

Part 2. Administrative Procedures: Configuring Infoprint Manager

Chapter 3. Using Multiple Network Adapter Cards	9	
Chapter 4. Setting up and using the i-data 7913 IPDS Printer LAN Attachment.	11	
How the i-data 7913 IPDS Printer LAN Attachment fits into network configurations.	12	
Prerequisite tasks	13	
The bootp daemon	13	
Filling out the bootp configuration worksheet	14	
Editing the bootp configuration file	15	
Running the bootp daemon	16	
Setting up the bootp daemon on AIX	16	
Setting up the BOOTP daemon under OS/2	17	
Configuring a twinaxial printer	18	
Setting the controls for the i-data 7913 IPDS Printer LAN Attachment	18	
Model 03 for IBM token-ring networks	18	
Model 04 for ethernet LANs	19	
Attaching the i-data 7913 IPDS Printer LAN Attachment	19	
Starting the i-data 7913 IPDS Printer LAN Attachment	20	
Testing the i-data 7913 IPDS Printer LAN Attachment	20	
Shutting down the i-data 7913 IPDS Printer LAN Attachment	21	
Chapter 5. Selecting a type of actual destination	23	
PC-based applications or a host system using IP Printway	23	
Host system using PSF Direct	24	
MVS Download, DPF, or a combination of PC-based applications and host systems	25	
PSF attachment types	25	
Chapter 6. Using the Internet Printing Protocol (IPP) with Infoprint Manager	27	
Printing to an IPP-enabled printer	27	
Printing through the IPP Gateway	27	
Chapter 7. Using Infoprint Manager notifications	29	
Understanding notification profiles	29	
Event identifiers	30	
Delivery method and delivery address	30	
Infoprint Select and notifications	32	
Using the 'exit' delivery method	33	
Default notification profiles	34	
Using notification profiles with default jobs	35	
Changing notification profiles	36	
For jobs, default jobs, servers, queues, and actual destinations	36	
For logical destinations	36	
Viewing an existing notification profile	37	
Adding or modifying events or users in a notification profile	37	
Removing users from a notification profile	39	
Changing the delivery method	40	
Getting help for notification messages	40	
Other notification methods	41	
Notify-operator attribute	41	
Chapter 8. Managing security.	43	
Types of permission	43	
Security groups	45	
Identifying users and groups: wildcarding	46	
Working with ACLs and groups	46	
Infoprint Manager Security Permissions	46	
Chapter 9. Administering your Infoprint Manager server using a Windows Terminal Server client	49	
Installing the Windows Terminal Server client	49	
Chapter 10. Changing the server hostname and IP address	51	
Changing the server hostname and IP address	51	
Changing the hostname of a non-namespace server	52	
Changing only the server IP address	53	
Chapter 11. Setting up your Infoprint Manager server to use resources on a different Windows system	55	
Giving Infoprint Manager access to remote resources	55	
Using Windows NT	55	
Using Windows 2000	66	
Preparing for shared resources with MVS		
Download — Windows 2000 only	80	
Telling Infoprint Manager where your remote resources are located	82	
Chapter 12. Setting up SNMP.	89	
About SNMP communication	89	
Smart-defaulting actual destination attributes	89	
Detecting, reporting, and recovering from printer problems	89	
Setting up and using SNMP	90	
Determining if a printer works with SNMP	90	
Using the GUI	91	
Procedures	91	
Additional SNMP information	92	
Effects on performance with SNMP	92	
Server start-up	92	
Actual destination queries	92	
SNMP polling	92	
Additional GUI status window	93	
Chapter 13. Creating a new resource-context object	95	

The search order for AFP™ resources	95	Associating auxiliary-sheet objects with PSF printers	135
Chapter 14. Creating a PSF other-driver printer	97	Activating auxiliary sheets for accounting or auditing information	135
Chapter 15. Using a PSF other-driver for printing all jobs from a single actual destination	101	Associating the post-print accounting user-exit program with an actual destination	135
Configuring the printer	101	Associating an input data user-exit program with an actual destination	135
Document formats that cause a job to be transformed	101	Associating an output data user-exit program with an actual destination	135
Job attributes that cause a job to be transformed	102		
PSF header sheet attributes that cause a job to be transformed	103		
Chapter 16. Using PSF DSS user-exit programs	105	Chapter 18. Gathering accounting and auditing data about print jobs	137
Supported types of PSF DSS user-exits	105	Working with Infoprint Manager server accounting information	137
Sample PSF DSS user-exit programs	106	How does the pdaccount command collect Infoprint Manager accounting information?	137
Creating and using your own PSF DSS user-exit programs	107	How do you manage the Infoprint Manager accounting logs?	139
Compiling and installing the user-exit program	107	Usage considerations by DSS for accounting information	140
Activating the user-exit program	108	Working with PSF accounting, post-print accounting, and auditing data about the job	142
Header page, trailer page, and separator page user-exit program inputs and outputs	108	What do the accounting and audit PSF DSS user exits provide?	143
Fields that provide input information	109	What does the post-print accounting user exit provide?	143
Other input values	110	How do you format accounting, post-print accounting, and audit data for viewing?	144
Fields that provide output information	112	How do you manage the contents of the data files?	145
Accounting, post-print accounting, and audit user-exit program inputs and outputs	112		
Fields that provide input information	113	Chapter 19. Using the Infoprint Manager line printer daemon (LPD)	147
Fields that provide output information	115		
Input data user-exit program inputs and outputs	115		
Fields that provide input information	115		
Output data user-exit program inputs and outputs	116		
Fields that provide input information	117		
Structure of a user-exit program	117		
User-exit programs for the line data transform	118		
apka2e	118		
asciinp.c	118		
asciinpe.c	119		
Input record exit	119		
Output record exit	121		
Resource exit	122		
Non-zero return codes	124		
Attributes of the line data input file	124		
Using the iconv command to convert code pages	125		
Example 1	126		
Example 2	126		
Chapter 17. Creating and managing auxiliary sheets and PSF DSS user-exit programs	127		
Preparing to work with auxiliary sheets	127		
Terms relating to auxiliary sheets	127		
Infoprint default auxiliary-sheet objects	128		
Examples of Brief-Style and Full-Style Auxiliary Sheets	129		
Sequence of user-exit programs	132		
Directory locations of user-exit programs supplied with Infoprint Manager	133		
Creating and configuring auxiliary sheet objects	134		
Creating new auxiliary-sheet objects using the pdcreate command	134		

Chapter 3. Using Multiple Network Adapter Cards

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

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

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

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

- Class A IP address: 10.xx.xxx.xxx
- Class B IP address: 192.168.xxx.xxx
- In the properties for the secondary network adapter card, you should not set a Default Gateway, DNS, or WINS. If these fields are filled in when you configure the card, clear the default values.
- On the printers that you connect to the secondary network, do not specify a Default Gateway. If you must specify a Default Gateway, enter the IP address of the secondary network adapter card.

Chapter 4. Setting up and using the i-data 7913 IPDS Printer LAN Attachment

The i-data 7913 IPDS Printer LAN Attachment is a hardware protocol converter that communicates between Infoprint and a coaxial or twinaxial IPDS printer device. You can use the i-data 7913 IPDS Printer LAN Attachment to connect this type of printer to any LAN-attached workstation. The i-data 7913 IPDS Printer LAN Attachment requires TCP/IP communications protocols and runs on either an IBM token-ring network or an Ethernet LAN. You must install one i-data 7913 IPDS Printer LAN Attachment for each coaxial or twinaxial IPDS printer device that you plan to install, and you must install and configure each unit separately.

In order to use the 7913, you must also configure and start a bootp daemon on a system somewhere in your TCP/IP network. The bootp daemon initializes the i-data 7913 IPDS Printer LAN Attachment with the TCP/IP address information that it uses to communicate over the network. bootp daemons are available for most operating systems; AIX and OS/2 provide bootp daemons in their base TCP/IP packages. If you have an AIX or an OS/2 system that can run the bootp daemon on your network, you can use it. Instructions for configuring and starting the bootp daemon on both of those systems are provided in this chapter. The base Windows TCP/IP does not include a bootp daemon, but versions are available from Microsoft and other vendors as extensions to the base TCP/IP package. You can install one of those products using the documentation provided by the other vendors.

Once you have installed and tested the i-data 7913 IPDS Printer LAN Attachment and the bootp daemon, you can define it to Infoprint Manager as a PSF TCP/IP-attached printer. Create this actual destination using the Infoprint Manager Administration GUI.

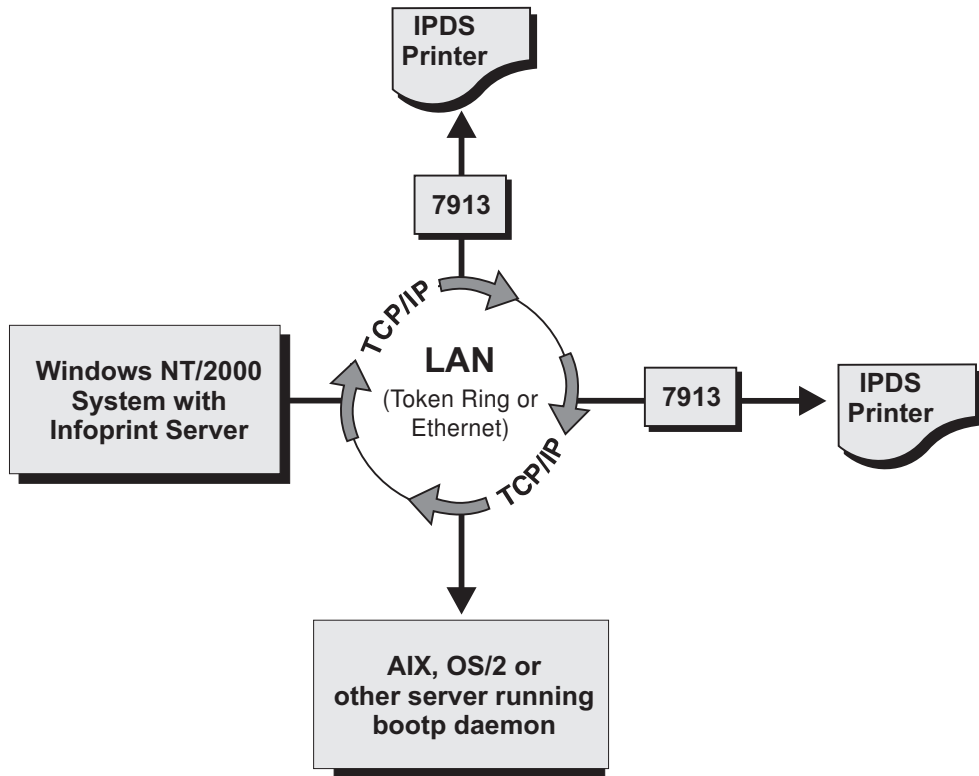


Figure 1. i-data 7913 IPDS Printer LAN Attachment in a TCP/IP LAN Environment

How the i-data 7913 IPDS Printer LAN Attachment fits into network configurations

Use the configuration in Figure 2 for an TCP/IP network with the **bootp** daemon installed on another server. Normally, the server must belong to the same subnetwork as the i-data 7913 IPDS Printer LAN Attachment.

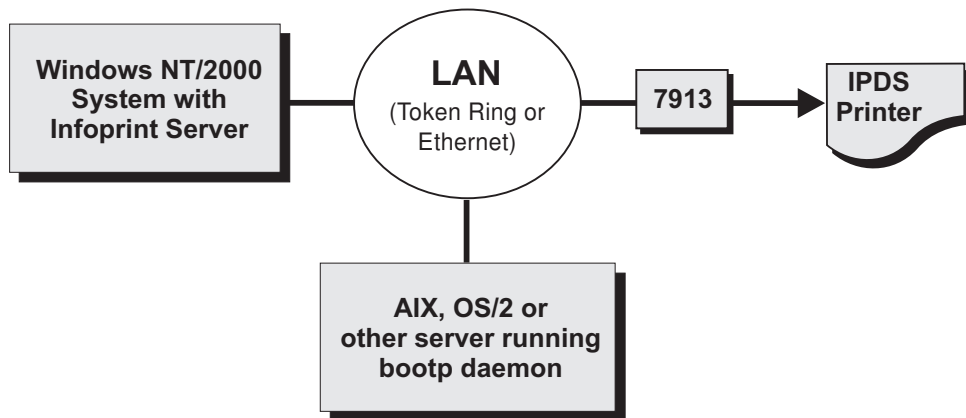


Figure 2. Network Configuration for Separate Internet Protocol Hosts

Prerequisite tasks

Before you begin to install the i-data 7913 IPDS Printer LAN Attachment, you must satisfy the following conditions:

1. You have one i-data 7913 IPDS Printer LAN Attachment for each IBM coaxial or twinaxial printer that you plan to install. You can order the 7913 unit as a printer feature.

The following printers require the i-data 7913 IPDS Printer LAN Attachment:

- IBM 3112 Page Printer Model 001
- IBM 3116 Page Printer Models 001, 002, and 003
- IBM 3112 Page Printer Model 001
- IBM 3812 Page Printer Model 2 with feature 3190 or 3192
- IBM 3816 Page Printer Models 01D and 01S with feature 7653 or 7654
- IBM 3912 Page Printer Models AS1 and NS1
- IBM 3916 Page Printer Models AS1 and NS1
- IBM 3930 Page Printer Models 02D and 02S
- IBM Laser Printer 4028 Models AS1 and NS1, including support for the 4028 with magnetic ink character recognition (MICR), Request for Price Quotation (RPQ) S02177
- IBM 6400 Line Matrix Printer (all models)
- IBM 6408 Line Matrix Printer Model CTA

Note: Infoprint Manager does not support twinaxial connections for this printer device.

- IBM 6412 Line Matrix Printer Model CTA

Note: Infoprint Manager does not support twinaxial connections for this printer device.

2. You have installed IBM Infoprint Manager for Windows, Version 2, Release 1 or higher.
3. You have either an IBM token-ring network or an Ethernet LAN installed and operating.
4. For each i-data 7913 IPDS Printer LAN Attachment that you plan to install, you have obtained the following information from your network administrator:
 - Internet Protocol (IP) address
 - Subnet mask
 - Gateway address

The bootp daemon

The TCP/IP **bootp** daemon supports one or more i-data 7913 IPDS Printer LAN Attachments. When you start an i-data 7913 IPDS Printer LAN Attachment, it broadcasts an initialization request to all systems on the LAN. The **bootp** daemon searches the **/etc/bootptab** file and returns information that defines the i-data 7913 IPDS Printer LAN Attachment within the network.

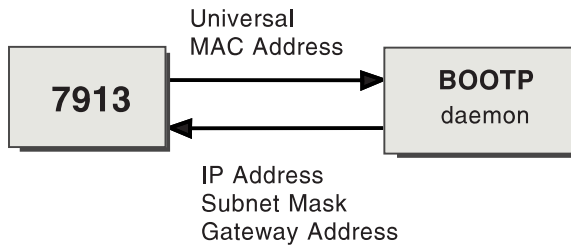


Figure 3. Data Flow between the i-data 7913 IPDS Printer LAN Attachment and the bootp Daemon

Figure 3 shows the exchange of the following data:

Universal medium access control (MAC) address

The universal MAC address is unique for each i-data 7913 IPDS Printer LAN Attachment. It is printed on the side panel of the i-data 7913 IPDS Printer LAN Attachment and defined in the hardware. The i-data 7913 IPDS Printer LAN Attachment broadcasts its universal MAC address when it starts. The **bootp** daemon uses the universal MAC address to find the i-data 7913 IPDS Printer LAN Attachment in the **bootptab** file.

IP address

The IP address is the full address of the i-data 7913 IPDS Printer LAN Attachment as defined to the network. This address must be unique for each i-data 7913 IPDS Printer LAN Attachment.

Subnet mask

The subnet mask is a bit template that indicates which part of the IP address represents the network. A 1 in the subnet mask indicates that the corresponding bit in the IP address is part of the network address; a 0 indicates that it is not. For example, if the IP address is 9.99.12.137 and the subnet mask is 255.255.255.0, the network address is 9.99.12.0.

Gateway address

The gateway address is the address of the system that routes data to other networks.

Filling out the bootp configuration worksheet

Before you begin to modify the **/etc/bootptab** file, fill out a **bootp** configuration worksheet for each i-data 7913 IPDS Printer LAN Attachment.

Table 1. bootp Configuration Worksheet

Parameter	Where to Find the Value	Value
Name	Any name that identifies the i-data 7913 IPDS Printer LAN Attachment	
Network type (token ring or Ethernet)	System administrator	
Ring speed (token-ring only)	System administrator	
Universal MAC address	Side panel of the i-data 7913 IPDS Printer LAN Attachment	
IP address	System administrator	
Subnet mask	System administrator	
Gateway address	System administrator	

Table 1. bootp Configuration Worksheet (continued)

Parameter	Where to Find the Value	Value
Printer type (twinaxial or coaxial)	Printer documentation	

Editing the bootp configuration file

bootp daemons use a text configuration file, usually named `bootptab`. On AIX, this file resides in the `/etc` filesystem. On OS/2, the file is placed in the `/etc` subdirectory under the install path for the TCP/IP product (usually in `/tcpip/etc`). Existence and location of the configuration file on Windows system depend upon the TCP/IP extension that you selected.

The **bootp configuration** file contains a separate entry with configuration information for each i-data 7913 IPDS Printer LAN Attachment. To configure this file, edit it and add the values from Table 1 on page 14. You must have **Administrator** authority to edit the **bootp configuration** file.

Figure 4 shows an example of a simple **bootp configuration** file. There are entries for two i-data 7913 IPDS Printer LAN Attachments with the same subnet mask and gateway address.

```
P4028: ht=6: ha=00036EFFFFFF: ip=9.9.12.137:-
      sm=255.255.255.0: gw=9.99.12.254:
P3812: ht=6: ha=00036E00049F: ip=9.9.12.138:-
      sm=255.255.255.0: gw=9.99.12.254:
```

Figure 4. A Simple `bootptab` File

The keywords and fields in Figure 4 are as follows:

EntryName

A name of your choice that identifies the entry. This name has no meaning to the system.

ht=*HostHardwareType*

Enter **6** for a token ring network or **ethernet** for an Ethernet LAN. This keyword must precede the **ha** keyword.

ha=*HostHardwareAddress*

The universal MAC address of the i-data 7913 IPDS Printer LAN Attachment.

ip=*IPAddress*

The IP address of the i-data 7913 IPDS Printer LAN Attachment.

sm=*SubnetMask*

The subnet mask for the IP address.

gw=*GatewayAddress*

The gateway address for the network.

: Delimiter indicating the end of a value.

- Continuation character indicating that the entry continues on the next line.

Figure 5 on page 16 shows a more complex way to describe the same **bootp** configuration as in Figure 4. The file begins with a global entry for the subnet mask used by all i-data 7913 IPDS Printer LAN Attachments. An entry for the

gateway address points to the subnet mask entry. In turn, each entry for an i-data 7913 IPDS Printer LAN Attachment points to the gateway address entry. This style is convenient to use when many i-data 7913 IPDS Printer LAN Attachments share the same subnet mask and gateway address. If the gateway address changes, you need to update only one entry.

```
# Global entry with information that every host uses
global.lan:-
    sm=255.255.255.0:

# Subnet entry with gateway address
subnetida.trlan:-
    tc=global.lan: gw=9.99.12.254:

# Entries for 7913 units
P4028: tc=subnetida.trlan: ht=6: ha=00036EFFFFFF:-
    ip=9.9.12.137:

P3812: tc=subnetida.trlan: ht=6: ha=00036E00049F:-
    ip=9.9.12.138:
```

Figure 5. A More Complex bootptab File

In addition to the keywords you saw in Figure 4 on page 15, Figure 5 contains the following keywords:

tc=template

A pointer to another entry that contains addressing information.

Comment character

Running the bootp daemon

Once the bootptab configuration file has been edited to include the TCP/IP network information for each of your i-data 7913 IPDS Printer LAN Attachment, follow instructions specific to your operating system to start the bootp daemon. The daemon can be configured to be controlled by the inet superdaemon by adding the bootp daemon to the list of services that the inet daemon controls. The bootp daemon may also be started manually from a command line at boot time if you want to maintain control over the daemon.

If you do not want to maintain an active bootp daemon on your network, the i-data 7913 IPDS Printer LAN Attachment allows you to load default network information into the device; the 7913 automatically uses this information if no bootp daemon responds to the address request within the timeout period (usually about two minutes). Follow the detailed instructions provided with your 7913 to modify your bootptab configuration file to force the network information into the device. You will still have to temporarily start a bootp daemon to initially load the network information into each 7913, but once the information is loaded, the bootp daemon will only be required if you want to change network information for a specific i-data 7913 IPDS Printer LAN Attachment, if you wanted to add an additional 7913 to your network, or if you needed to load updated microcode into existing i-data 7913 IPDS Printer LAN Attachments.

Instructions for setting up the bootp daemon on AIX and OS/2 systems follow.

