to the command processor part of pdserver1 and informs the command processor it wants to monitor Queue A. The command processor part of pdserver1 discovers that Queue A is located in pdserver2 and contacts (b) pdserver2 requesting GUI events be sent. GUI B then connects (c) to the command processor part of pdserver1 and informs the command processor it wants to monitor pdserver1 and pdserver2. The command processor part of pdserver1 knows it has already requested pdserver2 to send it GUI events, so it just contacts (d) the server part of pdserver1 and requests it to send GUI events.

Dispatching GUI events
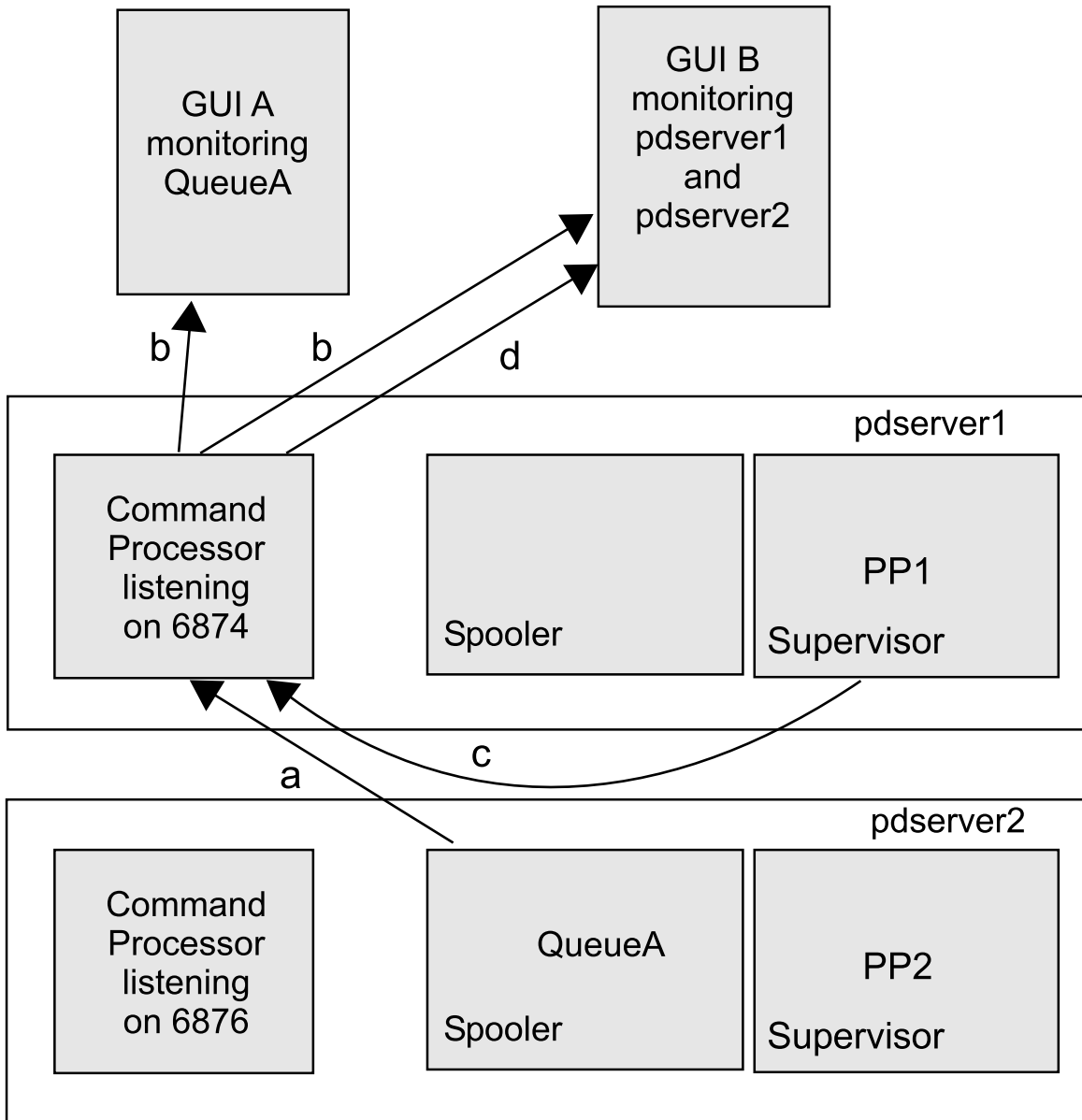
Now as events occur in the servers, the GUIs are kept informed. In the previous Figure, a GUI event is generated in pdserver2 that sends it (a) to the command processor part of pdserver1. The command processor part of pdserver1 knows that both GUI A and GUI B are interested in events from pdserver2, so it sends (b) the event to both GUIs. Then a GUI event is generated in pdserver1 that sends it (c) to the command processor part of pdserver1. The command processor part of pdserver1 knows that only GUI B is interested in events from pdserver1, so it sends (d) the event to GUI B.

When a pdserver starts, there might be GUIs already running that are interested in it. The GUIs are notified and things proceed as normal.

Event registration log messages

When a command processor registers or unregisters with the server part of a pdserver for GUI events, an entry can be made in each of the servers' error logs. The server being asked to send GUI events to a command processor's entry looks like:

```
5010-426 Server pdserver2 received a register signal from communication
daemon pdserver1.
```

The command processsor's entry looks like:

```
5010-454 Successfully registered to receive notification events from
server pdserver2.
```

Whether the messages appear in the server's error log depends on the log severity level setting.

When to use a separate command processor

The pdserver the GUI is connected to is used to process all the requests that the GUI makes for any object in the InfoPrint Manager system. It also receives and manages the multitude of events sent from all the objects in the InfoPrint Manager system the GUI is monitoring. Often all the GUIs and clients are connected to the pdserver running on the default port, 6874. If the pdserver the GUI is connected to is receiving requests from other clients, such as, **pdpr** commands, and doing other functions, such as, printing jobs, it might be doing too much work for it to process GUI events fast enough. Starting a separate pdserver that only works as a command processor, that is, it contains no destination, queue and job objects, is a good solution. Not only does this new pdserver have more cycles to process GUI events, but it also takes some of the load off the pdserver doing the printing functions. It allows AIX to move the processing around to other CPUs in a multi- processor environment too. All threads of a process are run on one CPU. By creating a new process, AIX is free to run this process and its threads on any free CPU.

When to increase the PD_LISTEN_COUNT

9

A given pdserver can handle 10 concurrent requests at a time. It queues up to eight times more additional requests. If more requests than that are received, it returns a `5010-117 Server is too busy` message. This is similar to receiving an "All circuits are busy now, please try your call again later" message from the phone company.

You can increase the number of communication lines (request threads) a pdserver allows by setting a value on the `PD_LISTEN_COUNT` environment variable before starting the pdserver. For example, you can set `PD_LISTEN_COUNT=20` that allows the pdserver to process 20 concurrent requests and queue up to eight times as many more before returning a `5010-117 Server is too busy` message.

Increasing the server communication lines (request threads) can result in any or all of the these:

- The pdserver requires more RAM to handle the additional threads.

- The pdserver requires more CPU time to handle the additional threads.

- More context switching occurs as the processor moves from thread to thread. This further increases the amount of CPU time required.

- If the pdserver contains objects, contention between the threads for access to an object increases causing threads to wait on other threads.

If a command processor pdserver is servicing several GUI clients, as well as sending print files to remote pdservers, you might want to increase its `PD_LISTEN_COUNT`. However, because of the side effects listed for increasing the server communication lines, especially the contention between threads for access to objects, you might not want to increase the `PD_LISTEN_COUNT` on pdservers acting as spoolers or supervisors. Make sure the environment variable is not set when starting these types of pdservers

10. Best practices recommendations

- Use unique server names
- Backup your system regularly
- Applying the latest InfoPrint Manager service updates
- Applying the latest operating system service updates
- Monitor file system utilization
- InfoPrint Manager server error logs
- PSF error logs
- Maintaining accounting logs
- Shutting down a pdserver
- Using the kill options
- Solving problems caused by antivirus software

This section describes some common practices and recommendations that are useful when maintaining an InfoPrint Manager system.

Use unique server names

Naming conventions for InfoPrint Manager server objects should be consistent and clear. Often, the servername reflects the location of the server or some other unique characteristic, such as `boulder1` or `Infoprt2`. Traditional naming conventions have included an `-ld` for logical destinations, a `-q` for queues, and an `-ad` for actual destinations. For example, server `boulder1` has three actual destinations, two logical destinations, and one queue. The server name is `boulder1`; the actual destination names are `printer1-ad`, `printer2-ad`, and `printer3-ad`; the logical destination names are `ip4000-ld` and `printers-ld`; and the queue name that contains all these actual destinations and logical destinations is `boulder1-q`. Remember, a naming convention should be used that is consistent and clearly describes each object.

↓ Note

PSF actual destination names cannot be longer than 64 characters, and pdserver names used with an SAP system cannot be longer than nine characters. For example, **printer1-ad** would need to be **prnt1-ad** for a PSF actual destination.

Backup your system regularly

Making regular backups of your working system should be a normal part of your system maintenance. It is always recommended that you backup your system before applying any InfoPrint Manager or operating system service update.

You can do a full system backup or just backup your InfoPrint Manager configuration.

Full system backup

Use this url to access detailed instructions on backing up your system in the AIX or Linux documentation.

Applying the latest InfoPrint Manager service updates

Staying current with InfoPrint Manager service updates is always recommended.

10

Monitor the InfoPrint Manager web page and download service updates as they become available. To download service updates:

1. Go to https://dl.ricohsoftware.com/.

2. Click **Software Downloads**.

3. Enter your **Entitlement ID** and click **Submit**.

4. In the software list, select **InfoPrint Manager**.

5. Click the appropriate **Download** link or **Publications**.

Applying the latest operating system service updates

Staying current with maintenance is always recommended. Monitor the operating system web page and download service updates as they become available.

Monitor file system utilization

Since temporary files are used when jobs are printed, it is possible that a file system will fill up. It is recommended that you regularly check the file system usage. You can use the **df -k** command to show current file system usage.