Setting up the bootp daemon on AIX

After you have configured the **bootp configuration** file, you must set up and start the **bootp** daemon on AIX. Follow these steps:

1. Log on as **root**.
2. Add the following lines to the **/etc/services** file if they are not already present:


```
bootps 67/udp # bootp server port
bootpc 68/udp # bootp client port
```
3. Use *one* of the following methods to start the **bootp** daemon:
 - To have the **inetd** super-daemon start the **bootp** daemon, follow these steps:
 - a. Add the following line to the **/etc/inetd.conf** file if it is not already present:


```
bootps dgram udp wait root /etc/bootpd bootpd
```
 - b. On the AIX command line, enter the following commands so that the **inetd** super-daemon can recognize the update:


```
inetimp
refresh -s inetd
```
 - To start the **bootpd** daemon manually, follow these steps:
 - a. Verify that the **/etc/inetd.conf** file does not include a line that controls the **bootpd** daemon. If necessary, comment out or delete the line and refresh the **/etc/inetd.conf** file.
 - b. On the AIX command line, enter one of the following commands:


```
bootpd -s
```

or

```
bootpd -s -d -d -d -d -d
```

The **-s** flag indicates that the **inetd** super-daemon does not control the **bootpd** daemon. Each **-d** flag displays an additional amount of debugging information. You can specify the **-d** flag up to five times.

Setting up the BOOTP daemon under OS/2

To use the BOOTP daemon under OS/2, the SERVICES file in the ETC directory (**/tcpip/etc/services**) must contain the following entries:

```
sbootp 67/udp # bootp server
cbootp 68/udp # bootp client
```

To test the BOOTPD command, enter the following from an OS/2 command line:

```
bootpd -d -d -d -d -d
```

where:

- **-d** displays debugging information, such as hardware address. Each additional **-d** increases the amount of debugging information displayed. You can specify **-d** up to five times.

When you are satisfied that the BOOTPD command is working correctly, you can add the command to your startup procedure.

Add the BOOTPD command to the **/tcpip/bin/tcpstart.cmd** file (assuming that TCP/IP was installed in the **tcpip** subdirectory). If the BOOTPD and the 7913 are on the same LAN segment (no bridges), enter the following:

```
start BOOTPD
```

If the BOOTP daemon and the 7913 are on different LAN segments (bridges required), enter the following:

```
start BOOTPD -b
```

Configuring a twinaxial printer

If you are using the i-data 7913 IPDS Printer LAN Attachment to connect a twinaxial printer, set the **Station address** to **0** at the printer device.

Note: You cannot use the **CANCEL** button on a twinaxial printer to cancel print jobs.

Setting the controls for the i-data 7913 IPDS Printer LAN Attachment

There are two models of the i-data 7913 IPDS Printer LAN Attachment, each with different controls:

- Model 03 for IBM token-ring networks
- Model 04 for Ethernet LANs

You must set the controls before you attach the i-data 7913 IPDS Printer LAN Attachment to the printer or to the network.

DANGER

Disconnect the i-data 7913 IPDS Printer LAN Attachment from the electrical outlet before setting the controls.

Model 03 for IBM token-ring networks

Figure 6 shows the i-data 7913 IPDS Printer LAN Attachment Model 03 for IBM token-ring networks.

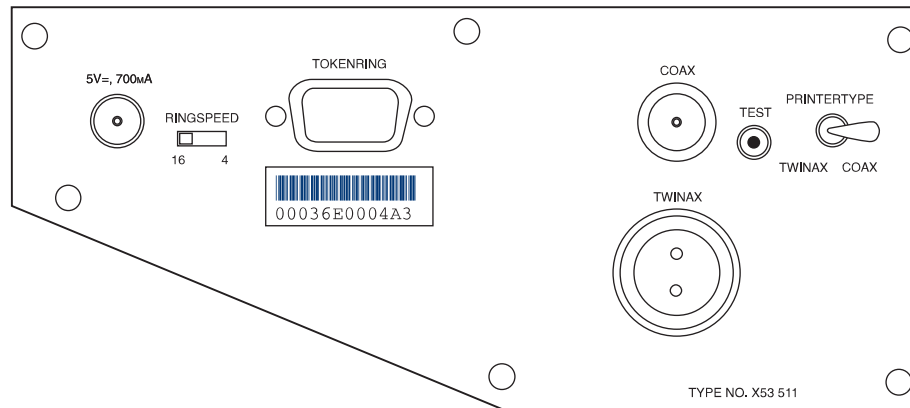


Figure 6. i-data 7913 IPDS Printer LAN Attachment Model 03 for IBM Token-Ring Networks

To set the controls on this model, follow these steps:

1. Disconnect the i-data 7913 IPDS Printer LAN Attachment from the electrical outlet.
2. Set the **RINGSPEED** switch to the speed of the token ring: **4** or **16** megabits per second. You can find out the speed from your system administrator.
Attention: If the **RINGSPEED** switch is not set correctly, the LAN may malfunction.
3. Set the **PRINTERTYPE** switch to **COAX** for an IBM coaxial printer or **TWINAX** for an IBM twinaxial printer.

Model 04 for ethernet LANs

Figure 7 shows the i-data 7913 IPDS Printer LAN Attachment Model 04 for Ethernet LANs.

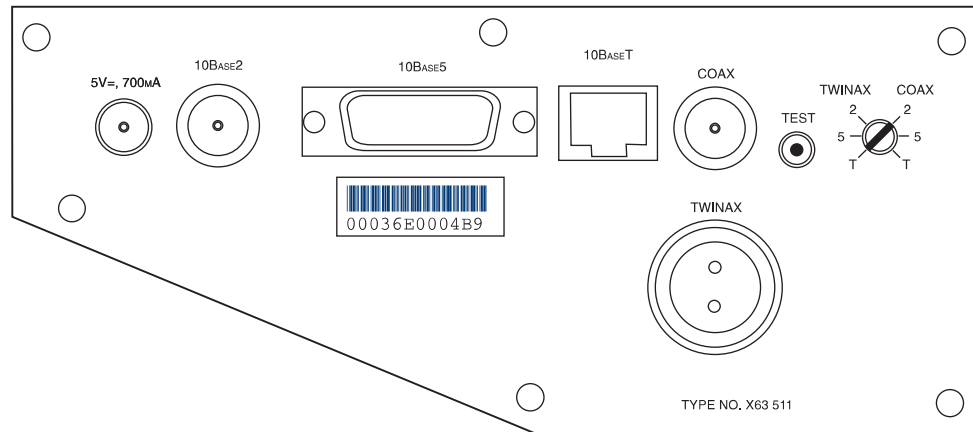


Figure 7. i-data 7913 IPDS Printer LAN Attachment Model 04 for Ethernet LANs

To set the controls on this model, follow these steps:

1. Disconnect the i-data 7913 IPDS Printer LAN Attachment from the electrical outlet.
2. Set the **TWINAX/COAX** rotary switch to one of the following positions:
 - COAX 2**
10base2 to coax
 - COAX 5**
10base5 to coax
 - COAX T**
10baseT to coax
 - TWINAX 2**
10base2 to twinax
 - TWINAX 5**
10base5 to twinax
 - TWINAX T**
10baseT to twinax

Attaching the i-data 7913 IPDS Printer LAN Attachment

Attach the i-data 7913 IPDS Printer LAN Attachment by cabling it to the token ring or LAN and to the printer. For coaxial printers, you must also install the ferrite cable block shipped with the i-data 7913 IPDS Printer LAN Attachment. The ferrite cable block reduces electrical interference.

Note: Cables are not provided with the i-data 7913 IPDS Printer LAN Attachment or with Infoprint Manager.

1. Unplug the i-data 7913 IPDS Printer LAN Attachment.
2. Use a LAN connection cable to attach the i-data 7913 IPDS Printer LAN Attachment to the token ring or LAN.
3. Use a coaxial or twinaxial connection cable to attach the i-data 7913 IPDS Printer LAN Attachment to the printer.

4. For coaxial printers, clamp the ferrite cable block to the coaxial cable. Position the ferrite cable block about 10 centimeters (4 inches) from the i-data 7913 IPDS Printer LAN Attachment.

Starting the i-data 7913 IPDS Printer LAN Attachment

To start the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Turn on the printer device.
2. Plug the i-data 7913 IPDS Printer LAN Attachment power cable into an electrical outlet.

The **POWER** light goes on. Several seconds later, the **LAN** light blinks. When the **bootp** daemon initializes the i-data 7913 IPDS Printer LAN Attachment, the **LAN** light changes from blinking to steady.

Note: If no bootp daemon responds within the timeout limit, the i-data 7913 IPDS Printer LAN Attachment uses default network information that it has stored internally, if it is available. If no default information has been stored, the 7913 device will remain unavailable to your network.

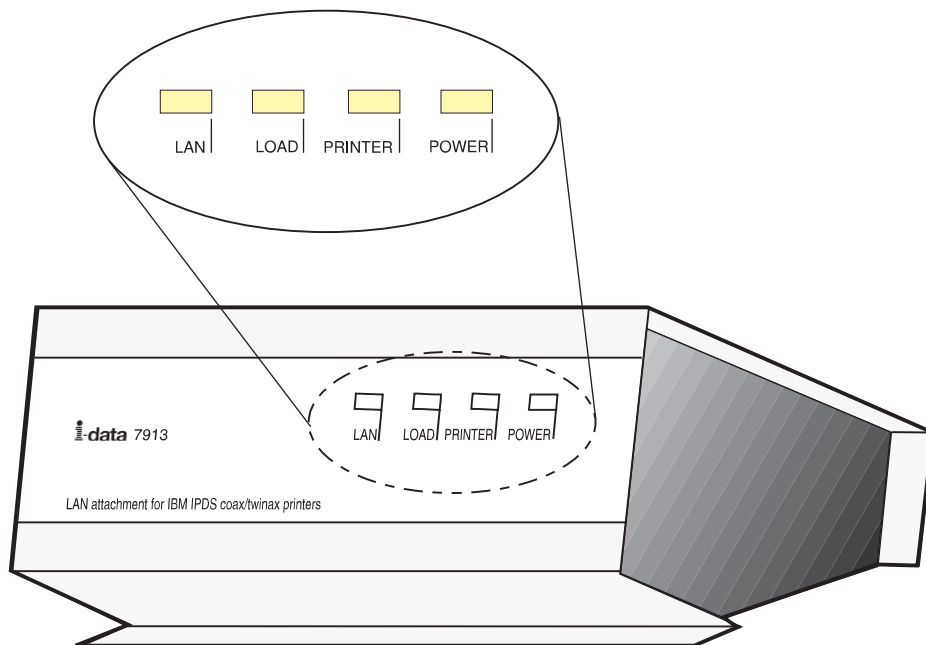


Figure 8. Status Indicators on the i-data 7913 IPDS Printer LAN Attachment

Testing the i-data 7913 IPDS Printer LAN Attachment

After you have started the i-data 7913 IPDS Printer LAN Attachment, follow these steps to test it:

1. Press the **TEST** button on the side panel of the i-data 7913 IPDS Printer LAN Attachment. This breaks the connection between Infoprint Manager and the i-data 7913 IPDS Printer LAN Attachment and causes the printer to print a test page.
2. From any system on your LAN, ping the i-data 7913 IPDS Printer LAN Attachment to test both the physical LAN connection and the IP network connection.

- a. Open a DOS window. On Windows NT systems, click the Windows Start button and select **Program** → **Command Prompt**. On Windows 2000 systems, click the Windows Start button and select **Program** → **Accessories** → **Command Prompt**.
- b. Type **ping** and the IP address or host name of the i-data 7913 IPDS Printer LAN Attachment, for example:
`ping 9.99.12.73`

If the **ping** command is successful, the response looks like this:

```
ping 9.99.12.73 with 32 bytes of data
Reply from 9.99.12.73: bytes = 32 time<10ms ttl=128
Reply from 9.99.12.73: bytes = 32 time<10ms ttl=128
Reply from 9.99.12.73: bytes = 32 time<10ms ttl=128
Reply from 9.99.12.73: bytes = 32 time<10ms ttl=128
```

Shutting down the i-data 7913 IPDS Printer LAN Attachment

To end the connection between Infoprint and the i-data 7913 IPDS Printer LAN Attachment, you should treat it as you would any other physical destination. Issue a shutdown command to break the connection to Infoprint Manager, then unplug the device to ensure that the connection will not be inadvertently restored.

Chapter 5. Selecting a type of actual destination

Once you have decided how many logical destinations, queues, and actual destinations you want to use and how to configure them, you need to determine which kind of actual destinations to create. In Infoprint Manager for Windows, there are four basic types of actual destinations: PSF printers (including PSF Direct only destinations), Passthrough printers, BSD printers, and IPP printers. The names of the destination types refer to the destination support systems (DSSs) that process the print data and send it to the printer.

Consult the tables below to help you determine what kind of destinations to create based on where your print jobs are coming from:

- PC-based applications or a host system using IP Printway
- Host system using PSF Direct
- Both PC-based applications and host systems (using PSF Direct or MVS Download)

PC-based applications or a host system using IP Printway

Use this table to see which type of destination fits your needs.

Table 2. PC-based applications or a host system using IP Printway

If the data stream that your applications send to Infoprint Manager is:	PostScript, PCL or any non-IPDS data stream			
and the data stream that you send to this printer is:	the same that the application sends to Infoprint Manager			PCL, PPDS ¹ , IPDS
and your jobs are going to be sent to a printer:	through a Windows defined port	using a command such as lpr	that is IPP enabled	that is attached to your Infoprint Manager system in any of the ways listed in the PSF Attachment Types table
Create this kind of actual destination:	Passthrough ²	BSD	IPP	PSF
Notes:				
1. You can only print PPDS on a non-Windows system, for example, by sending the PPDS data stream to an AIX system.				
2. If you are printing PCL data to an NT-defined printer, and have therefore chosen to create a Passthrough destination, you may want to consider creating a PSF (other driver or command) destination instead. A Passthrough destination uses fewer processing resources, but a PSF destination provides much more functionality, such as accounting and automatic datastream transforms.				

For example, an advertising firm is installing Infoprint Manager for Windows to manage printing for its 1,200 employees. The company has 450 PCL and PostScript printers that are between one and seven years old, and about 20 printers that are more than seven years old. They have recently purchased 5 PCL printers that are also enabled to use the Internet Printing Protocol (IPP). Most of the print jobs that users generate are from word processing, graphics, or desktop publishing programs. These applications create mostly PCL and PostScript print jobs, which don't need to be transformed to another data stream to be printed on the existing printers.

The print administrator decides to create three kinds of actual destinations.

- Passthrough printers will send jobs to the PostScript and PCL printers.
- IPP printers will send jobs to the new IPP-enabled printers.
- BSD printers will send jobs to the oldest of the printers (the ones that still receive jobs using the lpr command).

Host system using PSF Direct

If you send print jobs to your printers from a host system using PSF Direct and you don't plan to allow job submission to those destinations through any other paths (such as the IPP gateway or a Windows gateway printer), use this table to see which type of destination fits your needs. This table assumes that a host PSF is being used to send IPDS data to Infoprint Manager. Use the Management Console to create these destinations.

Table 3. Host system using PSF Direct

If the data stream that you send to your printer is:	IPDS	PCL or PPDS ¹
Create this type of actual destination:	PSF Direct only (TCP/IP or channel)	PSF Direct only (non-IPDS) Note: This destination type is equivalent to a PSF command destination
Note::		
1. You can only print PPDS on a non-Windows system, for example, by sending the PPDS data stream to an AIX system.		

For example, a mail order clothing company stores all of its customer order information in a database on a mainframe. On the host system, the information is extracted and customer order sheets and shipping labels are generated. The order sheets and labels are then printed on a high-speed IPDS printer that is dedicated to these jobs. Because the print administrator wants to manage his print jobs and resources on the host system, he has chosen to use PSF Direct instead of MVS Download to submit jobs. PSF/MVS converts these jobs to IPDS and sends them to Infoprint Manager. Also, the printer is an older model and it is only set up to support channel attachment, so the administrator attaches it to the system using a parallel channel emulator card.

The print administrator uses the Management Console to create a PSF Direct only (channel) destination.

MVS Download, DPF, or a combination of PC-based applications and host systems

If you will be sending jobs to this destination from both PC-based applications and host systems (using PSF Direct, DPF, or MVS Download), use this table to see which type of destination fits your needs.

Table 4. MVS Download, DPF, or a combination of PC-based applications and host systems

If the data stream that your host system and applications send to Infoprint Manager is:	AFP, ASCII, DBCS ASCII, line data, PCL, PostScript, PDF, IPDS (via PSF Direct or DPF) ¹		PCL only	PS only
and the data stream that you send to this printer is:	IPDS	PCL or PPDS ²	PCL	PS
Create this type of actual destination:	PSF (TCP/IP or channel)	PSF (other driver or command)	PSF (other driver or command) OR any of the destination types listed under PC-based applications	One of the destination types listed under PC-based applications
Notes:				
1. PSF Direct and DPF can only submit jobs to PSF printers. MVS Download can submit jobs to any of the printer types listed.				
2. You can only print PPDS on a non-Windows system, for example, by sending the PPDS data stream to an AIX system				

For example, a film studio uses the same printers for all of their print jobs. During the day, employees in their business offices print reports, letters, and advertising materials. At night, they print royalty checks for their actors from information stored on their AS/400 host operating system using the PSF Direct function. By creating their print destination through the Infoprint Manager Administration GUI rather than the Management Console, the print administrator can ensure that the same destination can print jobs from both the LAN and the host.

PSF attachment types

PSF destinations are further divided by attachment type. Refer to the table below for information on choosing an attachment type.

Table 5. PSF attachment types

If you are sending this data stream to the printer:	IPDS	IPDS	PCL or PPDS	PCL or PPDS
--	------	------	-------------	-------------

Table 5. PSF attachment types (continued)

<p>and you are going to attach the printer to the system the Infoprint Manager is running on this way:</p>	<p>over the TCP/IP network</p>	<p>using the IBM 4159 Parallel Channel Emulator Card</p>	<p>over the TCP/IP network OR directly attaching it through a parallel or serial port</p>	<p>over the TCP/IP network using a command such as lpr to submit jobs to it</p>
<p>Select this attachment type:</p>	<p>TCP/IP</p>	<p>Channel</p>	<p>Other driver Note: You cannot use a PSF Other driver destination with PSF Direct</p>	<p>Command</p>

Chapter 6. Using the Internet Printing Protocol (IPP) with Infoprint Manager

The Internet Printing Protocol (IPP) is an application protocol that uses the Hypertext Transfer Protocol (HTTP) to allow you to print your documents to any IPP-enabled printer whose web address (or Uniform Resource Identifier-URI) you know. You can use IPP to send print data over both LANs and the Internet--so users who are directly connected to your LAN as well as those who work remotely have access to the same printers. Infoprint Manager for Windows provides two ways to configure your system to print using IPP.

Printing to an IPP-enabled printer

If you have an IPP-enabled printer, you can create an IPP printer just as you would any other actual destination--using the Create IPP Printer Wizard in the Infoprint Administration GUI. To print to the new destination, you can submit jobs using a print command such as `pdpr`, or you can install Infoprint Select on your workstation and create a Select printer that uses that destination.

Printing through the IPP Gateway

The IPP gateway is a service in Infoprint Manager that allows you to print using IPP even if you don't have an IPP-enabled printer, as long as you (and your users) have IPP client software installed on your workstation.

Note: Windows 2000 systems come with IPP client software built in. Users on other Windows platforms can download IPP client software. The Printer Working Group, the organization that is leading the development of IPP, provides a list of IPP client software that you can use to find an appropriate client at: <http://www.pwg.org/ipp/IPP-Products.html>

Once you have an IPP client installed, follow these steps to set up the gateway.

1. Enable the IPP Gateway using the Infoprint Manager Management Console. Refer to the Management Console online help for more information on completing this task.
 - a. The IPP Gateway runs as a Windows service called Infoprint Manager IPP Gateway. On Windows NT systems, the IPP Gateway service is set to start automatically by default. On Windows 2000 systems, the IPP Gateway is not set to start automatically by default. Refer to the Management Console online help for information on changing the default settings.
 - b. To start the IPP Gateway service while your system is running:
 - On Windows NT, select **Start** → **Settings** → **Control Panel**, and open the **Services** icon.
 - On Windows 2000, select **Setup** → **Administrative Tools** → **Services**, and click the **General** tab. Find the Infoprint Manager IPP Gateway service and start it.
2. Verify that the gateway is working. Open your Internet browser and enter the following web address, replacing *serverhostname* with the DNS hostname of the system that Infoprint Manager is installed on:
`http://serverhostname:631/printers`

Note: If you don't have a DNS name server running on your network, replace *serverhostname* with the dotted-decimal IP address of the system that Infoprint Manager is installed on instead.

If the gateway is working correctly, you will see a list of the printers that are available through the gateway. All of your Infoprint destinations should be listed. Find the address (URI) of the destination you want your users to print to.

3. . Refer your users to "Adding an Internet Printing Protocol (IPP) Gateway Printer to Your Desktop" on page 294 This step varies depending on which IPP client you are using; follow the instructions for adding a printer as provided by the IPP client. For the printer address, have your users enter the address that you found in step 2.

Chapter 7. Using Infoprint Manager notifications

Infoprint Manager can detect events (such as **job completed** or **destination needs attention**) as they occur, generate messages about those events, and deliver the messages to users who need the information. For example, users can receive messages about jobs they submit, operators can be notified about Infoprint objects that require their attention, and administrators can be notified if someone makes a configuration change. With notifications, you don't have to query for status of the objects in your system—you just receive messages when specific events occur so you can take care of them.

You can tailor who receives these messages and how they are delivered based on the **notification profile** that is set for a particular Infoprint object.

Understanding notification profiles

Most Infoprint objects (the server, actual destinations, logical destinations, queues, jobs, and default jobs) have **notification profiles** associated with them. A notification profile is composed of one or more *notification profile entries*. Each notification profile entry has three main components:

Event Identifiers

Which events cause messages to be sent. See “Event identifiers” on page 30 for more information.