InfoPrint Manager provides automatic file system monitoring, as well as notification for low space conditions on the these file systems:

- **/var/pd** (PDBASE)

- **/var/psf**

- **/var/psf/segments**

See "Monitoring memory and space" in *RICOH InfoPrint Manager for AIX: Procedures* or *RICOH InfoPrint Manager for Linux: Procedures* for more information about InfoPrint Manager's automatic file system monitoring and notification.

InfoPrint Manager server error logs

The InfoPrint Manager server error log is located in `var/pd/servername/error.log`.

The error and trace logs have attributes to control their size. They are backed up every time the pdserver is started. By default, only 10 backup error logs are retained, but you can change the number of logs to retain. See the chapter entitled "Customizing error logs in the InfoPrint Manager for AIX server" in the *RICOH InfoPrint Manager for AIX: Procedures* or *RICOH InfoPrint Manager for Linux: Procedures* for more information.

Recommended setting for error logs

The settings for the InfoPrint Manager server `error.log` are specified in `/usr/lpp/pd/bin/spl_error.cfg`. Copy the `spl_error.cfg` file into the `/var/pd/servername` directory if you want to customize the error log only for `servername` or into the `/var/pd/` directory if you want to customize the error log for all InfoPrint Manager servers.

The default setting of `debug` is recommended. The `debug` setting captures error, warning, audit, and debug messages. For more information about the different error log settings, see the "Log attributes" section in the "InfoPrint object attributes" chapter in *RICOH InfoPrint Manager: Reference*.

Working with error logs

To disable an error log, issue this command:

```
pddisable -c log servername:default_error
```

To start using an error, issue this command:

```
pdenable -c log servername:default_error
```

To stop an `error.log` from wrapping, you can do one of two things:

- Issue this **pdset** command from the command line:

```
pdset -c loc -x log-wrap=false servername:default_error
```

- Copy `/usr/lpp/pd/bin/spl_error.cfg` to `/var/pd/spl_error.cfg` and change `log-wrap=true` to `log-wrap=false`.

Note

1. Changes made with **pdset** or to the `spl_error.cfg` file are persistent across runs, that is starts and stops of the pdserver.
2. These changes can also be made on the `spl_trace.cfg` trace logs. The trace object is called `default_trace`. Enabling the InfoPrint Manager tracing mechanism might impact the server performance. Enable it only at the request of a Service Representative.

PSF error logs

InfoPrint Manager maintains these PSF error logs:

- `/var/psf/actual destination name/error.log`, which includes:
  - Connection, printer, or PSF error messages
  - Printer sense bytes
  - Operator command messages

  Note

  InfoPrint Manager only keeps one backup version (`.BAK`) of the PSF error log.

- `/var/psf/appe/apped.log`

- `/var/psf/appe/appeTraceFile.log`

- `/var/psf/pcl2afp/pcl2afpd.log`

- `/var/psf/psf2afp/ps2afpd.log`

- `/var/psf/psf2afp2/ps2afpd.log`

10

Maintaining accounting logs

See "How do you manage the contents of the InfoPrint AIX server accounting logs" in *RICOH InfoPrint Manager for AIX: Procedures* for information about maintaining accounting logs.

See "How do you manage the contents of the InfoPrint Linux server accounting logs" in *RICOH InfoPrint Manager for Linux: Procedures* for information about maintaining accounting logs.

Shutting down a pdserver

You can shut down a pdserver many ways. Perhaps on **AIX** the easiest is to use the **Stop a pdserver** SMIT panel (`smit pd_stop_server`. This panel presents different levels of shutting down the pdserver:

- Select **now** if you want the server to shut down immediately, cancelling any jobs currently printing on the actual destinations contained in the server.

- Select **after-current** if you want the server to shut down after the jobs printing currently on the actual destinations in the server have completed.

- Select **after-all** if you want the server to shut down after the all jobs printing currently on the actual destinations in the server have completed. This includes all pending, processing, and printing jobs.

On **Linux** you can start a pdserver from the InfoPrint Manager Management Interface, go to **InfoPrint Manager->Servers -> Stop**.

You can also shut down the pdserver with the **stop_server** and **pdshutdown** commands. For information about these commands, see the "InfoPrint Manager Tools" chapter in *RICOH InfoPrint Manager: Reference*.

↓Note

The server is not really shut down until the PDB shutdown message is logged. The prompt comes back before the server is completely shut down. That is, the **pdshutdown** command returns success before the server is completely shut down.

Using the kill options

If the pdserver is not responding to commands, the server is hung, and it might need to be "killed." If this situation repeats itself, you should open a Service Request ticket to get help from service to prevent hangs in the future.

Type of kills:

- `kill` or `kill -15` or `kill -TERM` commands are similar to the **pdshutdown -w now** command.
- `kill -6` or `kill -ABRT` allows the pdserver to close the **pdb**, but it generates a full core file for the pdserver. To take advantage of this kill option, make sure your **ulimit -c** value is set to a large enough value to capture the entire core.
- `kill -9` or `kill -KILL` does not close the **pdb**, and this might corrupt the **pdb**.

Solving problems caused by antivirus software

Programs that scan and lock files, such as antivirus products, often cause installation and performance problems.

In some situations, antivirus software has caused RICOH InfoPrint Manager to:

- Fail to install.
- Fail when running a custom step.
- Experience degraded performance.
- Block network traffic on some ports.

You may need to exclude some directories from scanning in order to avoid these consequences.

Suggested directories include:

- /var/pd
- /var/psf

If you experience any of these symptoms while running RICOH InfoPrint Manager on a server with antivirus software installed, contact your Ricoh support representative.

10

Useful AIX commands and tools

For information and recommendations about the commands and tools introduced in this section, see:

- *AIX 5L Performance Tools Handbook*, which is available at http://www.redbooks.ibm.com/redbooks/pdfs/sg246039.pdf.

- AIX documentation, which is available at https://www.ibm.com/docs/en/aix.

Much more detail and examples on these commands are given in these references.

df

The **df** command displays information about total space and available space on a file system. You can specify the name of a device on which the file system is located, the directory on which the file system is mounted, or the relative path name of the file system. You can also specify a file or directory that is not a mount point, and the **df** command displays information for the file system on which the file or directory is located. If you just specify the command with no parameters, it displays information for all currently mounted file systems. File system statistics are displayed in units of 512-byte blocks by default.

The **df** command gets file system space statistics from the **statfs** system call. However, when you specify the **-s** flag, you get the statistics from the virtual file system (VFS) specific file system helper. Under certain exceptional conditions, such as when a file system is being modified while the **df** command is running, the statistics displayed by the **df** command might not be accurate.

⬇️ Note

Some remote file systems, such as the Network File System (NFS), do not provide all the information that the **df** command needs. The **df** command prints blanks for statistics that the server does not provide.

To display information about all mounted file systems, enter:

```
df
```

If your system has the **/, /usr, /site,** and **/usr/venus** systems mounted, the output from the **df** command resembles this:

```
Filesystem 512-blocks  Free %Used Iused %Iused    Mounted on
/dev/hd0       19368    9976  48%   4714     5%    /
/dev/hd1       24212    4808  80%   5013    19%    /usr
/dev/hd2        9744    9352   4%   1900     4%    /site
/dev/hd3        3868    3856   0%    986     0%    /usr/venus
```

netstat

The **netstat** command is a monitoring tool that displays a wide range of network information.

Status information	Command
To find the status of the configured network interfaces	`netstat -in`
To check the routing information	`netstat -rn`
To observe a system's mbuf usage	`netstat -m`
To view the number of mbuf errors	`netstat -v`
To see statistics for all protocols configured on the system	`netstat -s`

Status information	Command
To list the active sockets for your system	`netstat -a`
For only one protocol	`netstat -p <protocol>`
To show statistics for the IP protocol	`netstat -p ip`
To show statistics for the Internet Control Message Protocol	`netstat -icmp`
To show statistics for the TCP protocol	`netstat -p tcp`
To show statistics for the UDP protocol	`netstat -p udp`
To get information about packets sent and received, as well as sent and received and dropped	`netstat -D`
To get the state of all sockets	`netstat -an`
To get information about the network buffer cache	`netstat -c`

vmstat

Use the **vmstat** command to determine where the system performance problems are. **vmstat** reports statistics about kernel threads, virtual memory, disks, and CPU activity.

Here is the syntax for the **vmstat** command:

vmstat [-fsklt] [drives] [interval [count]]

drives

Hard drives to monitor (hdisk0, hdisk1, and so on).

interval

The update period. If **vmstat** is run without an interval, it shows the statistics from reboot. If an interval is set, the first line is the statistics since the system startup, and the next lines show the statistics from the previous line.

count

Number of iterations.

Reports from vmstat	Command
Forks report: Number of forks since the last system startup.	`vmstat -f`
Interrupts report: The example shows an interrupts report with a delay of two seconds, three times.	`vmstat -i <interval> <count>vmstat -i 2 3`
Sum structure report: The absolute count of paging events since system initialization.	`vmstat -s`
I/O activity: The example shows an I/O report taken every two seconds 10 times with time stamps included.	`vmstat -Ivmstat-It 2 10`
Time stamps: The **-t** option gives time stamps on each line of data. This is useful analyzing data.	`vmstat -t <interval> <count>`

topas

The **topas** command is a performance monitoring tool that is ideal for broad spectrum performance analysis. The **topas** command runs in a terminal window and does not have any options for logging information. All information is real time.

Use this command to watch while a system is operating to determine what subsystem should be examined and which processes are doing the most work.

The **topas** command gives this information:

- CPU utilization

- Network Interface Statistics

- Disk drive statistics

- Process statistics

- Event and queue statistics

- File and tty statistics

- Paging statistics

- Memory statistics

- NFS statistics

The **topas** command can be invoked simply by specifying **topas** on the command line. The command also has a number of attributes and subcommands that can be used. See the documentation listed in Useful AIX commands and tools, p. 89 for more details on the **topas** command.

no

You use the **no** (network options) command to set the network attributes.

Network attributes	Command
Print a list of all configurable attributes	`no -a`
Set attribute back to its default value	`no -d attribute`
Display the value of an attribute	`no -o attribute`
Change the value of an attribute	`no -o attribute=NewValue`

When change the attributes with the **no** command, it is not a permanent change, that is, the changed attributes are lost when you reboot. To make sure that the changes are permanent, add them to the `/etc/rc.net` file.