Delivery method

How the messages are delivered. See “Delivery method and delivery address” on page 30 for more information.

Delivery address

Where the messages are sent. See “Delivery method and delivery address” on page 30 for more information.

In addition, a notification profile entry can include a comment (text that Infoprint will add to the general message text) and a locale (to set the language that the message will be sent in). Comments are optional. If you don't specify a locale, Infoprint delivers messages using the locale (language) that the user is running in.

Some objects need only one entry in their notification profiles. For example, when a person submits a job, he is probably the only one who needs to receive information about it, and he will probably be content to have all of the messages about that job delivered the same way.

For other objects, there are very good reasons to have multiple entries. For example:

- If operators manage printers from more than one workstation, you can create an entry for each workstation. That way, all of the messages go to all of the machines.
- If operators can address some messages (like **destination-needs-attention**), but an administrator needs to address others (like **object-deleted**), you can create an entry for each operator and one for the administrator. Messages are delivered to the person who can handle them.

- If an administrator likes to use Infoprint Manager Notifications while he's at work, and wants to keep a more permanent record of how often some events occur, he can set up two entries. One delivers messages to the Infoprint Manager Notifications client and one writes certain messages into a file that he can check periodically.

Event identifiers

The **event identifiers** component lists the Infoprint events that trigger notifications. You can find a full list of Infoprint notification events and their descriptions in Chapter 9 of the *Infoprint Manager: Reference*. In the descriptions, you will see what objects each event supports. If you add an event to an object that it does not support, no messages will ever be generated for that event. For example, you can add the **queue-backlogged** event to a notification profile entry for an actual destination, but **queue-backlogged** only supports the queue, job, and server objects. As a result, the actual destination notification profile will not generate any messages about that event.

That chapter also lists the Infoprint Manager *event classes*, or groups of related notification events. When you add event identifiers to a notification profile, you can add individual events, event classes, or a combination of the two.

Each type of Infoprint object has a default set of event identifiers. Refer to "Default notification profiles" on page 34 for a list of defaults for each object.

Delivery method and delivery address

The **delivery method** component shows how you want notification messages to be delivered. The **delivery method** and **delivery address** components are strongly related; the delivery method you choose determines the type of value you use for the delivery address.

In addition, delivery method and delivery address are the *keyed components* in notification profile entries; they are the two values that Infoprint Manager checks when you modify a notification profile. If the delivery method and delivery address that you specify match the delivery method and delivery address of an existing entry, Infoprint Manager updates that entry. If both values don't match any of the existing entries, Infoprint Manager adds a new entry to the profile.

There are various delivery methods defined for Infoprint objects. All methods are valid for all objects **except** for **job-log**; **job-log** is only valid for jobs. Table 6 on page 31 lists the delivery methods, along with descriptions and how to set the delivery address for each type.¹

1. Two other delivery methods, **sapcb** and **sapconnect** exist, but they are only used internally in conjunction with the Infoprint Manager SAP Print Feature. You should never use either of those values in a notification profile entry.

Table 6. Infoprint notification delivery method characteristics

Delivery Method	Description	Delivery address
<p>message</p> <p>Message is the default delivery method for all objects.</p>	<p>Sends a message to the Infoprint Manager notification server, where it waits for an Infoprint Manager Notifications client to pick it up. Once a message is picked up, it is deleted from the notification server; there are no further records of those messages.</p>	<p>The delivery address listed in the notification profile can be whatever you want it to be– a user id, office number, IP address, driver’s license number– as long as it is unique. In order to receive messages sent to that address, you must set the delivery address listed in the Settings dialog in the Infoprint Manager Notification client to the same value. The Infoprint Manager Notifications client will only pick up messages that specify an identical delivery address.</p>
<p>electronic-mail (synonyms: e-mail, email)</p>	<p>Sends an E-mail to the address listed in the delivery address, OR, for jobs submitted using Infoprint Select, to the Select message window. Refer to “Infoprint Select and notifications” on page 32 for more information.</p> <p>To use the e-mail delivery method, you must also set the smtp-server-host attribute on your Infoprint Manager server to the IP address or DNS hostname of the system that your SMTP mail server runs on.</p>	<p>The E-mail address that you want notification messages to be sent to. For example, kjones@printerco.com. If you submit jobs using Infoprint Select, this value is set for you.</p>
<p>exit</p>	<p>Doesn’t actually send a message. Instead, when the event specified in the event identifier occurs, Infoprint Manager runs an exit program or script. Refer to “Using the ‘exit’ delivery method” on page 33 for more information.</p>	<p>The full path to the notification exit program or script (with hard-coded arguments, if any).</p>

Table 6. Infoprint notification delivery method characteristics (continued)

Delivery Method	Description	Delivery address
delivery-method-wireless	Sends notifications to wireless devices, such as pagers, cellular phones, palm pilots, and laptop computers.	Because wireless notifications are delivered as emails, you must set both the smtp-server-host and the smtp-server-port server attributes on this Infoprint server. If your Infoprint server resides behind a firewall, you may have to configure your SMTP server. For information on this task, refer to <i>Configuring and Using Infoprint Email</i> on the web at http://www.printers.ibm.com/R5PSC.NSF/Web/41aixadmin . Note that this URL has been divided so it prints correctly in the text.
file	Writes the message in the file you specify in the delivery address. If the file doesn't exist, Infoprint Manager will create it. If the file does exist, Infoprint Manager will overwrite it the first time it writes a message. After that, messages are added to the end of the file.	The directory path and name of the file that you want to write your notification messages in. This file must reside on the same system as your Infoprint Manager server.
file-add-to	Adds the message to the end of the file you specify in the delivery address. If the file doesn't exist, Infoprint Manager will create it.	The directory path and name of the file that you want to write your notification messages in. This file must reside on the same system as your Infoprint Manager server.
job-log Only valid for jobs.	Writes messages into the job-log attribute of the job. To see the messages, issue the pdl s command for the job-log attribute on the job you want to know about.	No value needed.
none	Does not send any notifications; turns notification off.	No value needed.

Infoprint Select and notifications

When you submit a print job using Infoprint Select, a default notification profile is sent with the job. That notification profile sets the event-identifiers to a default set of events and the delivery method to either **e-mail** (if you have the Select message window running) or to **none** (if you don't have it running). The Infoprint Select default notification profile overrides any notification settings you configure in a default job.

If you don't want to use the Select default notification profile, use the appropriate procedure below.

Windows NT

1. On the Windows system that Infoprint Select is installed on, right click on **My Computer** and select **Properties**.
2. In the System Properties window, click the **Environment** tab.
3. Click one of the variables listed under **System Variables** to signal that you want to create a system variable. The variable and its value will appear in the fields at the bottom of the dialog.
4. Clear the **Variable** and **Value** fields.
5. In the **Variable** field, type: `pdnotify` and in the **Value** field type: `override_select`.
6. Click **Set**.
7. Verify that `pdnotify` appears in the list of **System Variables** and not in the **User Variables**. If it appears in the list of User variables, delete it and start again from step 3.
8. Click **OK**.
9. Restart your system to make the changes take effect.

Windows 2000

1. On the Windows system that Infoprint Select is installed on, right click on **My Computer** and select **Properties**.
2. In the System Properties window, click the **Advanced** tab.
3. On the Advanced tab, click **Environment Variables**.
4. In the **System variables** box on the **Environment Variables** dialog, click **New**.
5. In the **Variable Name** field, type `pdnotify` and in the **Variable Value** field, type `override_select`.
6. Click **OK**.
7. Verify that `pdnotify` appears in the list of **System variables** and not in the **User variables**. If it appears in the list of User variables, delete it and start again from step 4.
8. Click **OK** in the **Environment Variables** dialog.
9. Click **OK** in the System Properties window.
10. Restart your system to make the changes take effect.

Using the 'exit' delivery method

The **exit** delivery method allows you to set up automatic responses to system events beyond just sending a message. When you set the delivery method for a given event to **exit**, you use the delivery address to specify the explicit path to a script or program on your system. Whenever that event occurs, Infoprint Manager runs the script or program you specify.

For example, if the Infoprint Manager server runs low on disk space, the **disk-space-low** event generates a message for the system administrator. The administrator then has to do something to clean up the disk so Infoprint Manager can continue to run effectively. However, using the **exit** delivery method can automate that process by launching a "disk clean up" script. The administrator can still receive the message, but will know that the script is already working on the problem.

If you are using the Print Optimizer DSS, Infoprint Manager provides a sample script called **poexitcleanup** that you can use with the **exit** delivery method. The

script demonstrates one way that you can automate cleaning up disk space in the Print Optimizer resource filesystem after the **disk-space-low** event occurs.

By default, when you use the exit delivery method, Infoprint Manager only passes two values along to the script or program: the message number of the event that occurred and the text of the message. As a result, the command line that the exit delivery method executes is composed of the path that you specified in the delivery address, the last three digits of the message number, and the text of the message.

Therefore, if you create a notification profile entry for the **job-completed** event, setting the delivery method to **exit** and the delivery address to `c:\Infoprint\exits\myscript`, when a job finishes printing, Infoprint Manager will generate and execute a command line similar to this one:

```
c:\Infoprint\exits\myscript 280 "5010-280 Finished processing job ofc,
Job 20 (3520800004)."
```

The scripts or programs you write may require other information, such as the destination name or the printer model. To pass those values, you can include them in the delivery address of the notification profile entry; they will be added to the end of the command line when Infoprint Manager creates it.

Default notification profiles

By default, the person who creates an object (server, destination, queue, job) receives the notification messages about that object. The default notification profile settings provide information such as object status, error conditions, or configuration changes. Some of these events require the person managing the object to intervene; for example, someone may need to clear a paper jam.

By default, the events listed in the following table generate messages for the corresponding objects. These messages are delivered to the user who created the object: job submitters receive messages about their print jobs; the administrator receives messages about destinations and servers that he or she created. The default delivery method is **message**, so the notifications are sent to the Infoprint Manager Notification client on the system that the person used to create the object.

Table 7. Default notification profile settings

Object	Events
Actual destination	destination-needs-administrator destination-needs-attention destination-needs-operator destination-timed-out object-cleaned object-deleted
Default Job	No defaults set. Refer to "Using notification profiles with default jobs" on page 35 for more information.

Table 7. Default notification profile settings (continued)

Job	destination-needs-attention destination-needs-operator destinations-not-ready-for-job document-aborted-by-destination document-aborted-by-server document-cancelled-at-destination job-aborted-by-server job-cancelled-by-operator job-cannot-be-scheduled job-completed past-discard-time
Logical destination	object-cleaned object-deleted
Queue	object-cleaned object-deleted queue-backlogged
Server	internal-server-error low-on-memory memory-exhausted object-cleaned object-deleted out-of-disk-space out-of-dpf-haids out-of-dpf-job-numbers out-of-dpf-raids sapcbd-down sapconnect-down server-shutdown-complete

Determining how much notification administrators, operators, and job submitters require takes some planning and, as you familiarize yourself with Infoprint, you should discover when you require more or less notification. You can start by using the defaults and see how many and what kind of messages you receive. If you find that you need more or less notification, or if you want to modify which users receive notifications, or if you want messages to be sent using a different delivery method, you can change the notification profiles.

Using notification profiles with default jobs

When someone submits a print job, Infoprint Manager sends messages about that job based on a default notification profile. As a result, the delivery method is **message**, the delivery address is "the user ID of the person who submitted the job @ the address of the system the job was submitted from," and the event identifiers are those listed in Table 7 on page 34. (If you submit jobs using Infoprint Select, the default notification profile is different. Refer to "Infoprint Select and notifications" on page 32 for additional information.)

If you want to change the settings in the default notification profile, you can create a notification profile on a default job. There are several things to keep in mind when you set up a notification profile on a default job:

- You can only set one default job on each logical destination. As a result, all of the jobs will use the same notification profile.
- If you set a delivery address in the notification profile for the default job, all messages will go to that address. However, if you leave the delivery address blank, Infoprint Manager will fill it in with the address that the job was

submitted from (you can't specify this address; it is sent with the job). This option only works with the **message** and **e-mail** delivery methods and is the only time that the notification profiles are merged.

- If you submit a job with an explicit notification profile (for example, by including the **notification-profile** attribute when you submit a job using **pdpr**), that profile overrides the one set in the default job.

Note: If you submit jobs using Infoprint Select, an explicit notification profile is sent with every job. If you want to use a notification profile on a default job with Infoprint Select, refer to “Infoprint Select and notifications” on page 32 for information about overriding Select notifications.

Changing notification profiles

For jobs, default jobs, servers, queues, and actual destinations

You can use the properties notebooks in the Infoprint Manager Administration GUI to change notification profiles for jobs, default jobs, servers, queues, and actual destinations. Open the properties notebook and click the **Notification** tab for an actual destination, a job, or a default job. For a server, click the **Other** tab. On the page that opens, you can see the notification profile entries that already exist, including:

- How messages about that object are delivered
- Who they are sent to
- What events generate messages
- Any comments that are included with the messages

Use the **Add**, **Modify**, and **Delete** buttons to make changes to the notification profile. Refer to the online help for the Infoprint Manager Administration GUI for additional information. You can also change notification for jobs in the Infoprint Manager Operations GUI.

Note: If you change the delivery method, make sure you change the delivery address appropriately.

For logical destinations

The only way to change a notification profile for a logical destination is to use the **pdset** command. The procedures below provide examples of some common modifications you may want to make. You can use these procedures to change the notification profiles for other Infoprint objects as well, but you may find it easier to use the properties notebooks in the Infoprint Manager Administration GUI.

This procedure describes how to modify notification profiles by:

- “Adding or modifying events or users in a notification profile” on page 37
- “Removing users from a notification profile” on page 39
- “Changing the delivery method” on page 40

If you need to change other parts of a notification profile, refer to the *Infoprint Manager: Reference* for information on the **pdset** command and the **notification-profile** attribute.

Important: When you modify a notification profile, Infoprint Manager compares the new delivery method and delivery address to the delivery methods and delivery addresses in the existing entries and behaves accordingly:

- If you specify both of the values and both match the values in an existing entry, the modification replaces that entry.
- If you only specify one of the values, Infoprint Manager fills in the other value with the default values. Then, it checks to see if any of the existing entries match that combination. If they match, Infoprint Manager replaces that entry. If they don't match, Infoprint Manager creates a new entry.
- If you specify both values but only one matches, Infoprint Manager creates a new entry.
- If you specify both values and neither one matches, Infoprint Manager creates a new entry.

When you delete a notification profile entry, the delivery method and delivery address in your command must match those in an existing entry exactly.

Viewing an existing notification profile

If you want to see what values are set for a notification profile, follow these steps.

1. Open a DOS window on the computer that your Infoprint Manager server is installed on. On Windows NT systems, click the Windows **Start** button and select **Program** —> **Command Prompt**. On Windows 2000 systems, click the Windows **Start** button and select **Program** —> **Accessories** —> **Command Prompt**.

2. Type the following command. Fill in the values for the italicized items as defined below.

```
pdls -c Type of Infoprint object -r notification-profile Objectname
```

Type of Infoprint object

What type of object you want to perform this command on, not the specific object. For example, type **destination** not **printer1-ld**.

Objectname

The name of the destination or queue that this notification profile refers to.

3. Press **Enter**.

The notification profile will be displayed.

Note: If you have messages about a particular object sent to more than one person, the notification profile will display multiple entries, one for each person.

Adding or modifying events or users in a notification profile

Follow these steps to add or remove events in the notification profile for a particular delivery address, or to add another entry to the notification profile, so that messages will be sent to another address as well.

1. Open a DOS window on the computer that your Infoprint Manager server is installed on. On Windows NT systems, click the Windows **Start** button and select **Program** —> **Command Prompt**. On Windows 2000 systems, click the Windows **Start** button and select **Program** —> **Accessories** —> **Command Prompt**.
2. Type the following command. Fill in the values for the italicized items as defined below.


```
pdset -c Type of Infoprint object -x "notification-profile+={event-identifiers= event(s) delivery-method= method delivery-address="address"}" Objectname
```

Type of Infoprint object

What type of object you want to perform this command on, not the specific object. Type **destination**, **queue**, **job**, or **server**.

event-identifiers

The events that cause Infoprint Manager to send a message. You can have messages sent for any of the events that are listed in the events-supported attribute for the server. Refer to the *Infoprint Manager: Reference*- chapter 7, section *Attributes for Servers: events-supported*, for a list of these values. Not all events support all objects. Refer to chapter 9 in the *Infoprint Manager: Reference* for lists of what objects each event supports.

Note: You must type all of the events for this object that you want the user you specify in the delivery address to receive, even if they already appear in the notification profile. Any events that you do not type will not be listed in the notification profile.

delivery-method

How you want messages to be sent. Refer to Table 6 on page 31 for information about the various delivery methods.

delivery-address

Where you want to have messages sent. Refer to Table 6 on page 31 for information about what kinds of delivery address value to use with each delivery method. Enclose the value in double quotes.

Note: You cannot specify more than one delivery address here. If you want another user to receive messages about the same (or different) events, finish this procedure, then enter the command again with that user's delivery address.

Objectname

The name of the destination or queue that this notification profile refers to.

3. Press **Enter**.
4. Repeat the process for other users and objects.

Example: The default notification profile for queues is set to send you messages when the **queue-backlogged**, **object-cleaned**, and **object-deleted** events occur. You need to change the notification profile for the print queue **printer1-q** so that you receive messages when the **queue-no-longer-backlogged** and **queue-state-changed** events occur as well. Your backup person does not currently receive any messages about queues, but needs to receive all of the same messages you do.

Issue this command to change your notification profile. Remember to list the event-identifiers for **all** of the events that you want this user to receive notifications about. This list of event-identifiers replaces the existing list entirely.

```
pdset -c queue -x "notification-profile+={event-identifiers=object-cleaned
object-deleted queue-backlogged queue-no-longer-backlogged
queue-state-changed delivery-address="administrator@mydesk.office.com"}"
printer1-q
```


Then, issue this command to have the **same** queue messages sent to your backup person.

```
pdset -c queue -x "notification-profile+={event-identifiers=object-cleaned
object-deleted queue-backlogged queue-no-longer-backlogged
queue-state-changed delivery-address="backup@hisdesk.office.com"}"
printer1-q
```

Use the “Viewing an existing notification profile” on page 37 procedure to make sure the changes took effect.

Removing users from a notification profile

Follow these steps to remove an entire entry from a notification profile.

1. Open a DOS window on the computer that your Infoprint Manager server is installed on. On Windows NT systems, click the Windows **Start** button and select **Program** → **Command Prompt**. On Windows 2000 systems, click the Windows **Start** button and select **Program** → **Accessories** → **Command Prompt**.
2. Use the “Viewing an existing notification profile” on page 37 procedure to find the entry you want to remove.
3. Type the following command. Type the correct **delivery-method** and the **delivery-address** exactly as it is displayed for the entry that you want to remove.

```
pdset -c Type of Infoprint object -x "notification-profile-={delivery-
method=electronic-mail or message delivery-address="address"}"Objectname
```

4. Press **Enter**.
5. Use the “Viewing an existing notification profile” on page 37 procedure to verify that the entry has been removed.
6. Repeat the process for other entries.

Example: You and two other people currently receive messages about the logical destination **printer1-ld**. You don’t want to receive them any longer. You view the notification profile, and see these three entries:

```
printer1-ld: notification-profile=

{event-identifiers = object-cleaned object-deleted delivery-method = message
delivery-address = "admin@desk1.office.com" locale = en_US}

{event-identifiers = object-cleaned object-deleted delivery-method = message
delivery-address = "helpdesk@desk2.office.com" locale = en_US}

{event-identifiers = object-cleaned object-deleted delivery-method = message
delivery-address = "backup@desk3.office.com" locale = en_US}
```

Issue this command to remove your entry from the notification profile.

```
pdset -c destination -x "notification-profile-={delivery-method=message
delivery-address = "admin@desk1.office.com"}" printer1-ld
```

Note: If you display the notification profile to verify that the entry has been removed and find that it has not been removed, issue the command again, but this time instead of only specifying the **delivery-address**, type the entire notification profile exactly as it appears.

Changing the delivery method

You can choose to receive your notification messages in a variety of ways, as listed in Table 6 on page 31. Follow these steps to change the delivery method from the default to some other value.

1. Open a DOS window on the computer that your Infoprint Manager server is installed on. On Windows NT systems, click the Windows **Start** button and select **Program** → **Command Prompt**. On Windows 2000 systems, click the Windows **Start** button and select **Program** → **Accessories** → **Command Prompt**.
2. Use the "Viewing an existing notification profile" on page 37 procedure to find the entry you want to change.
3. Type the following command. Fill in the values for the italicized items exactly as they appear in the entry you want to change. Replace the **delivery-method** and **delivery-address** values as specified in Table 6 on page 31.

```
pdset -c Type of Infoprint object -x "notification-profile+={event-identifiers=event(s) delivery-method=method delivery-address="address" locale=yourlocale}"Objectname
```
4. Press **Enter**.
5. Use the "Viewing an existing notification profile" on page 37 procedure to verify that the entry was added.
6. Follow the procedure for removing a user to remove the old entry.

Example: You want to receive your messages about **printer1-q** in your E-mail instead of through Infoprint Manager Notifications. You view the notification profile, and see this entry:

```
printer1-q: notification-profile={event-identifiers = object-cleaned  
object-deleted queue-backlogged delivery-method = message delivery-address  
= "admin@desk1.office.com" locale = en_US}
```

Issue this command to add another entry to the notification profile with a different delivery method and delivery address.

```
pdset -c queue -x "notification-profile+={event-identifiers=object-cleaned  
object-deleted queue-backlogged delivery-method=electronic-mail  
delivery-address="admin@mycompany.com" locale=en_US}" printer1-q
```

Then view the notification profile again and see two entries:

```
printer1-q: notification-profile= {event-identifiers = object-cleaned  
object-deleted queue-backlogged delivery-method = message delivery-address  
= "admin@desk1.office.com" locale = en_US} {event-identifiers =  
object-cleaned object-deleted queue-backlogged delivery-method =  
electronic-mail delivery-address = "admin@mycompany.com" locale = en_US}
```

Follow the procedure for removing a user to delete the entry that would send messages to Infoprint Manager Notifications.

Getting help for notification messages

If you receive a notification message that you don't understand, Infoprint Manager provides a way to get more information about the situation. In some cases, the explanation even suggests some things you can do to resolve the problem.

To see the full text of an error message, open the Management Console and select **Help**—> **Message Help**. Type the message code in the fields at the top of the Message Help dialog, and click **OK**.

Other notification methods

Notify-operator attribute

The **notify-operator** attribute is a scaled-down version of a notification profile. It can only be set on Infoprint Manager queues and actual destinations and can only use the **e-mail**, **message**, **exit** and **wireless** delivery methods. In addition, it only sends messages when one of the following events occur.

1. A job is added to a queue.
2. An actual destination starts processing a job.
3. An actual destination finishes processing a job.

The syntax of the **notify-operator** attribute is:

delivery-method:delivery-address

Refer to Table 6 on page 31 for descriptions of the e-mail and message delivery methods and their corresponding delivery addresses. By default, the notify-operator attribute is set to **message:your user ID@your node** when you create an actual destination or a queue. To change the value for an actual destination, you can use the **Notification** tab of the properties notebook. To change the value for a queue, use the **pdset** command.

The messages that the notify-operator attribute delivers are not standard error messages like the ones you receive when you use a full notification profile. Instead, you write the messages that you want to send and specify them as attributes of jobs when you submit them.

Note: You must either specify those attributes in the command when you submit the job or set the text in the job and document defaults for the appropriate logical destination.

The job attributes that you use to specify these messages are the following.

- **job-message-to-operator**
The message that Infoprint Manager sends when it adds a job to the queue.
- **job-start-message**
The message that Infoprint Manager sends when the job starts processing.
- **job-end-message**
The message that Infoprint Manager sends when the job finishes processing.

You can type the text for all three of these messages in the appropriate field on the **Job Notification** tab of the properties notebook for a job or a default job.

In addition, you can use the **job-start-message** attribute in conjunction with the **job-start-wait** attribute. If you set the **job-start-wait** attribute on a job (or in a default job) to **true**, the actual destination receives the job and sends the job-start message. Then, it pauses the actual destination (whether or not a job is currently being printed) until the operator intervenes. This function is useful when different jobs need to be printed on different types of paper. Operators don't need to manually pause the actual destination to change paper between jobs; Infoprint Manager does it automatically. Once the printer is ready, the operator must resume

| the actual destination and the job prints. This function should be set for the first in
| a series of jobs, once you are certain that there are no jobs printing on the printer.
| Because it compels Infoprint Manager to pause the printer for every job with
| `job-start-wait=true`, you should not specify this value on the default job.

To use the **job-start-wait** attribute, you must set it to **yes** on the job or default job. You can specify the attribute in the command when you submit a print job or set it on the **Job Notification** tab of the properties notebook for default jobs.

Chapter 8. Managing security

Infoprint Manager security (sometimes called *non-DCE security*) is a feature that you administer through the Infoprint Manager Management Console. It allows you to protect your printing system by associating an Access Control List (ACL) with an Infoprint object or operation. An *ACL* is the list of users and groups who have permission to do something to or with an object, and what type of permission that is.

Important: Infoprint Manager Security references the login ID of the current user to determine who is attempting to perform an action. Otherwise, the Infoprint Manager security system is entirely separate from any Windows security you have set. For example, you cannot use groups that you create with the Windows User Manager in Infoprint Manager Security- you must create security groups using the Infoprint Manager Management Console.

If you do not want to use Infoprint Manager Security, you can turn it off using the server properties notebook in the Infoprint Manager Administration GUI by setting the **Security level** for your server to **none**.

Types of permission

In Infoprint Manager, users can have three levels of permission: **read**, **write**, and **delete**. The levels provide the following types of access.

- **Read-** For operations, the user can perform the operation. For servers and queues, the user can view the attributes. For destinations, the user can view attributes and submit jobs to that destination.
- **Write-** For all objects, the user can modify attributes.
- **Delete-** For all objects, the user can delete the object.

Note: The levels of permission are not cumulative. If you give a user **delete** permission only, he will **not** automatically have **read** and **write** permissions. Be sure to mark all of the levels of permission that the user needs. For more information, see "Infoprint Manager Security Permissions" on page 46.

If you place userA on the ACL for the logical destination "print2ld" and give her **read** permission, she can send her print jobs to it and can open the "print2ld" object to see its properties. However, she cannot make changes to those properties. If she tried to change any of them or tried to delete the destination, she would receive an error message. If you decide that userA needs to be able to do more and give her **write** permission as well, she will be able to change the properties of "print2ld," but still will not be able to delete it.

Important: If you protect a destination (logical or actual) so that only certain users can modify or delete its properties, you may inadvertently prevent other users from submitting print jobs to it. To be sure that all of your users can still print to the destination, add the wildcard character (*) to the ACL as a user with read permission.

You can also attach ACLs to the operations that you can perform on Infoprint objects. Allowing you to protect both operations and objects means that Infoprint Manager Security provides different levels of security: you can protect all objects by using ACLs at the operation level or you can protect individual objects with ACLs applied only to them. Or you can do both: protect all objects by using operation-level ACLs for some operations, and limit access to subsets of objects by using object-level ACLs.

For operations, there is only one level of permission: **read**. If a user has **read** permission, he can perform that action; if he does not, he cannot perform the action. For example, userB is a printer operator and must be able to move jobs to different positions in the print queue because some jobs need to be printed before others. You can give userB **read** permission for the operation **Reorder Job** to allow him to do his job. On the other hand, userC submits print jobs from his office workstation and doesn't like to wait for the jobs ahead of his in the queue to print. If you want to prevent him from moving jobs, don't put him on the ACL for the **Reorder Job** operation. When he tries to move his job to the top of the queue, the action will be denied.

When you install Infoprint Manager, many operations are already protected so that only members of the **admin** and **oper** groups can perform them. You can see the ACLs for operations in the Management Console by selecting the **Security-ACL-Operations** item in the left pane. If you want users to be able to perform those operations, you must either add those users to the individual ACLs or to a group that has permission (either the existing **admin** and **oper** groups or a new group that you create).

Note: If an object is protected, a user can only perform an operation on that object if he has both **read** permission for the operation and the appropriate level of permission for the object (for example, a job or a printer).

- If the object is not protected, any user with **read** permission on an operation can perform that operation.
- If the object is protected, the permission needed depends on the operation. For example: **List** requires **read** permission on the object, **Set** requires **write** permission, and **Delete** requires **delete** permission.

By default, Infoprint objects (destinations, queues, servers) are not protected, members of the **admin** group have read permission on all operations, members of the **oper** group have read permission on most operations, and **all** users have read permission on five operations. Those five operations are:

- **List/Query** (all objects)
- **Print**
- **Modify job**
- **Query job**
- **Remove job** (delete job)

However, users who are not members of the **admin** and **oper** groups can only modify and remove jobs that they submitted. In addition, if the ACL for the **Reorder job** action is changed so that everyone can use it, users who are not members of the **admin** and **oper** groups will only be able to reorder jobs that they submitted. By default, members of the **admin** and **oper** groups can perform all six of those operations on all jobs.

Note: If you decide to protect your queues, all users will still be able to perform the tasks listed above on their own jobs. Users who are on the ACL for a queue will be able to perform those tasks on all jobs in that queue.

Security groups

No matter what size organization you work in, manually adding every user to every ACL can be a time-consuming process. To reduce some of the work, you can create *security groups*, groups of users who need to have the same levels of permission for the same objects. You use the name of the security group like a user ID; instead of adding each user ID to an ACL, you add the group name. For example, if you want all of your help desk operators to be able to perform the same operations, create a group and name it **helpdesk**. Then, add **helpdesk** to the appropriate ACLs.

Important: User IDs and group names are case sensitive.

When you install Infoprint Manager, three security groups are created by default:

- **acl_admin**- Users who have authority to manage security by changing access control lists and groups. The default members are **Administrator@*** and the user who was logged on when Infoprint Manager was installed (for example, **myuserid@***).
- **admin**- Users who have administrator authority. The default members are **Administrator@*** and the user who was logged on when Infoprint Manager was installed (for example, **myuserid@***).
- **oper**- Users who have operator authority. The default member is **Administrator@***.

Notes:

1. You can modify these groups as needed. In the example above, you could have simply added the help desk operators to the default **oper** group and modified any permissions that weren't set to the level that you wanted them.
2. The default group members contain the wildcard character (*) for greater flexibility. See below for more information on wildcarding. If you don't want the Administrator user on other systems to be able to administer Infoprint Manager, replace the * with the explicit address of the system that the Infoprint Manager server is installed on, for example **Administrator@serversystem.company.com**.

You can add users to multiple groups, but you cannot make one group a member of another group. For example, if you hire five new print operators, you might create a group for them called **trainees**, since you only want them to have limited permissions until they are finished with their training. When they finish their training, you cannot add **trainees** as a member of the **operators** group. You will have to add their user IDs to the operators group one at a time. In addition, you will have to either delete the **trainees** group or delete the members from it—otherwise those users will have conflicting levels of permission.

When users are members of more than one group and each group has a different level of permission for a particular object, the most restrictive permission applies. In the example above, if you forgot to remove the new employees from the **trainees** group at the end of their training, they wouldn't be able to perform the tasks their job required- they would still be restricted.

Identifying users and groups: wildcarding

When you add users to ACLs or Security groups, you identify them by their user ID and the computer that they work on in this format: *username@computername*. The permissions you assign will only apply when that person accesses Infoprint Manager from that workstation.

Note: The *computername* that you specify must be the fully-qualified name of the workstation, for example: *joescomputer.boulder.ibm.com*

If, however, you or any of your users want to be able to work with Infoprint Manager objects from various workstations, you may not want to add multiple user ID/computer name combinations for the same person— you can use wildcarding instead. When you use wildcarding, you replace the computer name or user ID with the wildcard character (*) when you add a member to an ACL or Security Group. The wildcard character stands for “any computer” or “any user ID.” So, if you create an ACL member called **Administrator***, no matter what computer you log on to as **Administrator**, you have the same permissions. If you use the wildcard character before the computer name, for example ***@computer1**, any user who logs on to computer1 can perform the actions that the ACL member has permission for.

Working with ACLs and groups

Use the Infoprint Manager Management Console to manage the security of your print system. Open the **Security** folder in the left pane of the Management Console to display the Access Control Lists folder and Groups object. Open the Access Control Lists folder to see the types of ACLs that exist. Click on the type of ACL (**Operations**, **Destinations**, **Servers**, or **Queues**) to display the ACLs that exist. Click on **Groups** to see the groups that currently exist and their members.

For more detailed information on using the Management Console to work with groups and ACLs, refer to the online help (in the Management Console, click **Help** -> **Contents**, and look on the **Contents** tab under **Managing Security**).

Infoprint Manager Security Permissions

In Infoprint Manager Security, you can grant users three types of permission: **read**, **write**, and **delete**. The following table identifies what happens when you **deny** one of these permission levels to a user.

Note: In FST security (the security you administer through the Management Console), permission levels are not cumulative. If you grant a user **delete** permission for an object, she doesn't automatically have **read** and **write** permission; you must grant her those permissions as well.

Table 8. Effect of denying permission levels

	Read Permission Denied	Write Permission Denied ¹	Delete Permission Denied ¹
Server	<ul style="list-style-type: none"> • Cannot list object ⁴ • No effect on contained objects' permissions • No effect on job submission 	<ul style="list-style-type: none"> • Cannot modify object ² • Cannot create object within • No effect on contained objects' permissions • No effect on job submission 	<ul style="list-style-type: none"> • Cannot delete object • No effect on contained objects' permissions • No effect on job submission
Queue	<ul style="list-style-type: none"> • Cannot list object ⁴ • No effect on job submission 	<ul style="list-style-type: none"> • Cannot modify object ² • No effect on job submission 	<ul style="list-style-type: none"> • Cannot delete object • No effect on job submission
Logical Destination	<ul style="list-style-type: none"> • Cannot list object ⁴ • Cannot submit print to this object (directly, or indirectly through an AD) ³ 	<ul style="list-style-type: none"> • Cannot modify object ² 	<ul style="list-style-type: none"> • Cannot delete object
Actual Destination	<ul style="list-style-type: none"> • Cannot list object ⁴ • Cannot submit direct-print to this object (Note: May still submit to LD feeding this AD) 	<ul style="list-style-type: none"> • Cannot modify object ² 	<ul style="list-style-type: none"> • Cannot delete object

Notes:

1. In FST security, permissions are independent of each other. For example, you can grant someone **write** permission without granting him or her **read** permission. In contrast, DCE permissions are additive, so **write** permission implies that the user has both **read** and **write** permissions.
2. **Modify** includes set, enable/disable, pause/resume.
3. In DCE security, this check is only performed if the LD's **authorize-jobs** attribute is set to **true**. The **authorize-jobs** attribute is not used in FST security.
4. For FST security only. DCE always allows you to list the object .

Chapter 9. Administering your Infoprint Manager server using a Windows Terminal Server client

While you can perform most administrative tasks using the Infoprint Manager Administration GUI, there are a few that you must perform using the Management Console. However, since the Management Console must reside on the system that your Infoprint Manager server is installed on, performing those tasks can be inconvenient. The only way to perform administrative tasks on the Management Console on a different system is to use a Windows Terminal Server.

- If you use Windows NT, you must install Microsoft Windows NT Terminal Server as the operating system on the system that you install your Infoprint Manager server on.
- If you use Windows 2000, Terminal Server capabilities are included in Windows 2000 Server. Therefore, you must install your Infoprint Manager server on a Windows 2000 Server system, not a Windows 2000 Professional system.

Once you have Infoprint Manager installed on the correct version of Windows, you need to install the Terminal Server Client on any other systems that you want to be able to access the Management Console from. You can install the client on Windows 98, Windows NT, and Windows 2000 systems, but the process is slightly different based on the system that your Infoprint Manager server is installed on (Windows NT Terminal Server or Windows 2000 Server). Use the following procedure to create Terminal Server Client installation disks and install the client.

Installing the Windows Terminal Server client

To complete this procedure, you will need:

- One blank floppy disk if you are using Windows NT Terminal Server or
 - Two blank floppy disks if you are using Windows 2000 Server.
1. Log on to the system that your Infoprint Manager server is installed on.
 2. On Windows NT Terminal Server, click **Start** → **Programs** → **Administrative Tools** → **Terminal Server Client Creator**.
On Windows 2000 Server, click **Start** → **Programs** → **Administrative Tools** → **Terminal Services Client Creator**.
 3. In the dialog that opens:
 - On Windows NT Terminal Server, select **Terminal Server Client for Windows 95/NT Intel** and check the **Format disk(s)** check box. Click **OK**.
 - On Windows 2000 Server, select **Terminal Services for 32-bit x86 Windows** and check the **Format disk(s)** check box. Click **OK**.
 4. Follow the instructions in the pop up dialogs to create your Terminal Services Client installation disks.

Note: Save these disks. You will need them if you ever need to uninstall the client.

5. Close the dialog once the disks are created.
6. Log on to the machine that you want to install the Terminal Services Client on.
7. Insert the first Terminal Services Client installation disk in the drive.

8. Navigate to that disk drive and double click the **Setup.exe** icon to start the **Terminal Server/Services Client Setup** wizard.
9. Follow the instructions in the wizard to install the client.
When you finish the wizard, Terminal Server/Services Client will be added to your **Start** menu.
10. Start the client:
 - On Windows NT Terminal Server, select **Start —> Programs —> Terminal Services Client —> Terminal Server Client**.
 - On Windows 2000 Server, select **Start —> Programs —> Terminal Services Client —> Terminal Services Client**.
11. Find the name of the server that your Infoprint Manager server is installed on and connect to it.
Once you are connected, you can open the Management Console and use it just as you would on the server system itself.

Chapter 10. Changing the server hostname and IP address

Use this procedure to change the hostname of the system that your Infoprint Manager server runs on.

Notes:

1. If you have changed the hostname/IP address on the system and already restarted the system, the Infoprint Manager service will not be able to start and your Infoprint GUIs will not function properly. Complete the “Changing the server hostname and IP address” task to correct the problem.
2. If the system that your Infoprint Manager server runs on is part of a domain, remove it from the domain before you perform this procedure. Complete the “Changing the server hostname and IP address” task before you add the system to the domain again.
3. **For multi-server environments:**
 - If you are changing the hostname/IP address on the namespace system, you must stop all of the non-namespace servers and un-map the network drives all of the network drives that you have mapped to the namespace system before you start the procedure below. After you complete the procedure below, re-map the network drives. Then, use the Management Console (**Edit -> Service Configuration**) on each of the non-namespace servers to change the **Base namespace path** and **Remote namespace folder** to the new hostname and restart the non-namespace servers.
 - If you are changing the hostname of a Infoprint Manager non-namespace server (running on Windows NT or Windows 2000) that shares a namespace with Infoprint Manager for AIX (with the namespace on an AIX system), follow the “Changing the hostname of a non-namespace server” on page 52 procedure below.

Changing the server hostname and IP address

1. If your Infoprint Manager server is running, stop it by selecting **File -> Stop Server** in the Management Console.

Note: If you changed the hostname on the system and already restarted the system, your Infoprint Manager server will already be stopped.

2. If you use Infoprint Manager Security and have changed the member names of the **acl_admin** group so they include the hostname, follow these steps:
 - In the left pane of the Management Console, open the **Security** folder and click the **Groups** item.
 - In the right pane, check to see if any of the member names in the **acl_admin** group include the hostname.
 - Edit any user IDs in the **acl_admin** group that contain the old hostname by replacing the hostname with the wildcard character (*), so you will still be able to administer Infoprint Manager Security after the hostname is changed.
 - Make note of the members that you change so you can replace the wildcard with the new hostname once it is set.
3. Change the hostname/IP address using Windows interfaces. When the system asks, choose **not** to restart the system.

Note: If you have already changed the hostname/IP address of your system, you do not have to change them again. Continue with step 4.

4. In the Management Console, select **Edit->Changed Hostname**.
5. In the **Changed Hostname** dialog, enter the new hostname. Names are case sensitive. Make sure the new hostname that you type exactly matches the one on your Windows system and click **OK**.
6. When the system asks, restart the system so the changes take effect.
7. Update all of the client systems that communicated with the server using the old hostname/IP address to use the new one, including:
 - Infoprint Select clients
 - Infoprint Manager Administration GUI
 - Infoprint Manager Operations GUI
 - Windows gateway printers
 - IPP Gateway clients
 - Infoprint Manager Notifications clients
8. Update any Infoprint Manager Security ACLs or groups that reference IDs using the old hostname. In addition, if you changed any members of the **acl_admin** group in step 2, change the wildcard character to the new hostname.

Changing the hostname of a non-namespace server

Follow this procedure only if you are running both Infoprint Manager for AIX and Infoprint Manager for Windows as interoperating print servers and need to change the hostname of the system that Infoprint Manager for Windows is running on.

1. Complete “Changing the server hostname and IP address” on page 51 on the non-namespace server. At the end of the procedure, the Infoprint Manager for Windows NT/2000 server should be stopped and can no longer access the namespace on the AIX machine.
2. On the system that the Infoprint Manager for Windows NT/2000 runs on, disconnect the network drive that was mapped to the namespace on the AIX machine.
3. Log on to AIX server machine as **root**.
4. Follow this path: **Smitty-> Infoprint Printing System-> Infoprint Utilities-> Configure Secondary Servers (NFS)-> Link from Primary Server to Secondary Server**
5. In the dialog that appears:
 - a. Type the new hostname in the **Secondary Server Host Name** field.
 - b. Type the new server IP address in the **Secondary Server IP Address** field.
 - c. For **Is /ipdata used for job ticketing?** answer:
 - **Yes** if you use /ipdata with Infoprint Submit.
 - **No** if you do not.
 - d. For **Retain secondary servers previously defined?** answer **Yes**.
6. On the NT server machine, remap the network drive to the `\\aixserver\var\pddir\default_cell` filesystem using the same drive letter that you used for the network drive that you disconnected in step 2. Map the drive using the same user id and password that you used during the initial multiserver environment setup.
7. Log on to the AIX server machine as **root**.
8. Stop the Infoprint Manager for AIX server.

9. Clear the namespace by issuing the `clrfstns /var/pddir/default_cell` command.
10. Create the shared namespace by issuing the `crtfstns /var/pddir/default_cell` command.
11. Start the Infoprint Manager for AIX server.
12. Start the Infoprint Manager for Windows NT/2000 non-namespace server.

Changing only the server IP address

If you are only changing the IP address and you have updated your system name resolution so that the old name resolves to the new IP address, Infoprint Manager will be reset to use the new IP address with the old hostname when you restart the system. If any of your Infoprint clients refer to Infoprint Manager using the IP address, you must update them so they use the new IP address.