> **Note**

The **no** command does no range checking. If inappropriate attribute values are set, the system could become inoperable.

iptrace

The **iptrace** command can monitor more than one network interface at the same time.

ipreport

The **ipreport** command formats the binary data into a readable format.

ps

The **ps** (process status) command produces a list of processes on the system that can be used to determine how long a process has been running, how much CPU resource processes are using, and if processes are being penalized by the system. It will also show how much memory processes are using, how much I/O a process is performing, the priority and nice values for the process, and who created the process.

The **ps** command has a variety of options. Helpful **ps** commands, p. 92 shows some helpful **ps** commands.

Helpful ps commands

Action	Command
To display the top 10 CPU consuming processes.	`ps aux	head -1 ; ps aux	sort -rn +2	head -10`
To display the top 10 memory consuming processes.	`ps aux	head -1 ; ps aux	sort -rn +3	head` ⬇ Note 10 is the default value for the head command.
To display the top 10 memory consuming processes by **sz** (virtual size in kilobytes of the data section of the process).	`ps -ea1f	head -1 ; ps ea1f	sort -rm +9	head`
To display the processes in order of being penalized by the Virtual Memory Manager (sorted by the **C** column).	`ps -eak1	head -1 ; ps eak1	sort -rn +5`
To display the processes in priority order.	**ps -eak1	sort -n +6	head**
To display the processes in nice order	**ps -eak1	sort -n +7**
To display the processes in CPU time/utilization order.	`ps -vx	head -1 ; ps vx| grep -v PID	sort -rn +3	head -10`

Action	Command
To display the processes in order of real memory usage. The **RSS** value is the size of the working segment and the code segment in 1 kilobyte blocks.	`ps -vx	head -1 ; ps vx| grep -v PID	sort -rn +6	head -10`
To display the processes in I/O order. `PGIN` represents the number of page ins caused by page faults. All AIX I/O is classified as page faults.	`ps -vx	head -1 ; ps vx| grep -v PID	sort -rn +4	head -10`

svmon

The **svmon** command monitors virtual memory. It determines which processes, users, programs, and segments are consuming the most real, virtual, and page space memory. The statistics reported by **svmon** are expressed in terms of pages, where a page is 4 kilobytes.

The **svmon** command can be run like **vmstat** with **intervals** and **count** to record how memory is changing over time:

```
svmon -i<interval> <count>
```

Which takes a snapshot every ***<interval>*** seconds for ***<count>*** times.

Action	Command
To show system wide memory use.	`svmonsvmon -i<interval><count>`
To determine which processes use the most real memory. This example produces the top three real memory usage processes.	`svmon -uP -t 3`
To determine which processes use the most paging space. This example produces the top three processes that use the most paging space.	`svmon -gP -t 3`
To find out how much memory a WLM class is using.	`svmon -W`

Action	Command
To find out what segments are most utilized. This example shows how to find the top *<number>* segments.	**svmon -S -t** *<number>* **-i** *<interval>*
To find out what files a process or command is using, where *<PID>* is the process ID number.	**svmon -pP** *<PID>*
To find out which segments are in the paging space.	```svmon -gS```

xmperf

The **xmperf** program monitors statistics of the system. The **xmperf** program is not generally installed automatically with AIX. It is installable from the Performance Toolbox for AIX.

The **xmperf** program lets users set up data recording of numerous performance variables and then output the files to a spreadsheet format with time and date stamps. You can use the spread sheet for analysis.

The **xmperf** program is a GUI program, so you can monitor performance of the system without recording data. It offers all of the benefits of previously mentioned tools in one complete package.

Since **xmperf** is so large, it requires education to use it, and that education is beyond the scope of this book.

lsof

The **lsof** tool is public domain software that runs on AIX. The **lsof** tool lists open files for a process and can display information about any files that are opened by processes currently running on the system. It can also list communications opened by each process. An open file can be a regular file, a directory, a block special file, a character special file, an executing text reference, a library, a stream or a network file (Internet socket, NFS file, or UNIX domain socket).

To locate the current version of **lsof** for the AIX version and release you are using, search the web using the term "AIX lsof".

pdmemmon

The InfoPrint Manager **pdmemmon** script utility gathers memory-related statistics for a specific pdserver and logs the statistics in a file in PDBASE/*pdserver_name/* by default. Run this script as a background process. The utilities in this script require running as 'root' equivalent.

Note

- The **pdmemmon** script utility installs automatically in `/usr/lpp/pd/bin` when you install InfoPrint Manager for AIX.
- Only a person with `root` user authority or the InfoPrint Manager user can run this command.

Syntax

pdmemmon[**-h**] [**-r** *refresh_secs*] [**-d** *duration*] [**-o** *OutputFile*] *pdserver_name*

pdserver_name

 Name of the pdserver to monitor.

-h

 Display usage information only.

-r *refresh_secs*

 Specify how often to refresh the data in seconds; default is 300 seconds.

-d *duration*

 Specify how many hours to perform this monitoring: default is 24 hours. Specify a duration of 0 to perform only a single iteration of monitoring.

-o *OutputFile*

 Specify the output path and file into which the monitoring information is to be written. If you do not specify the **-o** flag, **pdmemmon** writes the results to standard output.