Chapter 11. Setting up your Infoprint Manager server to use resources on a different Windows system

Sometimes, users choose to store printing resources (such as pagedefs, formdefs, and fonts) on systems other than the one that their Infoprint Manager server runs on. If you choose that setup, you must first make sure that Infoprint Manager has access to the resources, then tell Infoprint Manager where the resources are located. Use the following procedures to complete these tasks.

1. "Giving Infoprint Manager access to remote resources"
2. "Telling Infoprint Manager where your remote resources are located" on page 82

Giving Infoprint Manager access to remote resources

Follow one of the procedures below to set up access to remote resources.

- If Infoprint Manager and the other system are running on Windows NT, refer to "Using Windows NT".
- If Infoprint Manager and the other system are running on Windows 2000, refer to "Using Windows 2000" on page 66.

Note: Both of your systems must be running on the same operating system. You cannot mix Windows NT and Windows 2000 systems in this environment. The systems should also operate in the same domain.

Using Windows NT

1. *Set up a domain user account that your Infoprint Manager server will run under.*
 - a. Log on to the system that your Infoprint Manager server runs on as a user who is a member of the Domain Administrators group.
 - b. Click the Windows **Start** button and select **Programs -> Administrative Tools (Common) -> User Manager for Domains**.

Note: This user account must be a domain account, not a local user account. Be sure you create the account with **User Manager for Domains**, not the **User Manager** for the local system. If **User Manager for Domains** does not appear in your Start menu, you must create this user on a different system in the domain that your Infoprint Manager server operates in, such as the Primary Domain Controller (PDC).

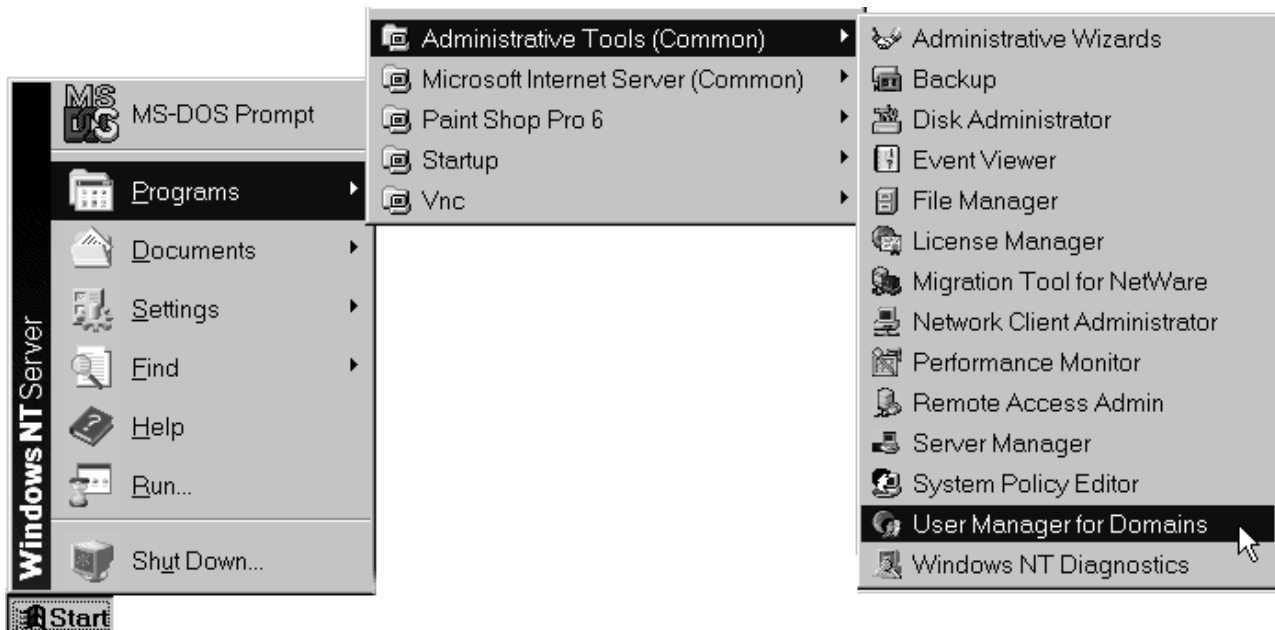


Figure 9. Windows NT — User Manager for Domains: Open

c. In the User Manager window, select **User** → **New User**.



Figure 10. Windows NT — User Manager for Domains: New User

- d. In the **New User** dialog, do the following:
- Type a name for this user account.
 - Type a password in both of the password fields.
 - Clear all of the check boxes, then select **Password never expires** (so you don't have to change the password and update all of your servers on a regular basis).
 - Click **Groups**.

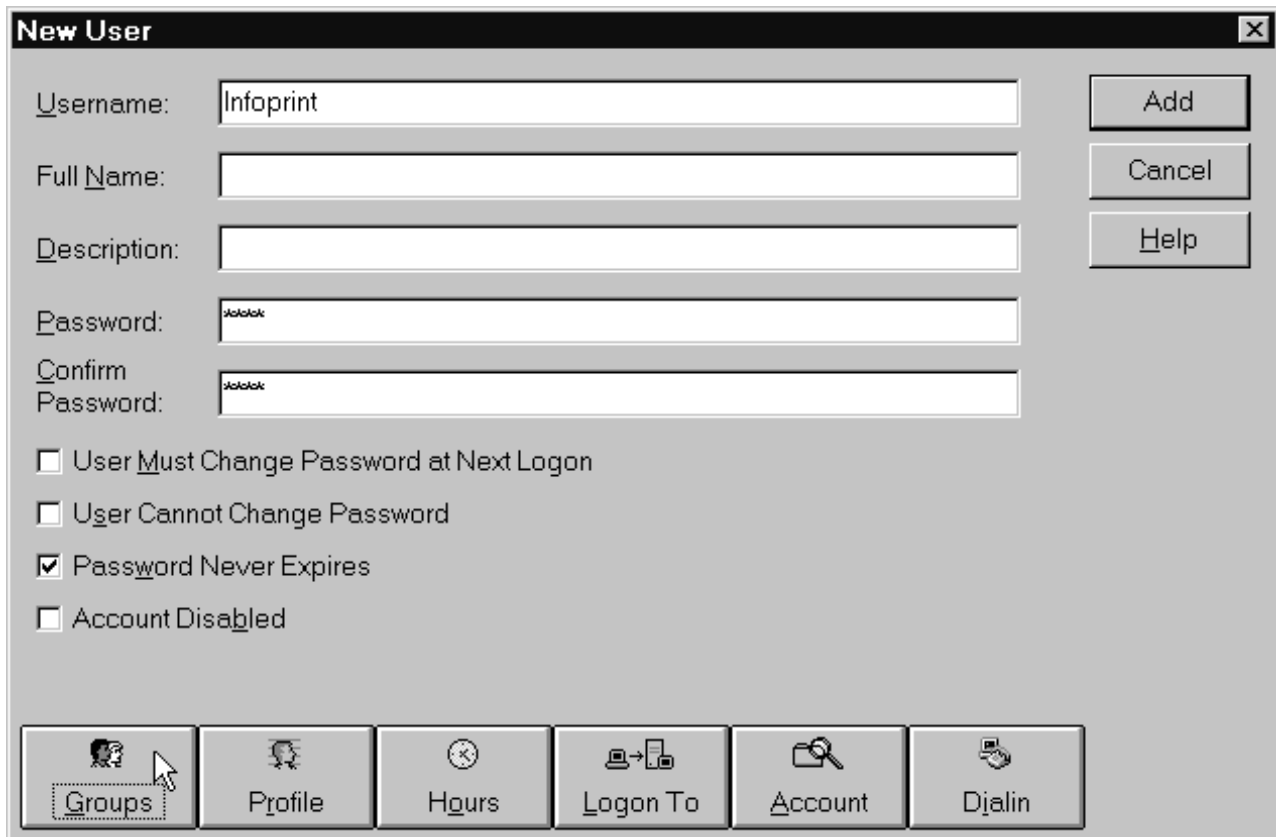


Figure 11. Windows NT — User Manager for Domains: Set User

- e. In the **Group Memberships** dialog:
 - 1) Find **Domain Admins** in the **Not member of** list and click to select it.

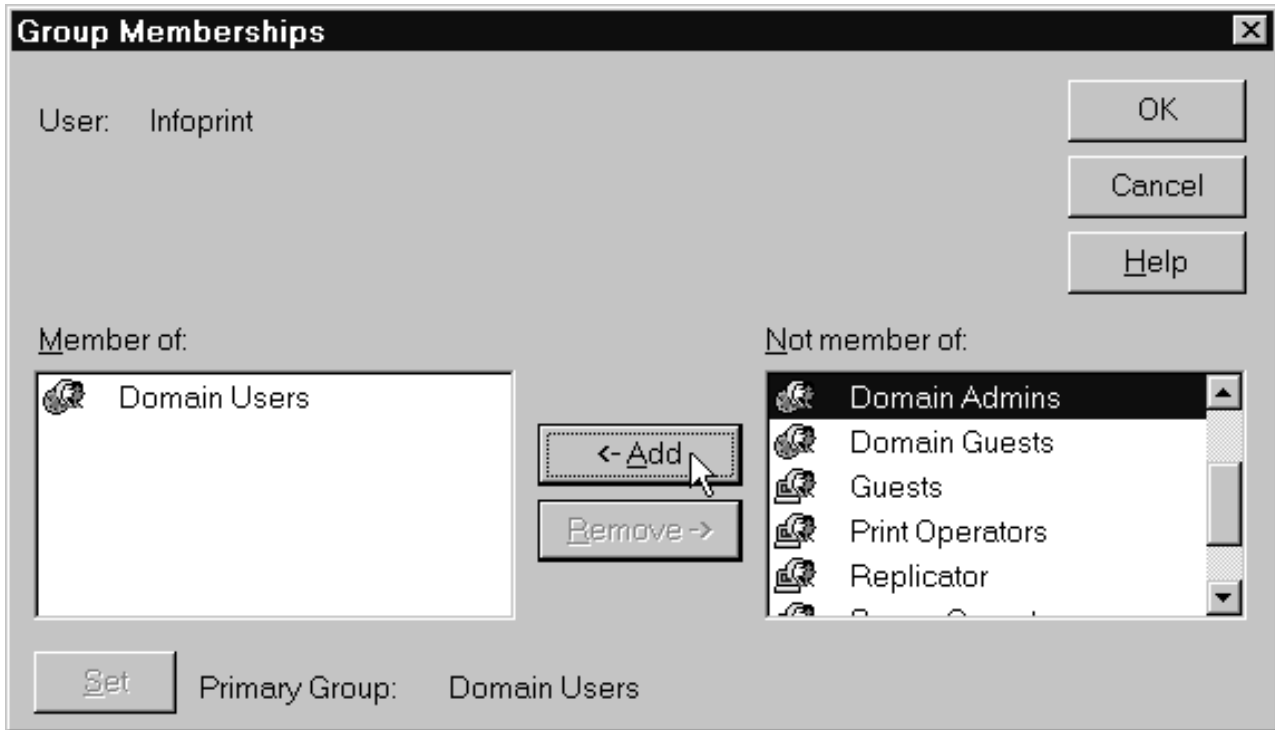


Figure 12. Windows NT — User Manager for Domains: Add to group

- 2) Click <- Add.
- 3) Select **Domain Admins** in the **Member of** list and click **Set** to make **Domain Admins** this user's primary group.

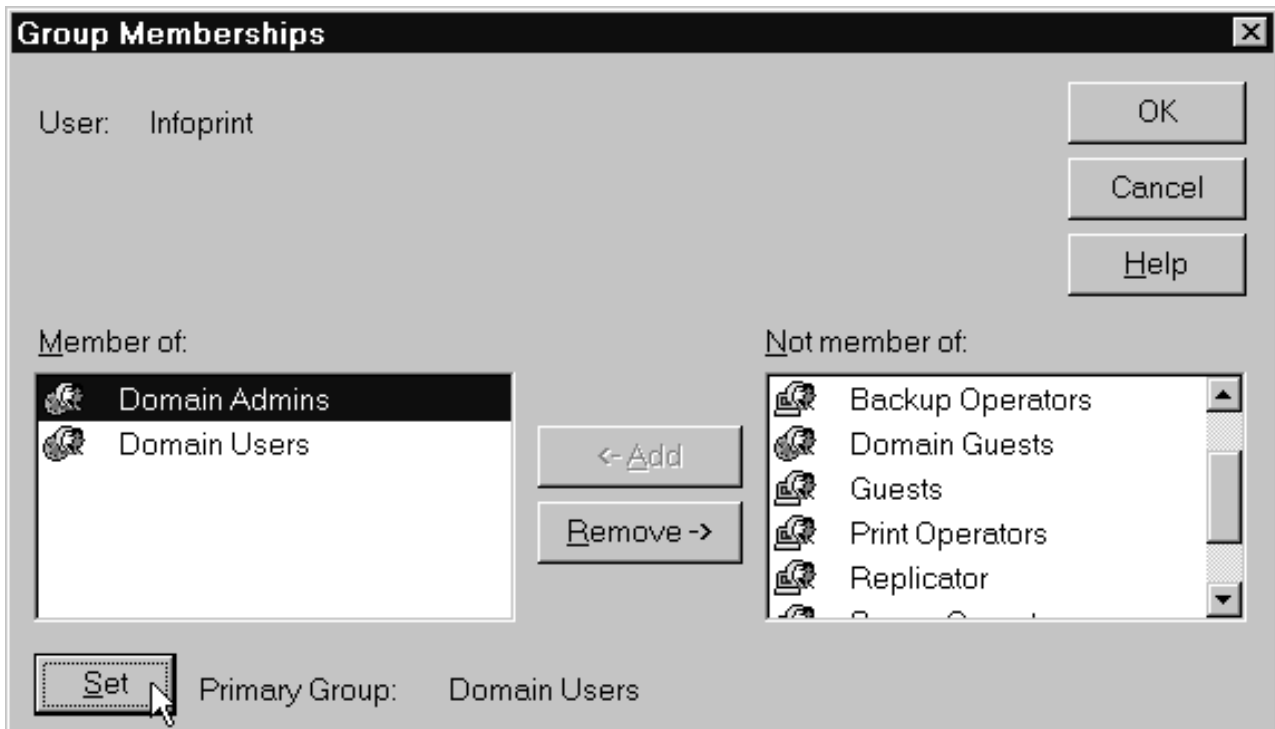


Figure 13. Windows NT — User Manager for Domains: Set User

- 4) Click **OK**.
 - f. In the **New User** dialog, click **Add** to create this user.
 - g. Click **Cancel** to dismiss the dialog.
 - h. Verify that the user you just created appears in the list of users in the User Manager window, then close it.
2. *Grant the appropriate user rights to the new domain user on the system that your Infoprint Manager server runs on.*
 - a. If you did not perform step one on the system that your Infoprint Manager server runs on, log on to it as a user who is a member of the Administrators group.
 - b. Click the Windows **Start** button and select **Programs -> Administrative Tools -> User Manager** or **Programs -> Administrative Tools -> User Manager for Domains** (either one will work this time).
 - c. In the User Manager window, select **Policies -> User Rights**.



Figure 14. Windows NT — User Manager: User Rights Policy 1

- d. In the **User Rights Policy** dialog:
 - Select the **Show Advanced User Rights** check box.
 - Select **Act as part of the operating system** in the **Right** drop-down list.
 - Click **Add**.

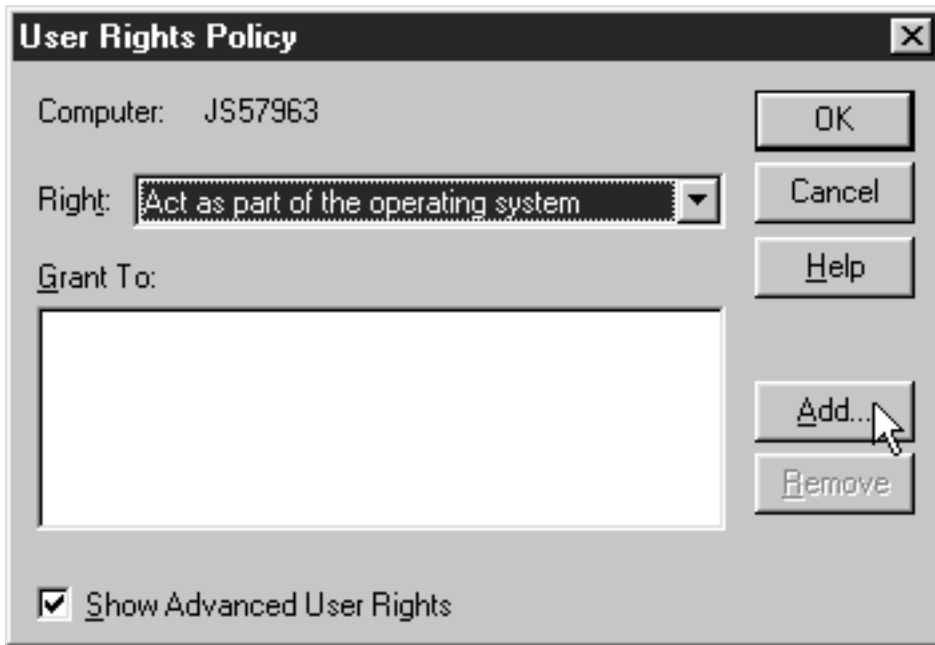


Figure 15. Windows NT — User Manager: User Rights Policy 2

- e. In the **Add Users and Groups** dialog, select the domain that your Infoprint Manager system operates in from the **List names from** drop-down list and click **Show Users**.

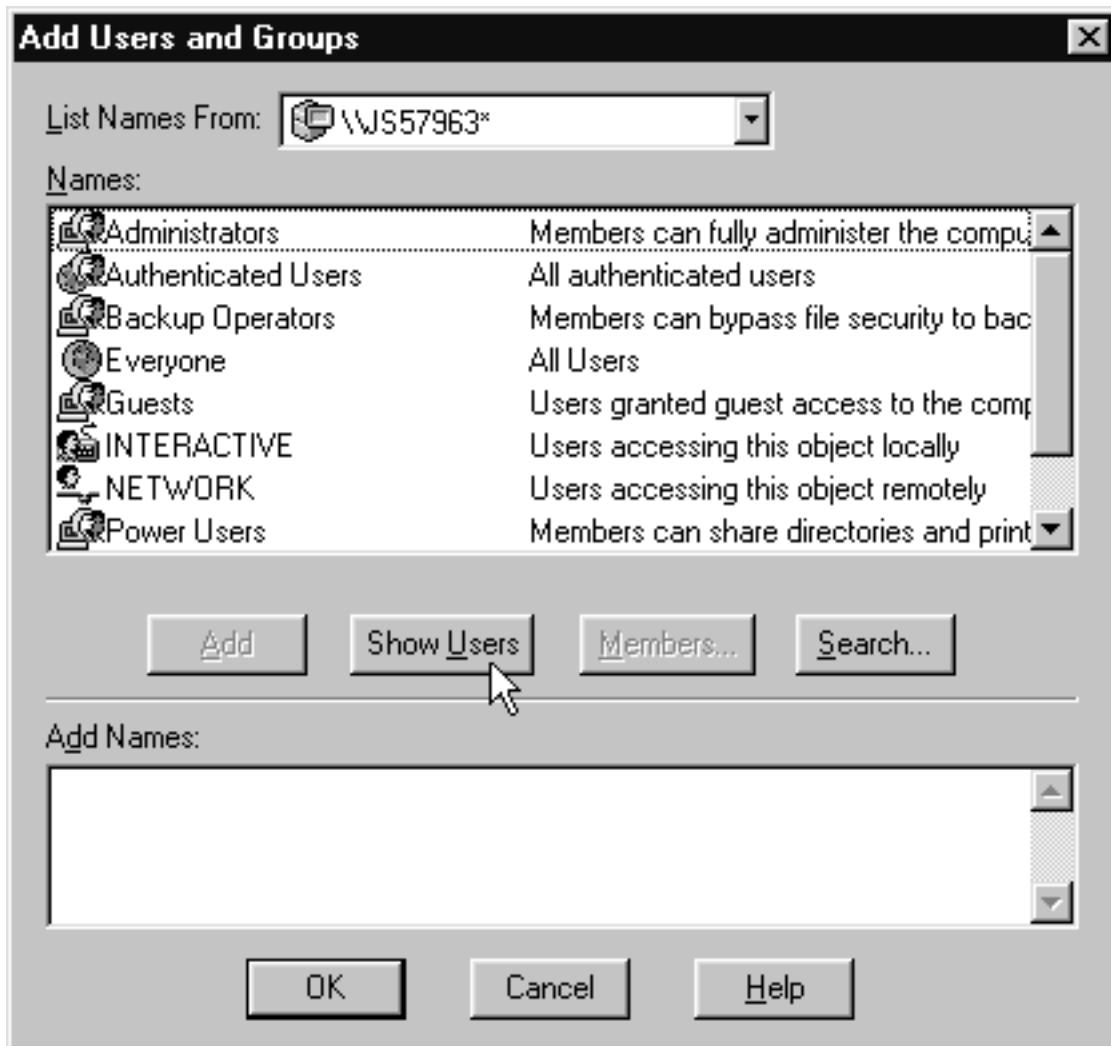


Figure 16. Windows NT — User Manager: Add user

- f. Find the domain user that you created in step 1 on page 55 in the **Names** box, and select it, then click **Add**. The domain user name will appear in the **Add Names** box.
 - g. Click **OK**.
 - h. Repeat from step 2d on page 59 for two other rights: **Create permanent shared objects** and **Log on as a service**.
 - i. Click **OK** to close the **User Rights Policy** dialog.
 - j. Close the User Manager window.
3. *Set up the Infoprint Manager server to use the new user account.*
 - a. Click the Windows **Start** button and select **Programs → IBM Infoprint Manager → Management Console**.
 - b. Stop the Infoprint Manager server by selecting **File → Stop Server**.

Note: If you are not able to stop the server because you are not authorized for that action, check the Infoprint Manager Access Control List (ACL) for the **Shutdown** operation. Log off and log back on as a user who is on the ACL.

- c. Select Edit --> Service Account/Domain.

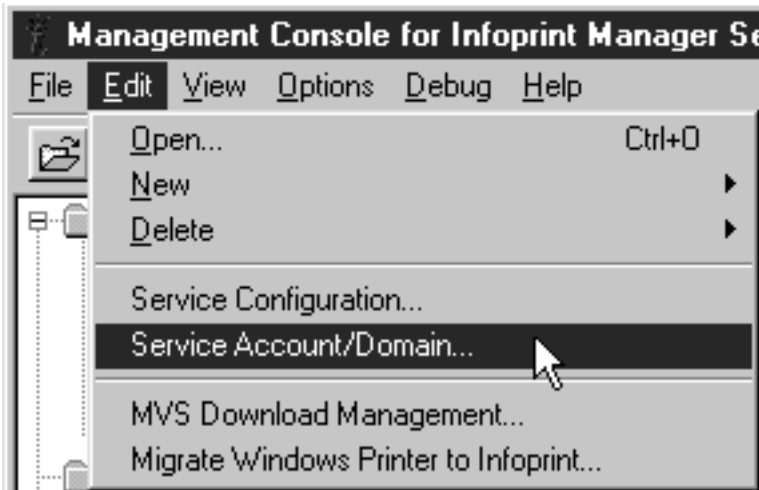


Figure 17. Windows NT — Management Console: Edit-> Service Account/Domain

- d. Enter the name of the domain that the Infoprint Manager server is installed in and the user name and password from step 1 on page 55.

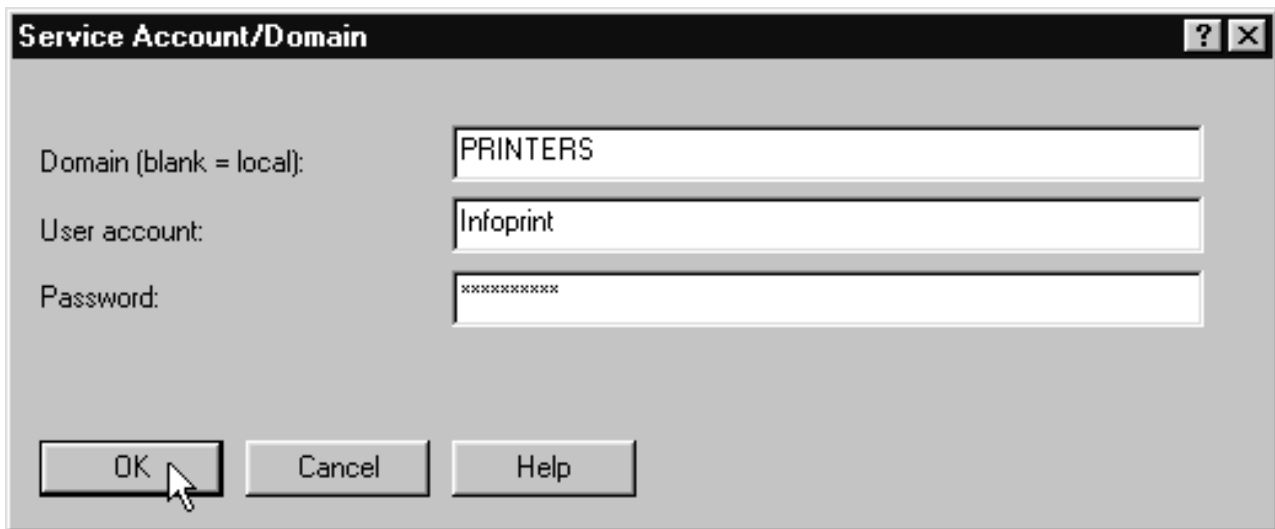


Figure 18. Windows NT — Management Console: Service Account/Domain dialog

- e. Click **OK**.
4. *On the machine that the AFP resources reside on, share the directory that the resources reside in so the domain user has access to it.*
 - a. Log on to the system that your resources reside on as a user who is a member of the Administrators group.
 - b. Click the Windows **Start** button, and select **Programs** -> **Windows NT Explorer**.
 - c. Navigate to the directory that your resources are stored in.
 - d. Right-click that directory and select **Sharing** from the pop-up menu.

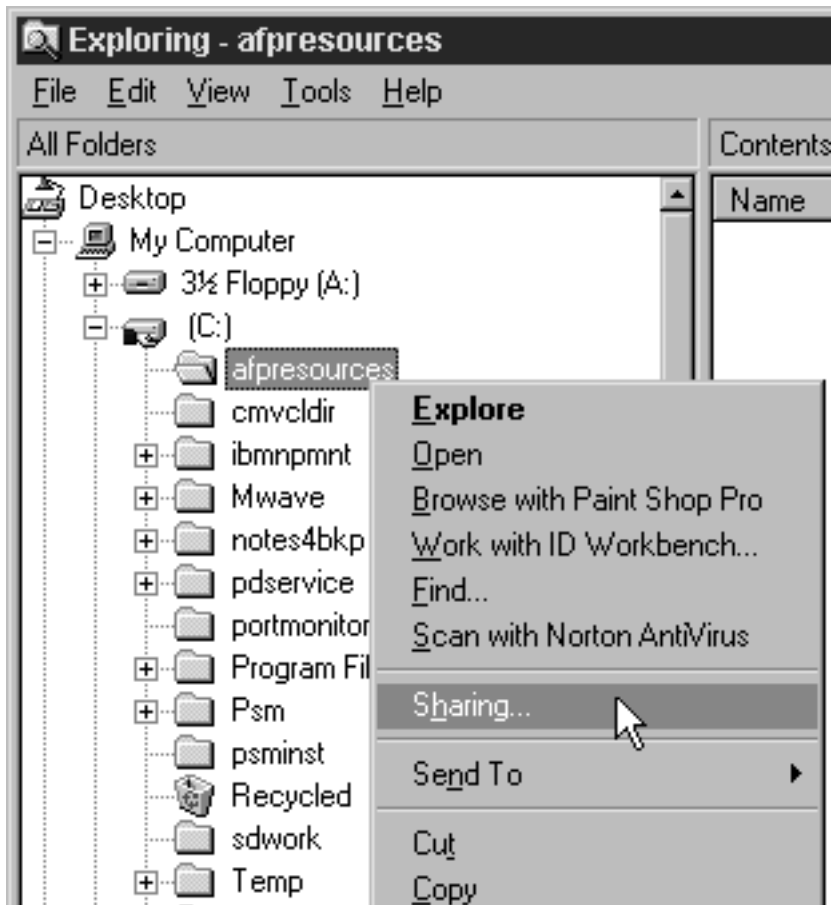


Figure 19. Windows NT — Explorer: Sharing

- e. In the **Properties** dialog, select **Shared as** and type **Shared to Infoprint Manager** in the **Comment** text box.

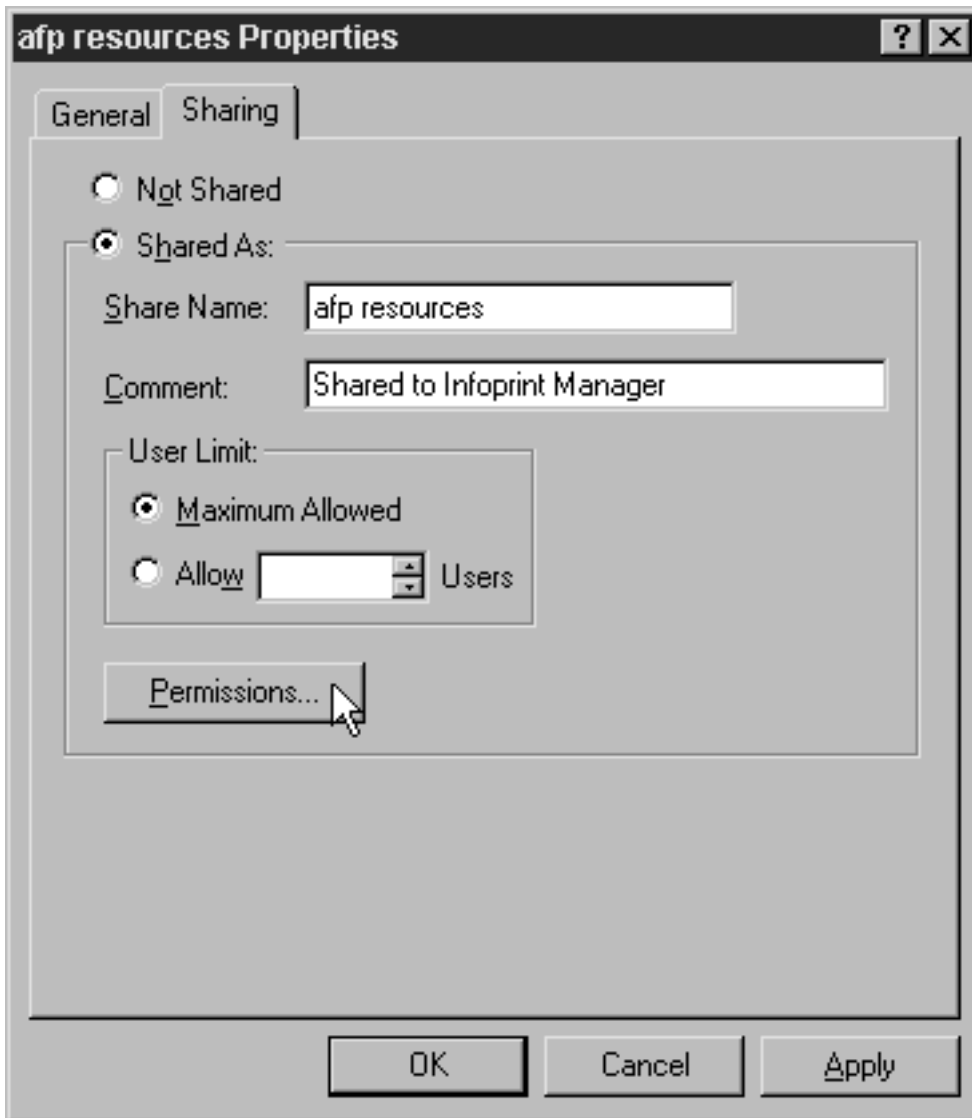


Figure 20. Windows NT — Explorer: Directory Properties dialog

- f. Click **Permissions**.
- g. In the **Access through Shared Permissions** dialog, click **Add**.

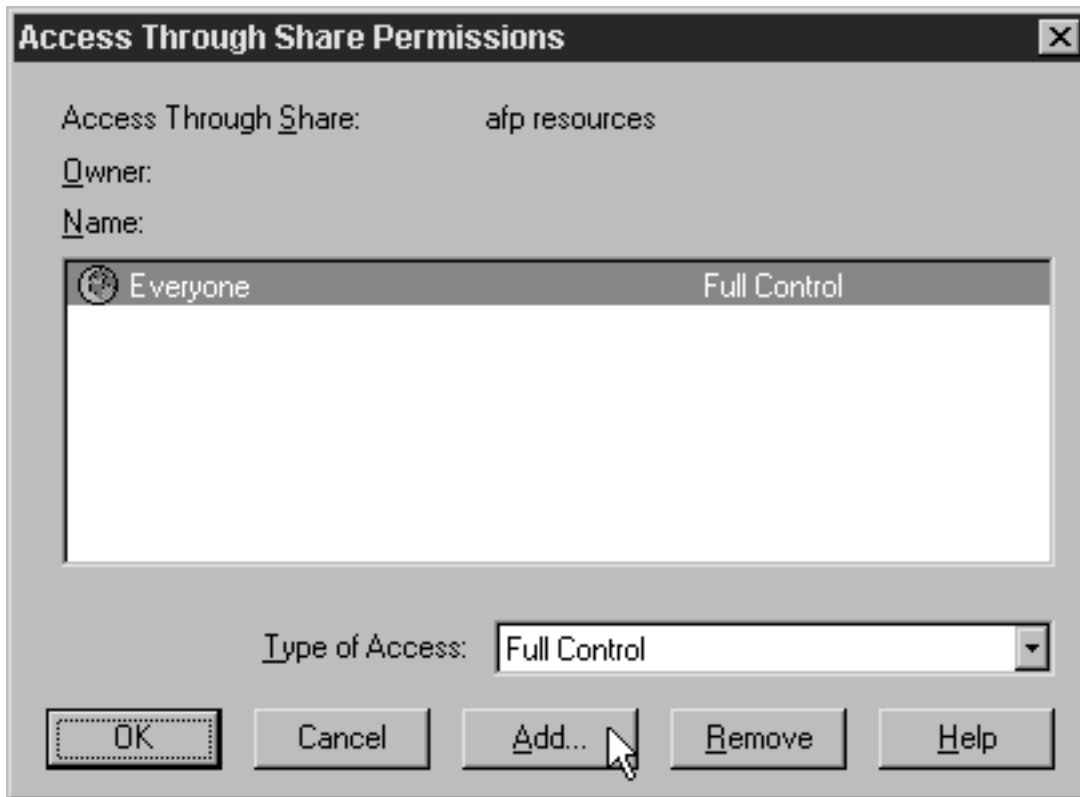


Figure 21. Windows NT — Explorer: Access through Shared Permissions

- h. In the **Add Users and Groups** dialog:
 - 1) Select the appropriate domain in the **List Names From** drop-down box.
 - 2) Find **Domain Admins** in the **Names** box.
 - 3) Click **Add**.
 - 4) Select **Full Control** in the **Type of Access** drop-down list.

Note: You can add any other users or groups that you want at this point.
 - 5) Click **OK**.

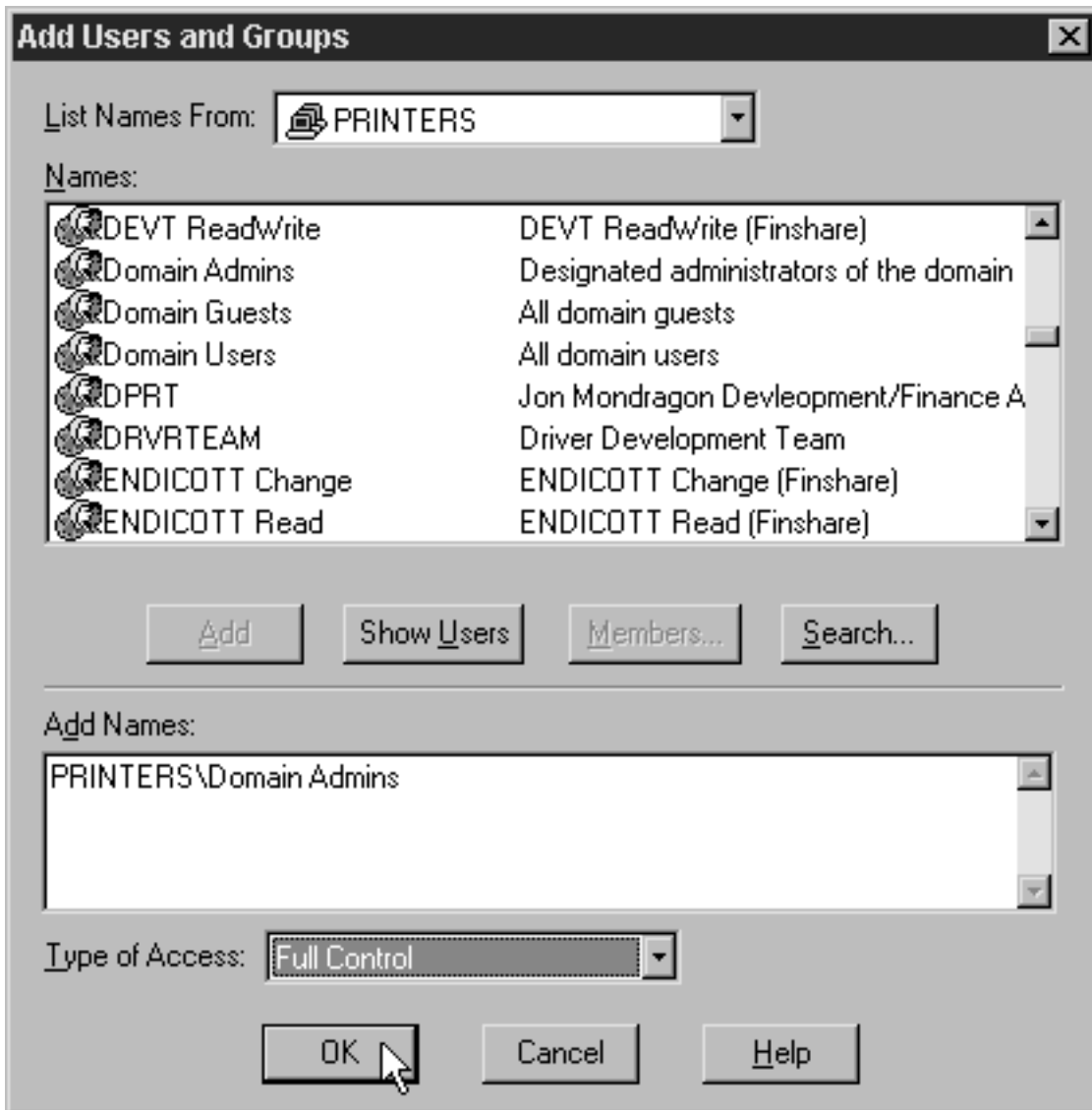


Figure 22. Windows NT — Add Users and Groups dialog

- i. Click **OK** in the **Access through Shared Permissions** dialog.
- j. Click **OK** in the **Properties** dialog.
- k. Return to the Management Console on the system that Infoprint Manager is installed on and select **File** → **Start Server** to restart the Infoprint Manager server.

Using Windows 2000

1. *Set up a domain user account that your Infoprint Manager server will run under.*
 - a. Log on to the domain that your Infoprint Manager servers operate in as a domain administrator.
 - b. Click the Windows **Start** button and select **Programs** → **Administrative Tools** → **Active Directory Users and Computers**.
 - c. In the left pane of the Active Directory Users and Computers window, find the Users directory.

- d. Right-click the Users directory and select **New** -> **User** from the pop-up menu.



Figure 23. Windows 2000 — Active Directory Users and Computers: New User

- e. In the first dialog of the New Object – User wizard, type a name for this user in the **Full name** and the **User logon name** fields.

Note: You do not have to fill in the other name fields.

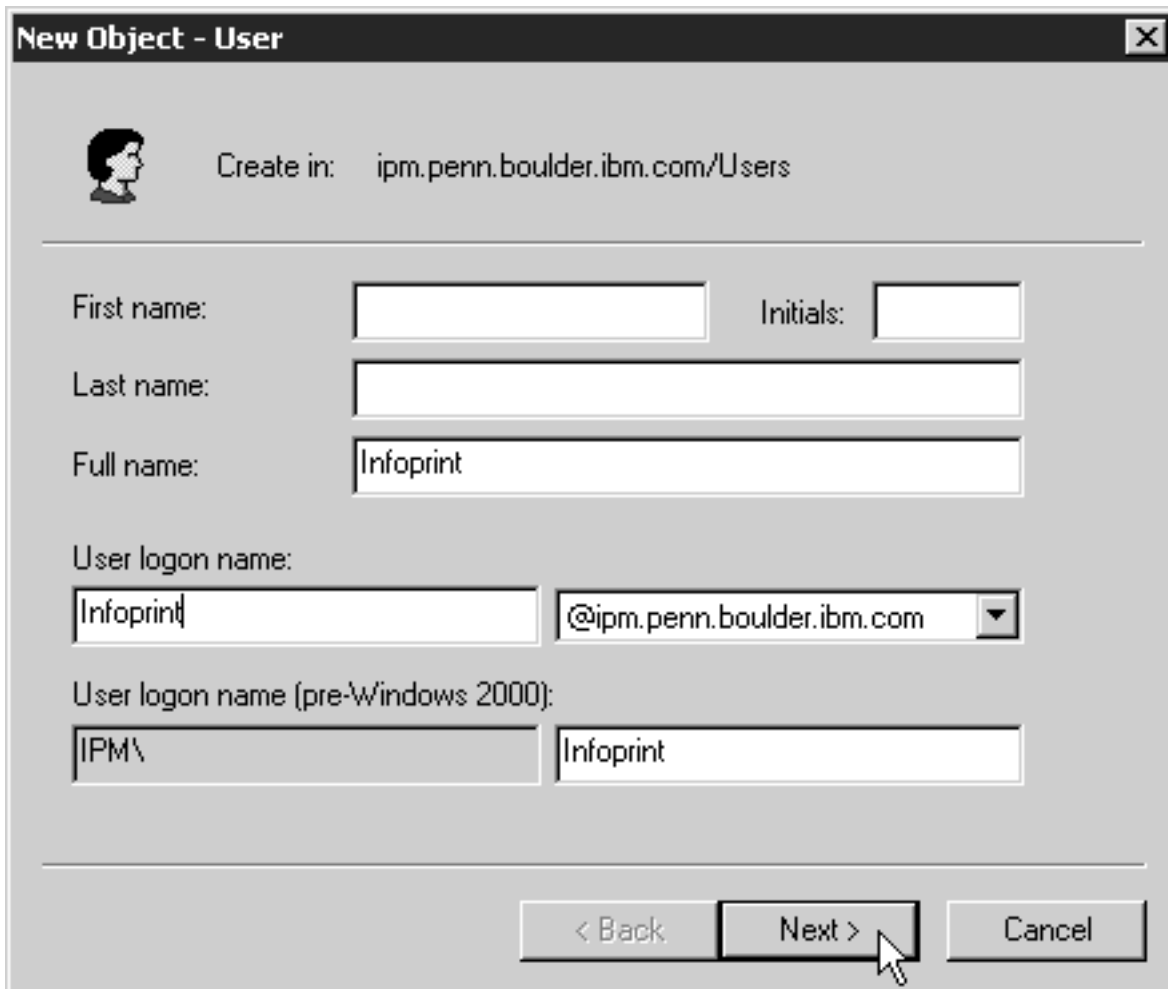


Figure 24. Windows 2000 — Active Directory Users and Computers: New Object- User

- f. Click **Next**.
- g. In the second dialog of the New Object – User wizard, enter a password for this user. Clear all of the check boxes, then select **Password never expires** (so you don't have to change the password and update all of your servers on a regular basis).