Useful Linux commands and tools

For more information about usage of the commands and tools introduced in this section, see the man page for each command.

df

The **df** command displays information about total space and available space on a file system. You can specify the name of a device on which the file system is located, the directory on which the file system is mounted, or the relative path name of the file system. You can also specify a file or directory that is not a mount point, and the **df** command displays information for the file system on which the file or directory is located. If you just specify the command with no parameters, it displays information for all currently mounted file systems. File system statistics are displayed in units of 1024-byte blocks by default. The command can also display inodes usage instead of block usage. The output in human readable format (e.g., 1K 234M 2G) is available using **-h** flag.

To display information about all mounted file systems, enter:

```
df
```

The output from the **df** command can resembles this:

```
Filesystem           1K-blocks     Used  Available Use%  Mounted on
/dev/sda3             20642428  8246984   11346868  43%  /
tmpfs                  1927132     9608    1917524   1%  /dev/shm
/dev/sda2               495844    88293     381951  19%  /boot
```

```
/dev/sda7                  445674536 111535892  311499644  27%  /home
/dev/mapper/tmp-sda6         4128448    139488    3779248   4%  /tmp
192.168.1.1:/ipm4r4-psfnc 51606272  17550080   31434752  36%  /ipm4r4-psfnc
192.168.1.1:/ipm4r4        51606272  17550080   31434752  36%  /ipm4r4
```

netstat

The **netstat** command is a monitoring tool that displays a wide range of network information.

Status information	Command
To find the status of the configured network interfaces	`netstat -tn`
To check the routing information	`netstat -rn`
To see statistics for all protocols configured on the system	`netstat -s`
To list both listening and non-listening sockets	`netstat -an`
To show statistics for the TCP protocol	`netstat -tn`
To show statistics for the UDP protocol	`netstat -un`

ip

Use the **ip** command to show and manipulate routing, devices, policy routing, and tunnels.

Status information	Command
To display the network device configuration	`ip link show`
To display the interfaces addresses and their properties	`ip addr show`
To display the eth0 addresses and properties	`ip addr show dev eth0`
To display the eth0 IPv6 addresses and properties	`ip -6 addr show dev eth0`
To display the IPv4 routing table	`ip route show`
To display the IPv6 routing table	`ip -6 route show`
To put a second IPv4 to eth0	`ip add add 192.168.100.1/24 dev eth0 label eth0:0`
To remove a second IPv4 from eth0:	`ip add del 192.168.100.1/24 dev eth0 label eth0:0`

vmstat

Use the **vmstat** command to determine where the system performance problems are. **vmstat** reports statistics about kernel threads, virtual memory, disks, and CPU activity.

Here is the basic syntax for the **vmstat** command:

vmstat [delay [count]]

delay

The delay between updates in seconds. If no delay is specified, only one report is printed with the average values since boot.

count

Number of iterations. If no count is specified and delay is specified on command line, count defaults to infinity.

Reports from vmstat	Command
Report 10 iterations with 2 seconds delay	`vmstat 2 10`
Report every 1 second	`vmstat 1`
Forks report: Number of forks since the last system startup	`vmstat -f`
Disk statistics	`vmstat -d`
Sum structure report: The absolute count of paging events since system initialization	`vmstat -s`
Time stamps: The **-t** option gives time stamps on each line of data. This is useful analyzing data	`vmstat -t <interval> <count>`

iostat

Use **iostat** to display informations about CPU statistics, input/output statistics for devices, partitions and network filesystems (NFS).

The basic syntax for the **iostat** command is:

iostat [delay [count]]

delay

The delay between updates in seconds. If no delay is specified, only one report is printed with the average values since boot.

count

Number of iterations. If no count is specified and delay is specified on command line, count defaults to infinity.

Reports from iostat	Command
Report CPU statistics (one iteration):	`iostat`
Report disk statistics every 1 second, 10 iterations	`iostat -d 1 10`
Report extended disk statistics	`iostat -dx`
Report disk statistics for /dev/sda1 partition	`iostat -d sda`
Report network filesystem (NFS) statistics	`iostat -n`

ps

The **ps** (process status) command produces a list of processes on the system that can be used to determine how long a process has been running, how much CPU resource processes are using, and if processes are being penalized by the system. It will also show how much memory processes are using, how much I/O a process is performing, the priority and nice values for the process, and who created the process.

The **ps** command has a variety of options. Helpful **ps** commands, p. 98 shows some helpful **ps** commands.

Helpful ps commands

Action	Command
To display the top 5 CPU consuming processes.	`ps aux	sort -rnk 3,3	head -n 6`
To display the top 10 memory consuming processes.	`ps aux	sort -rnk 4,4	head`

lsof

The **lsof** tool lists open files for a process and can display information about any files that are opened by processes currently running on the system. It can also list communications opened by each process. An open file can be a regular file, a directory, a block special file, a character special file, an executing text reference, a library, a stream or a network file (Internet socket, NFS file, or UNIX domain socket).

pdmemmon

The InfoPrint Manager **pdmemmon** script utility gathers memory-related statistics for a specific pdserver and logs the statistics in a file in PDBASE/*pdserver_name/* by default. Run this script as a background process. The utilities in this script require running as 'root' equivalent.

⬇ Note

The **pdmemmon** script utility installs automatically in `/usr/lpp/pd/bin` when you install InfoPrint Manager for Linux.

Syntax

pdmemmon[**-h**] [**-r** *refresh_secs*] [**-d** *duration*] [**-o** *OutputFile*] *pdserver_name*

pdserver_name

　　Name of the pdserver to monitor.

-h

　　Display usage information only.

-r *refresh_secs*

　　Specify how often to refresh the data in seconds; default is 300 seconds.

-d *duration*

Specify how many hours to perform this monitoring: default is 24 hours. Specify a duration of 0 to perform only a single iteration of monitoring.

-o *OutputFile*

Specify the output path and file into which the monitoring information is to be written. If you do not specify the **-o** flag, **pdmemmon** writes the results to standard output.

Temporary files created by InfoPrint Manager

Use this information when reading the tables in this appendix:

* *xxxxx* represents random letters and numbers

* 0123456789 represents the 10-digit Job ID

* .1. represents a document number, and can be an integer 1 or greater

* The Segment list file and Segment lists are created and managed by the **psfin** process, which deletes them when space is needed in the /var/psf/segments directory.

Temporary files related to processing jobs created by InfoPrint Manager in /var/pd/*servername*

Files	Persistence	Description of file
0123456789.log	Job in server	Job log
0123456789.1.InfoPrint4000.afp	Job in server	Rip file (AFP output)
0123456789.0.InfoPrint4000.afp	Job printing	Imposition file created by **afpReorder**
0123456789.0.server.js	Job printing	Job level Job Script
0123456789.1.js	Job printing	Document level Job Script
0123456789.1.afpds.js	Job printing	Document level Job Script
0123456789.1.mmconfig	Job printing	MMConfig file passed to **afpReorder**
0123456789.1.afpsplit	Job printing	Page Select AFP file
0123456789.1.reverse	Job printing	Reversed copy of AFP when using **reverse-page-order**
0123456789.1.server.pdpr*xxxx*.0123. trc (only when tracing Input manager)	Never deleted	Input manager segmenter trace file
ainbe*xxxxx*	Ainbe running	Script used to call ainbe
afpReorder.log	Never deleted	Contains log for last invocation of **afpReorder**
pdpr*xxxxx*	Job in server	Copy of submitted file
pjl*xxx*	Job printing	Temporary file containing PJL commands and data for a specific job

Files	Persistence	Description of file
psfin.trc (only when tracing Input manager)	Never deleted	Input Manager trace file
psfin*xxxxx*	Psfin running	Script used to call psfin
stat*xxxxx*	Ainbe running	Ainbe status file

Other temporary files related to processing jobs used by InfoPrint Manager

Filename	Persistence	Description of file
`/var/pd/`*servername*`/pdb/pdpr`*xxxx*	Job printing	Pulled copy of submitted file
`/var/psf/seglist/0123456789.0.`*xxxxx*	Until space is needed	Segment list file
`/var/psf/segments/0123456789.0.`*xxxxx*	Until space is needed	Segment lists
`/var/pd/`*printername*`.dirout1`	While the job is transforming to PCL/PPDS	Temporary file containing the PCL or PPDS output from the job PSF is currently transforming
`/var/pd/`*printername*`.0123456789.dirout2`	While job is printing	Temporary file containing the PCL or PPDS for a specific job

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, use software products successfully. The major accessibility features in InfoPrint Manager let users:

- Use assistive technologies such as screen-readers and screen magnifier software.
- Operate specific or equivalent features using only the keyboard.
- Customize display attributes such as color, contrast, and font size.

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in the InfoPrint Manager for AIX. Consult the assistive technology documentation for specific information when using it to access the InfoPrint Manager for AIX.

Keyboard navigation of the user interface

The InfoPrint Manager user interface complies with the accessibility guidelines for HTML. Users can use keyboard shortcuts or function keys (PF keys) to access the InfoPrint Manager user interface. For example, Alt+Shift+F2 lets you skip to the main content. See your browser documentation for keyboard shortcuts and default settings for the PF keys and for explanations about how to modify their functions.

GLOSSARY

This glossary defines technical terms and abbreviations used in InfoPrint Manager.

Special Characters

.Guidefaults file

A file created by InfoPrint in the home directory of a person using the InfoPrint GUI. InfoPrint 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 GUI, an icon that represents an operation that you can perform on an InfoPrint 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, 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, the person who creates and manages one or more components of a printing system, such as servers and actual destinations. By default, InfoPrint gives administrators authorization to perform some InfoPrint 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 Reader, you can view, navigate, print, and present any Portable Document Format (.pdf) file.

Adobe PageMaker

A desktop publishing program that produces PostScript documents.

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 an pSeries computer through the network or through a serial or parallel port that receives jobs from an AIX print queue.

In InfoPrint, 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, 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, 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. PhotoShop offers anti-aliasing when rasterizing an EPS file.

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 floppy disk, compact disc, or other 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, 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 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, 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, the amount of time calculated by InfoPrint that is required to print all of the jobs currently in a queue.

backspace

In InfoPrint, 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, 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, the destination support system that generates a print command string that the shell runs to print jobs.

BSD physical printer

In InfoPrint, 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, 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, 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, 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, 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, pertaining to the incoming format of the job and the output format required by the printer device. InfoPrint transforms the incoming format into the output format, as required. See also *document format*.