Figure 25. Windows 2000 — Active Directory Users and Computers: New Object— User 2

- h. Click **Next** to verify the settings for the user and complete the wizard.
- i. Once you have created the user, right-click the user in the right pane of the **Active Directory Users and Computers** window and select **Add members to a group** from the pop-up menu.

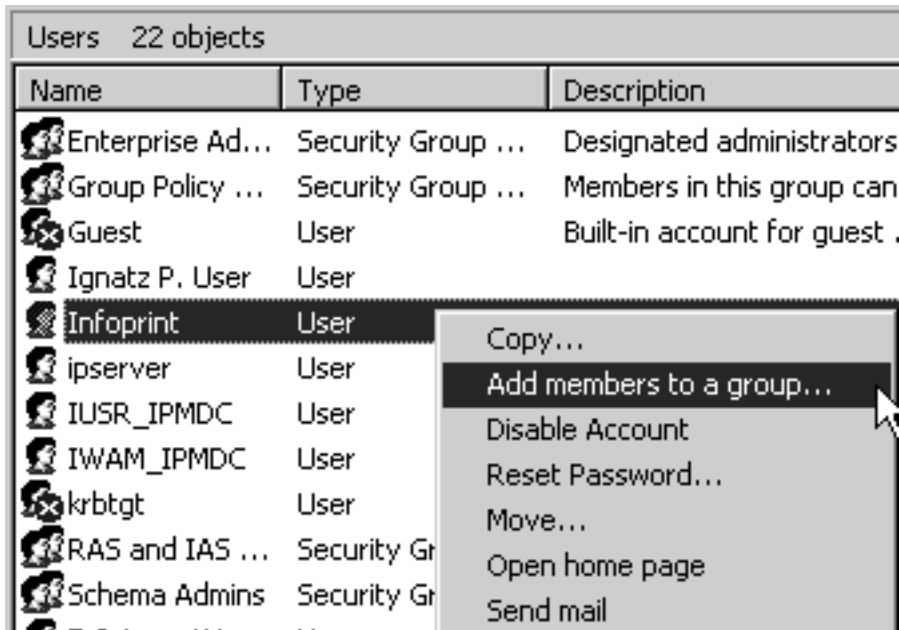


Figure 26. Windows 2000 — Active Directory Users and Computers: Add Members to Group

- j. In the **Select Group** dialog, find and select the **Domain Administrators** (or **Domain Admins**) group. Click **OK** to add your user to the group.

Note: You can leave the **Name** field blank.

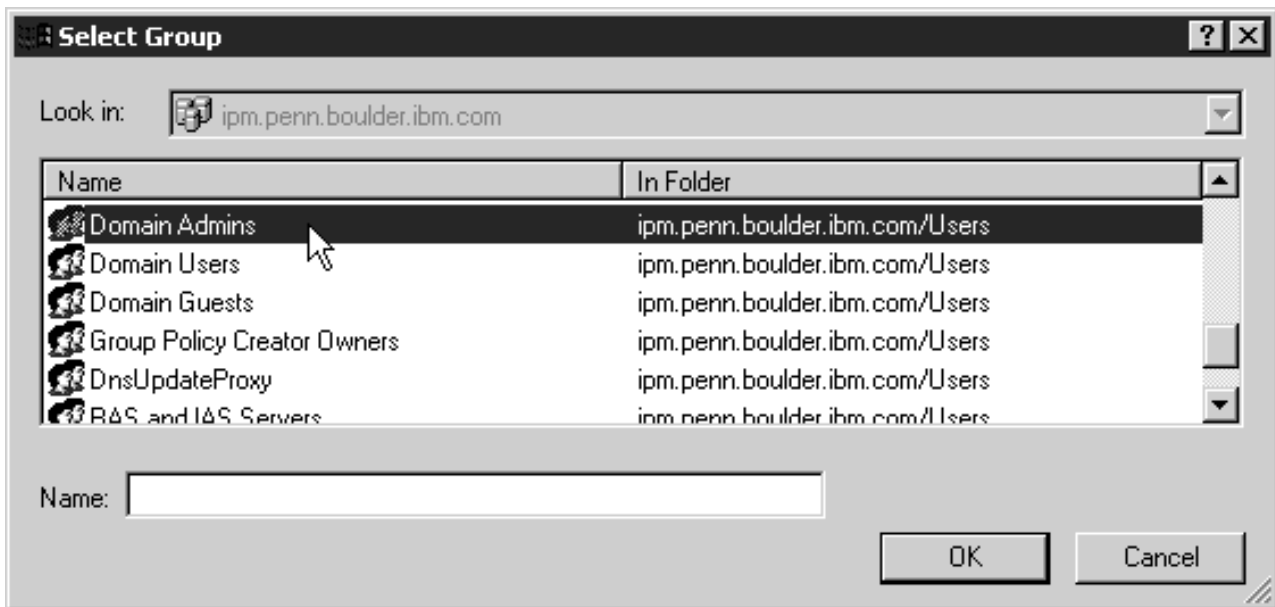


Figure 27. Windows 2000 — Active Directory Users and Computers: Select Group

2. *Grant the appropriate user rights to the new domain user on the system that your Infoprint Manager server runs on.*
 - a. Log on to the system as a user who is a member of the Domain Administrators group.
 - b. Click the Windows **Start** button and select **Settings --> Control Panel**.

- c. In the Control Panel window, double-click the **Administrative Tools** icon.
- d. In the Administrative Tools window, double-click the **Local Security Policy** icon.
- e. In the right pane of the Local Security Settings window, double-click **Local policies**.

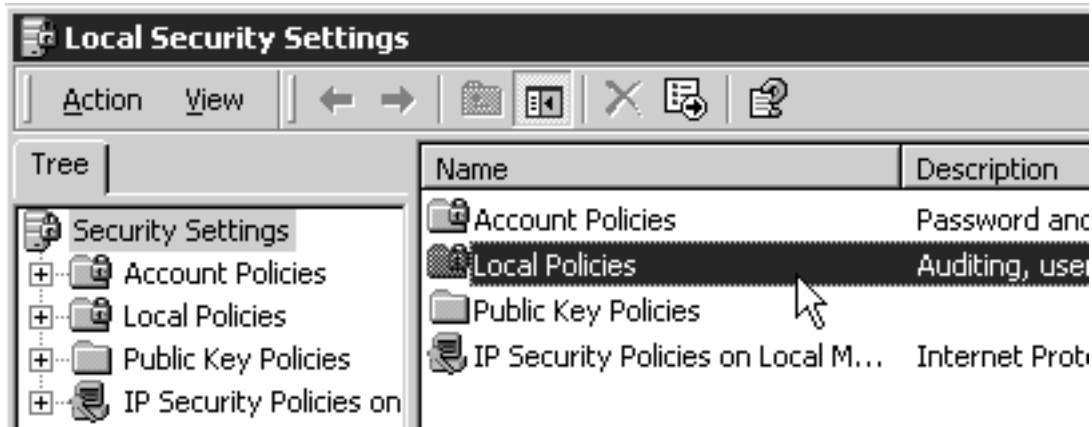


Figure 28. Windows 2000 — Local Security Policy: Local Security Settings

- f. Double-click **User Rights Assignment** in the right pane.



Figure 29. Windows 2000 — Local Security Policy: User Rights Assignment

- g. In the list of user rights, select **Act as part of the operating system** and double click it.

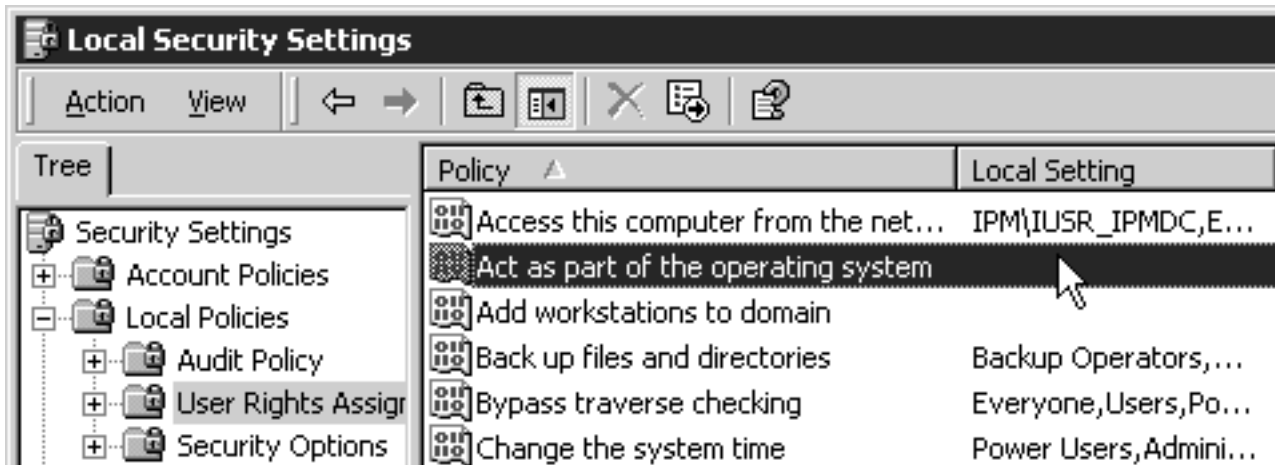


Figure 30. Windows 2000 — User Rights Assignment: Act as part of the operating system

- h. In the **Local Security Policy Setting** dialog, click **Add**.

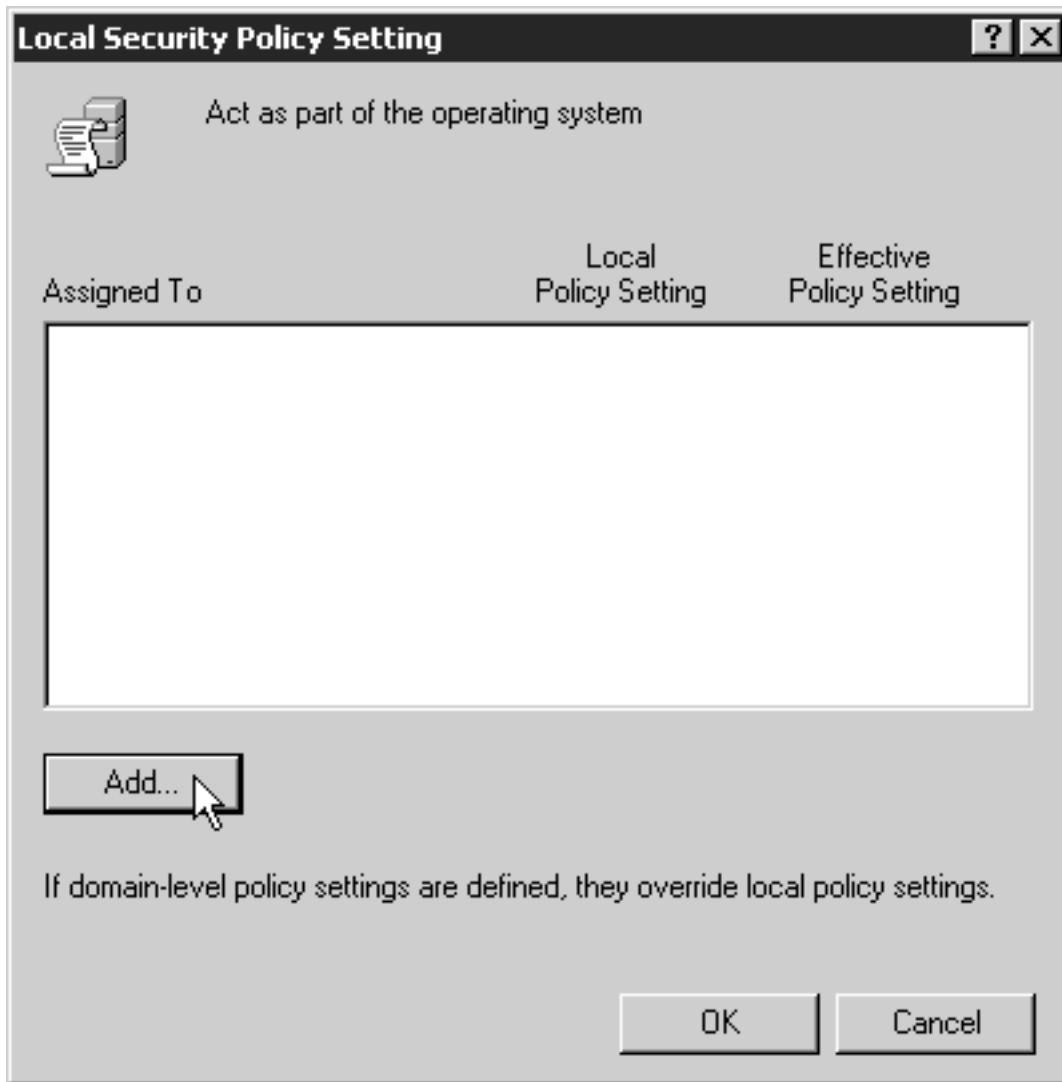


Figure 31. Windows 2000 — Local Security Policy Setting

- i. In the **Select Users or Groups** dialog, click the arrow on the drop-down list box and select the domain that your Infoprint Manager servers exist in.

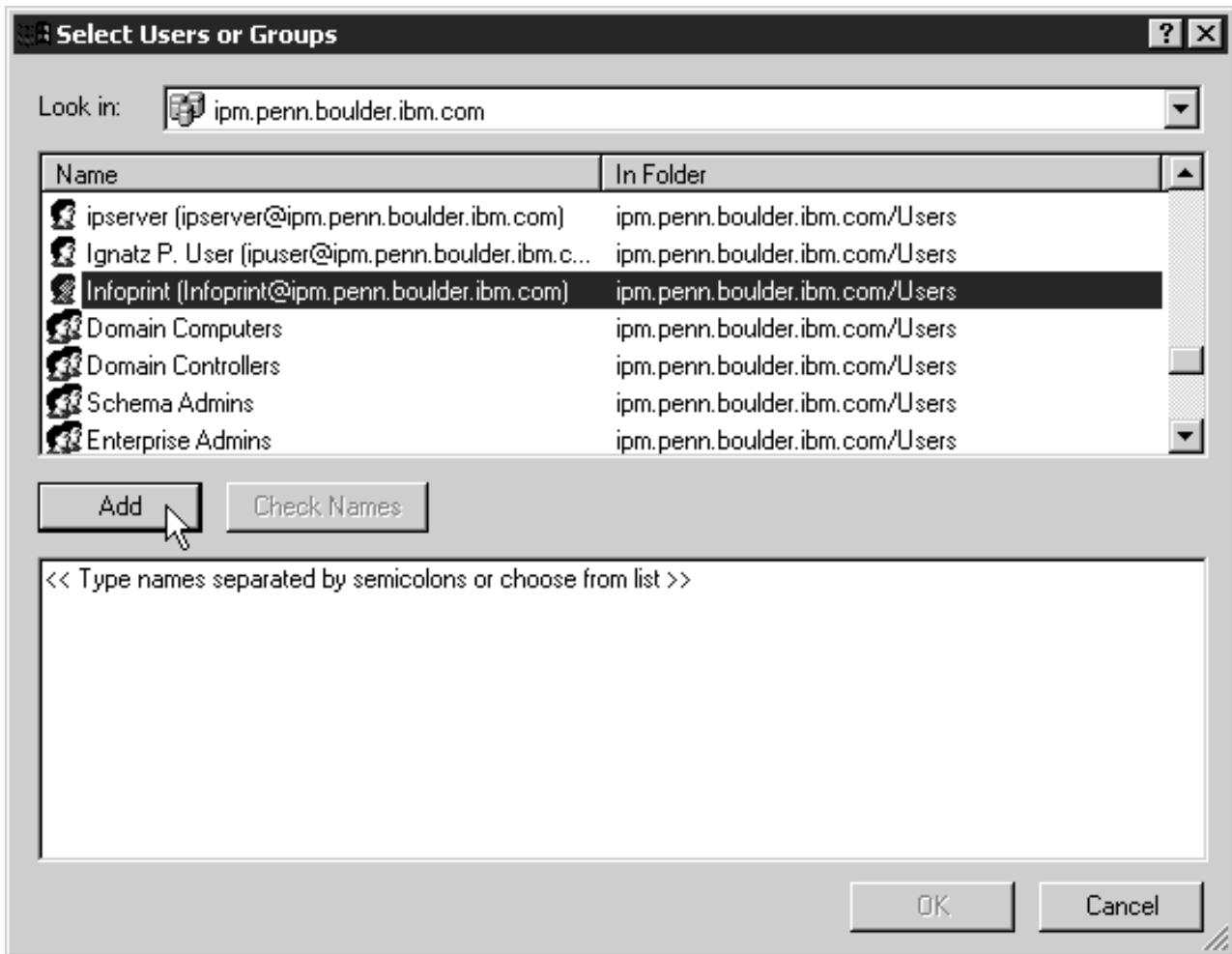


Figure 32. Windows 2000 — Local Security Policy Setting: Select Users or Groups

- j. Find and select the domain user you created and click **Add**.
- k. Click **OK** in the **Select Users or Groups** dialog.
- l. Click **OK** in the **Local Security Policy Setting** dialog.
- m. Repeat from step 2g on page 71 for two other user rights: **Create permanent shared objects** and **Log on as a service**.
3. *Set up the Infoprint Manager server to use the new user account.*
 - a. Click the Windows **Start** button and select **Programs → IBM Infoprint Manager → Management Console**.
 - b. Stop the Infoprint Manager server by selecting **File → Stop Server**.

Note: If you are not able to stop the server because you are not authorized for that action, check the Infoprint Manager Access Control List (ACL) for the **Shutdown** operation. Log off and log back on as a user who is on the access control list.

- c. Select **Edit → Service Account/Domain**.

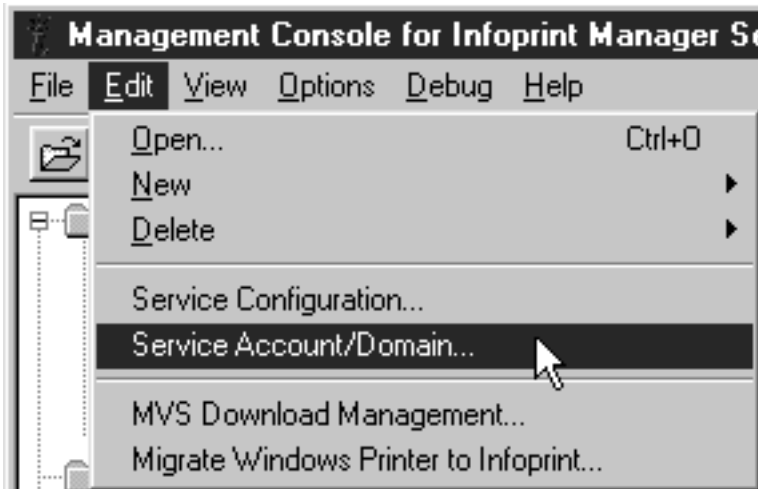


Figure 33. Windows 2000 — Management Console: Edit-> Service Account/Domain

- d. Enter the name of the domain that the Infoprint Manager server is installed in and the user name and password from step 1 on page 66.