DB2*/6000

DataBase 2/6000.

deadline

In InfoPrint, 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, an object that represents default attribute values for a document in a job. Synonymous with *initial value document*.

default job

In InfoPrint, an object that represents default attribute values for a job. Synonymous with *initial value job*.

default object

In InfoPrint, 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, 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, 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.

DFS

Distributed File Service.

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, 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, 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, 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, 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 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 supports the DPA standard.

document transfer method

In InfoPrint, the transfer method describes how documents are transferred to, or acquired by, servers. See *pipe-pull* and *with-request*.

document type

In InfoPrint, 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, assigning a job to an actual destination as soon as it is accepted. Early binding permits InfoPrint 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, an actual destination representing an electronic mailing system.

email destination support system

In InfoPrint, 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, 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, an occurrence in the printing system during an operation; for example, the completion of a command.

event log

In InfoPrint, a collection of messages about events that have occurred.

event notification

The notification that is sent by InfoPrint 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, 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 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, 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 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 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 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, 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 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, a global character is represented by an asterisk (*).

global ID

Global job identifier.

global job identifier

An unambiguous job identifier. In InfoPrint, 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 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 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 from scheduling it.

hostname

The name given to an AIX system.

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 software and 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 takes documents from creation to the published, kitted, and shipped product.

In InfoPrint software publications, InfoPrint Manager for AIX or any of its components.

InfoPrint Manager for AIX

The software component of InfoPrint. InfoPrint Manager for AIX handles the scheduling, archiving, retrieving, and assembly of a print job and its related resource files. It also tracks the finishing and packaging of the printed product.

InfoPrint Manager for AIX is based on Palladium technology developed at MIT/Project Athena. It conforms to the ISO DPA and POSIX standards.

InfoPrint Network

The local area network running TCP/IP protocol that InfoPrint uses to communicate among servers, clients, and output devices.

InfoPrint Select

The component of InfoPrint Manager for AIX that lets you submit jobs from a Windows workstation.

InfoPrint Submit Express

The component of InfoPrint Manager that lets you submit jobs with a job ticket from a Windows or Macintosh workstation.

InfoPrint 20

A black and white, large-format, cut-sheet, workgroup laser printer with optional duplexing and 600-dpi resolution.

InfoPrint 60

A duplex, black and white, cut-sheet printer with 600-dpi resolution.

InfoPrint 62

A non-impact, continuous-forms printer that runs at a maximum of 62 impressions-per-minute (depending on forms size), and is factory set to run at either 240 or 300 dpi on a maximum paper size of 370.8 mm (14.6 in.).

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, 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, 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. See *local job identifier*, *global job identifier*.

job-priority

In InfoPrint, 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, 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, waiting to assign a job to an actual destination until it is about to be processed. Late binding permits InfoPrint 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 only supported on Red Hat Enterprise Linux (RHEL) 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, a job identifier automatically generated by the server, identifying the job to the person who submitted it. InfoPrint 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, the language InfoPrint uses when sending notification and error messages or displaying the InfoPrint 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, 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, 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, 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 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, 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, 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 developed by Sun Microsystems 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. Contrast with *initially settable attribute* and *resettable attribute*.

notification

The act of reporting the occurrence of events.

In InfoPrint, notification of events appears as messages in event logs or messages sent to administrators, operators, and job submitters. In the InfoPrint GUI, notification of events also appears as changes to the appearance of icons.

notification-profile

In InfoPrint, an attribute associated with an object that contains information designating the people to whom InfoPrint 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 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 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 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, the person responsible for printer devices. Also, this person performs a subset of tasks for InfoPrint 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 GUI, an object that represents a logical destination that is not associated with an existing queue.

orphan logical printer

In the InfoPrint 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, 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.

Palladium

A distributed print system developed at MIT/Project Athena with participation from Digital Equipment Corporation (DEC), International Business Machines (IBM), and Hewlett-Packard (HP). It is a reference implementation for the OSI Document Printing Architecture (DPA) standard, ISO/IEC 10175.

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, 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, referring to options submitted with a job that are used by the device driver, but not InfoPrint. InfoPrint 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, 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, 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, a utility used to display information about a message.

pdpr

In InfoPrint, 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, 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, a document transfer method where InfoPrint saves the documents in a file and transfers the address of the file to the server. InfoPrint 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, 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, 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, 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, the destination support system that communicates with PSF for AIX to print jobs.

PSF physical printer

In InfoPrint, the object representing a printer device that uses the PSF destination support system.

Q

QuarkXpress

A desktop publishing system produced by Quark, Incorporated.

queue

In InfoPrint, 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 uses an Adobe RIP to convert PostScript to IPDS for such printers as the InfoPrint 4000 and InfoPrint 60.

rc.pd

In InfoPrint, a utility used to start the InfoPrint servers listed in the **/etc/pd.servers** file.

read-only

In InfoPrint, 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, 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, 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, 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, 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, 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, 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, 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, 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 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, 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, 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, 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, pertaining to the specific job attribute values that the actual destination can accept during job validation. InfoPrint 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, 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, 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, a PSF physical printer that represents a printer device attached through an MVS system and managed by PSF/MVS. InfoPrint 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, 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)

An IBM licensed 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)

An IBM licensed 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, 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.