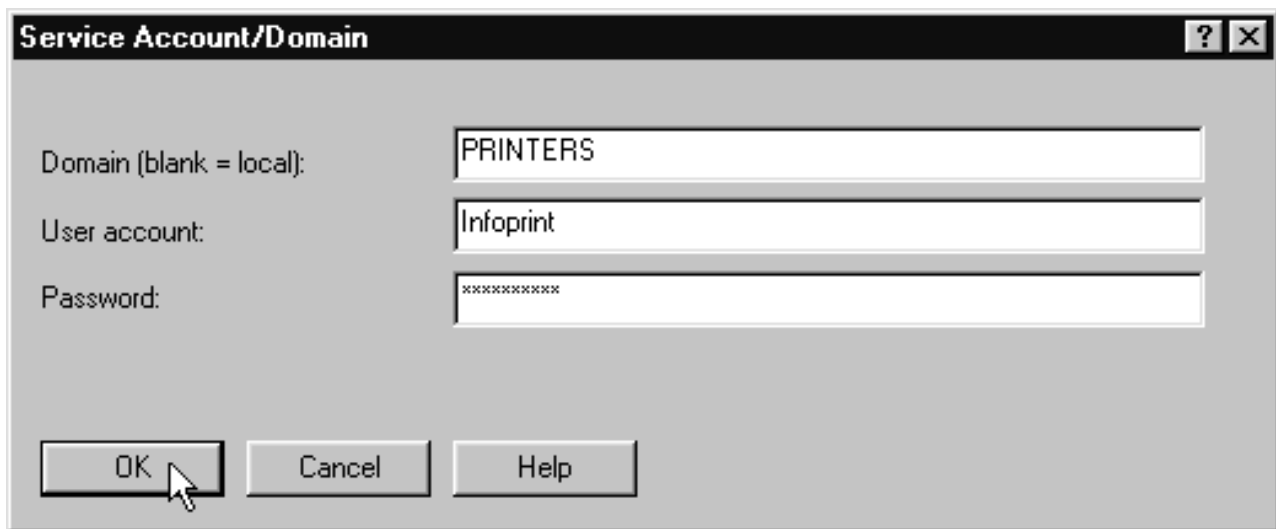


Figure 34. Windows 2000 — Management Console: Service Account/Domain dialog

- e. Click OK.
4. *On the machine where the AFP resources reside, share the directory that the resources reside in so the domain user has access to it.*
 - a. Log on to the system that your resources reside on as a user who is a member of the Administrators group.
 - b. Open **Windows 2000 Explorer** and navigate to the directory that your resources are stored in.
 - c. Right-click the folder and select **Properties** from the pop-up menu.

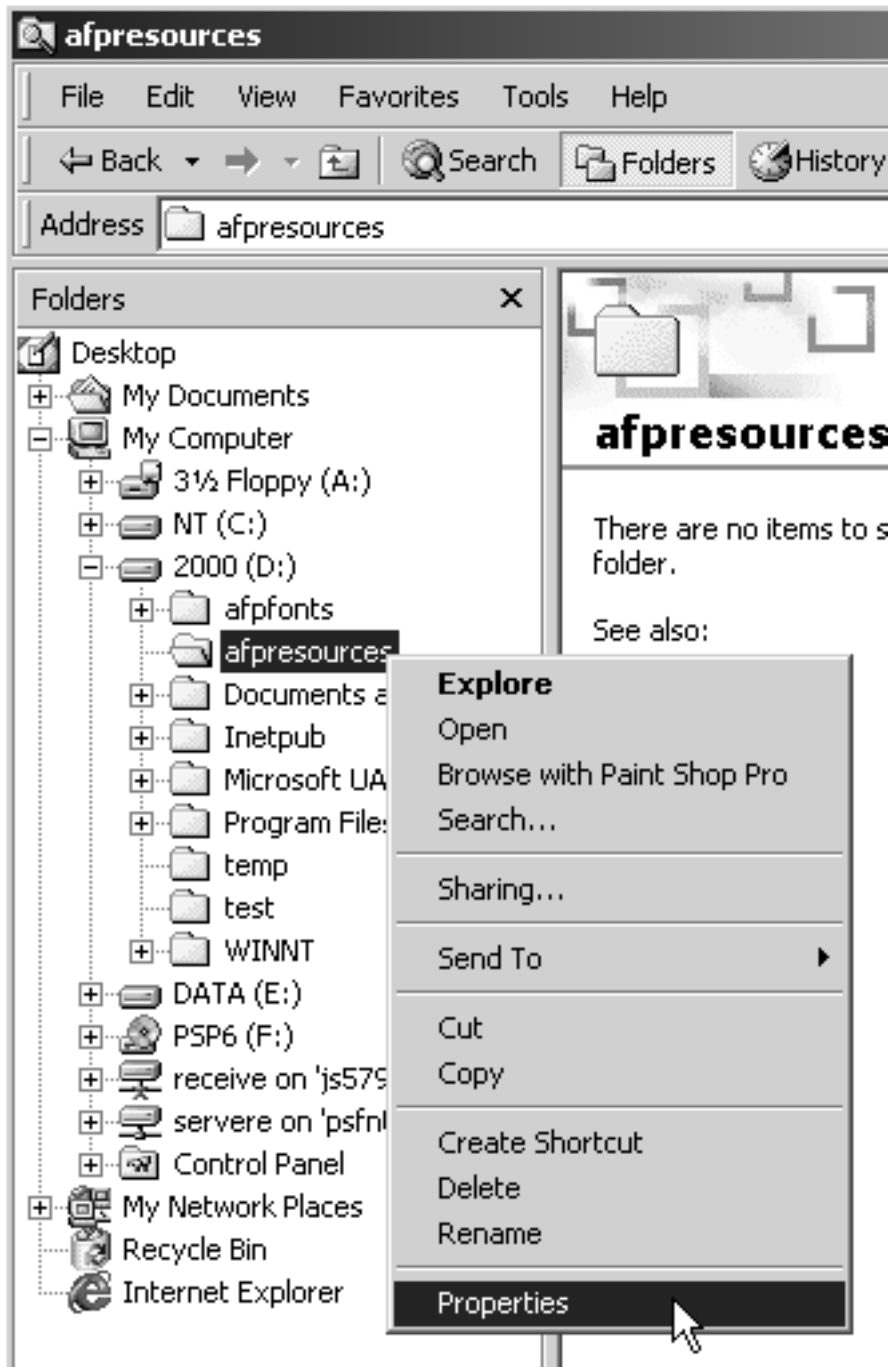


Figure 35. Windows 2000 — Explorer folder right-click menu

- d. In the **Properties** dialog, click the **Security** tab.

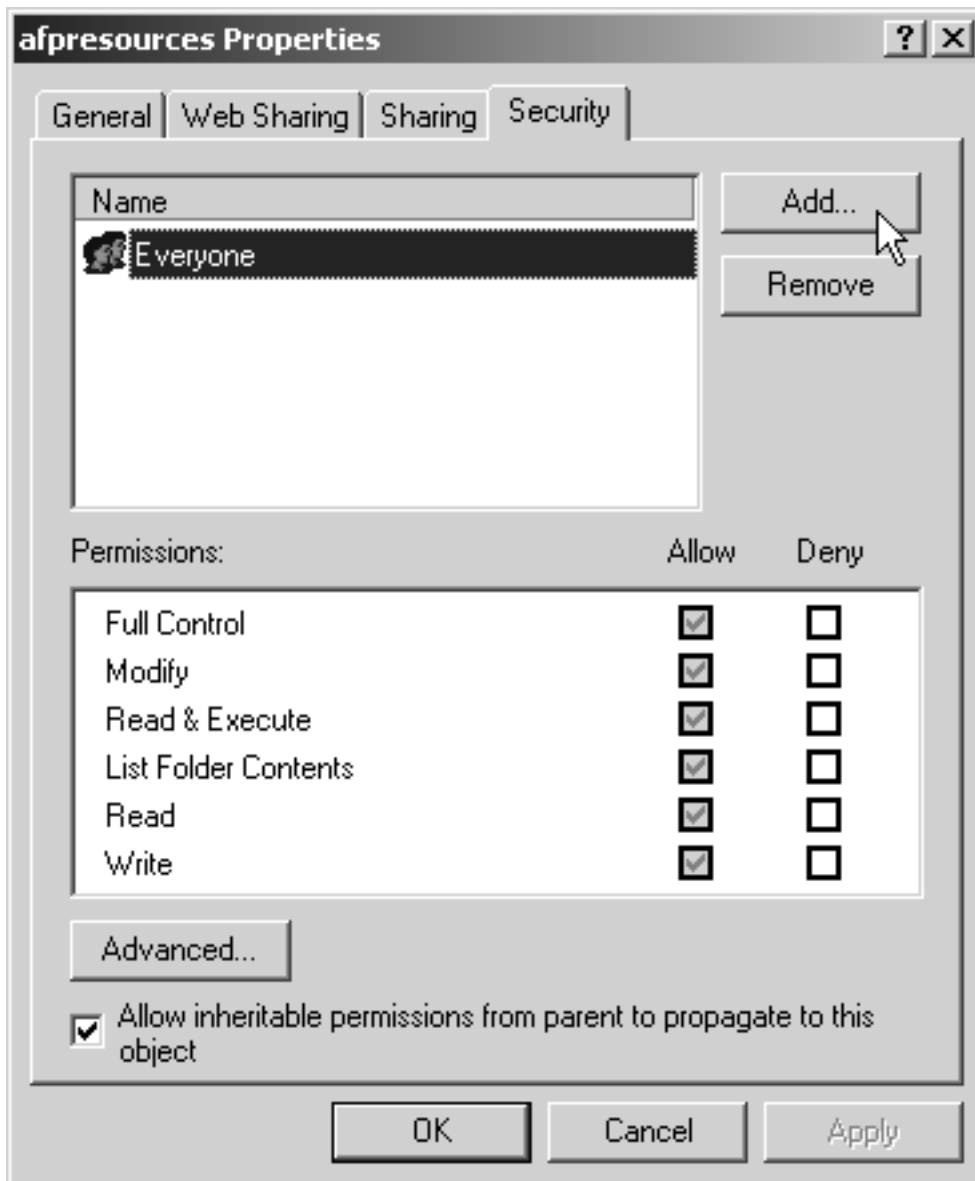


Figure 36. Windows 2000 — Properties dialog: Security tab

- e. If **Domain Admins** is not in the **Name** list, click **Add**.
- f. In the **Select Users, Computers, or Groups** dialog, choose the correct domain from the **Look in** drop-down list.
- g. Select **Domain Admins** in the **Name** list and click **Add**. You might also want to add the local administrators group as well.

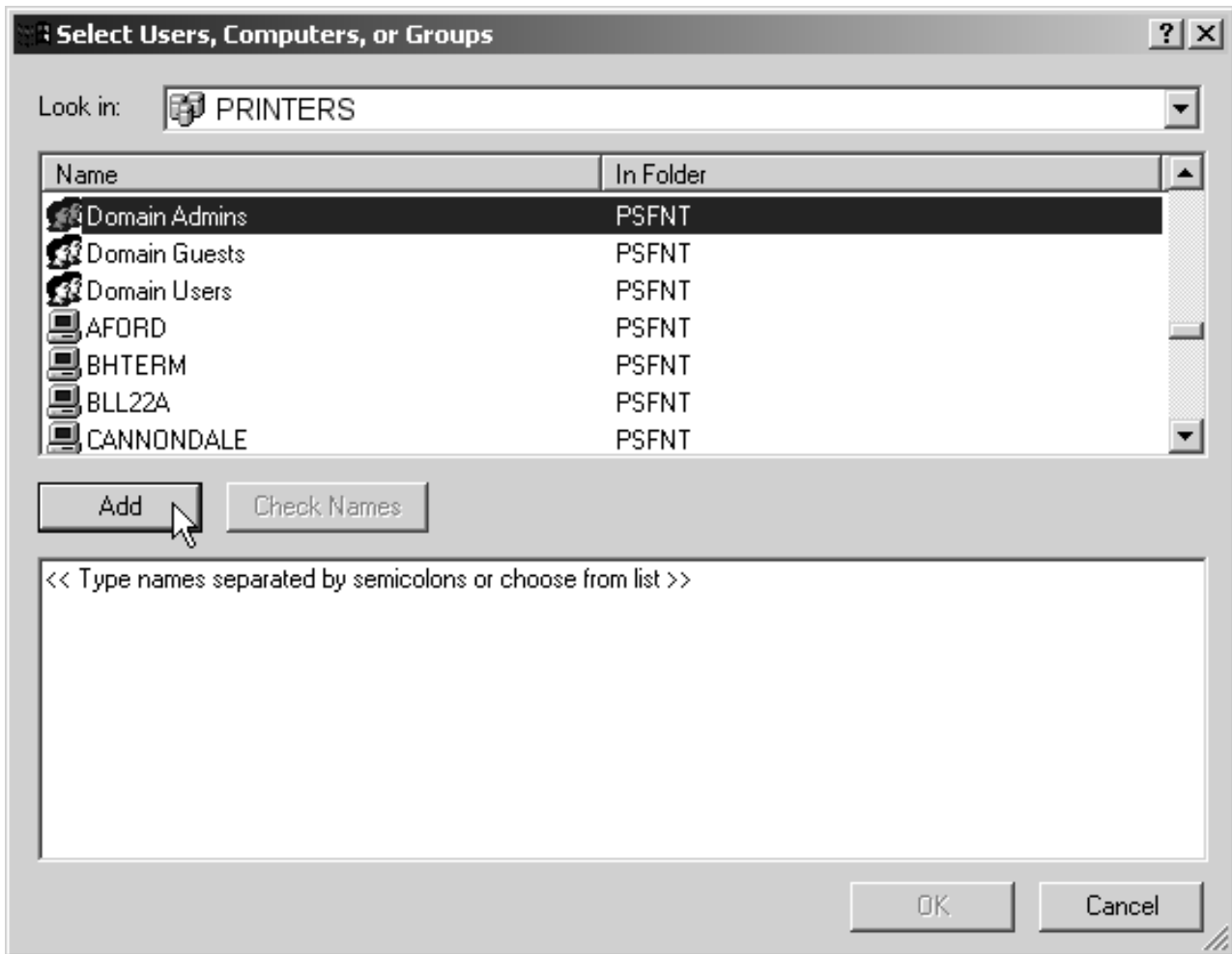


Figure 37. Windows 2000 — Add Domain Admins group to list

- h. Click **OK**.
- i. In the **Properties** dialog, make sure that **Full Control** is allowed for **Domain Admins** in the **Permissions** box.

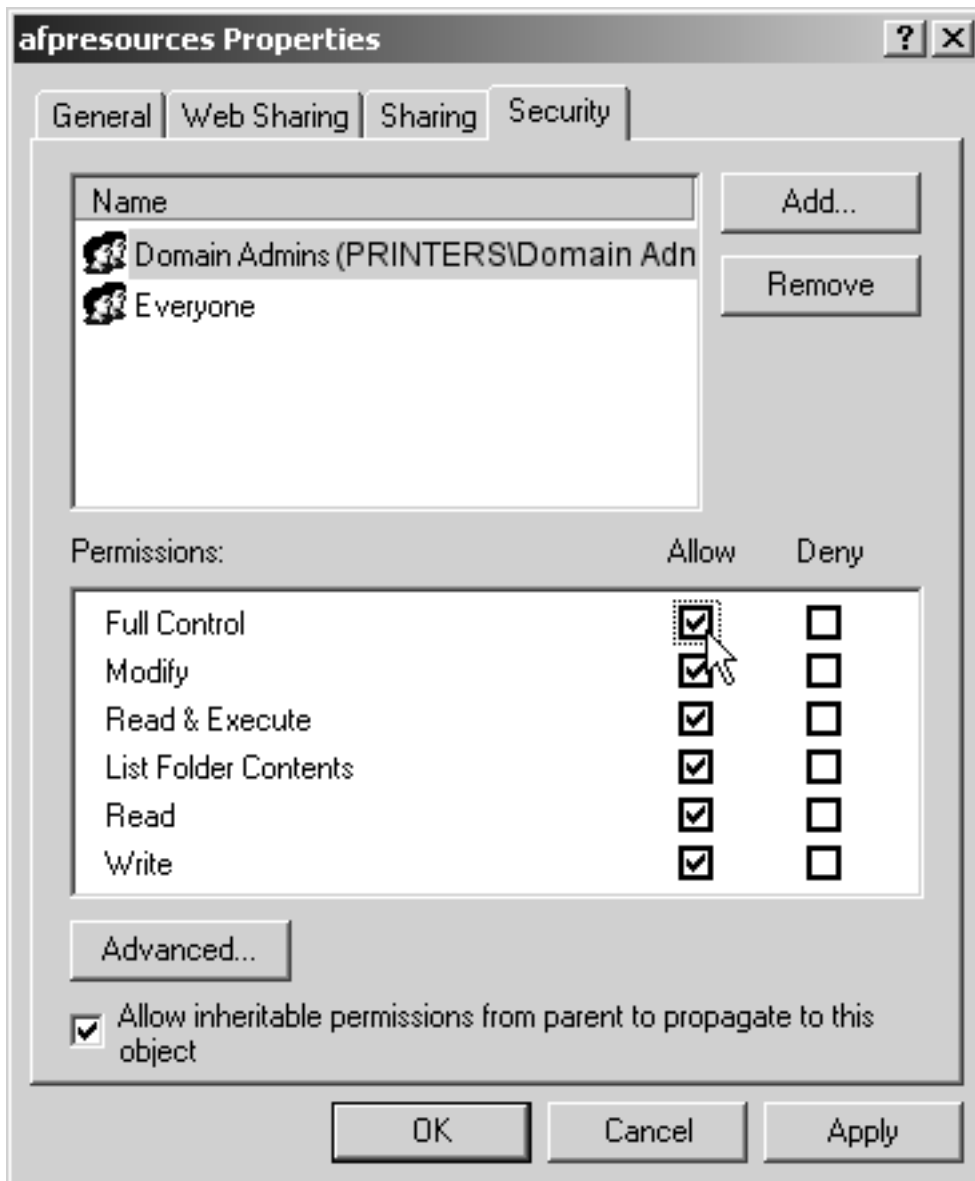


Figure 38. Windows 2000 — Properties dialog: Give Domain Admins Full Control

Note: You may want change the access level for the **Everyone** group as well.

- j. Click the **Sharing** tab.
- k. Select **Share this folder**. Verify the name of the folder in the **Share name** field and type a descriptive comment in the **Comment** field.

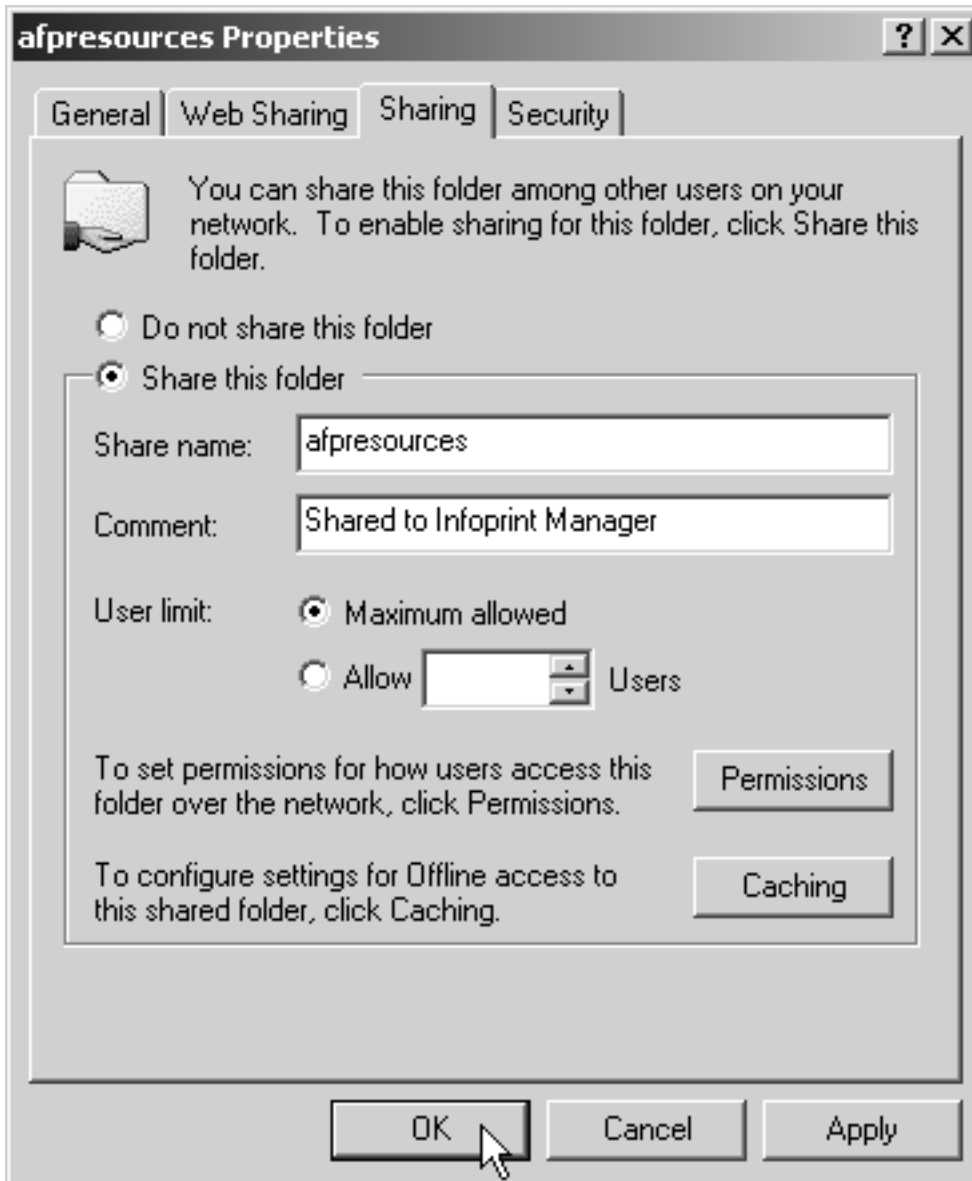


Figure 39. Windows 2000 — Properties dialog: Sharing tab

- l. Click **OK**.
- m. Return to the Management Console on the system that Infoprint Manager is installed on and select **File** → **Start Server** to restart the Infoprint Manager server.

Preparing for shared resources with MVS Download — Windows 2000 only

If you are running Infoprint Manager on Windows 2000 and will be using MVS Download multiple data set support, you must perform an additional task to share resources on a network drive:

1. Click the Windows **Start** button and select **Settings** → **Control Panel** → **Administrative Tools** → **Services**.
2. Select **Infoprint Manager MVS Download**.

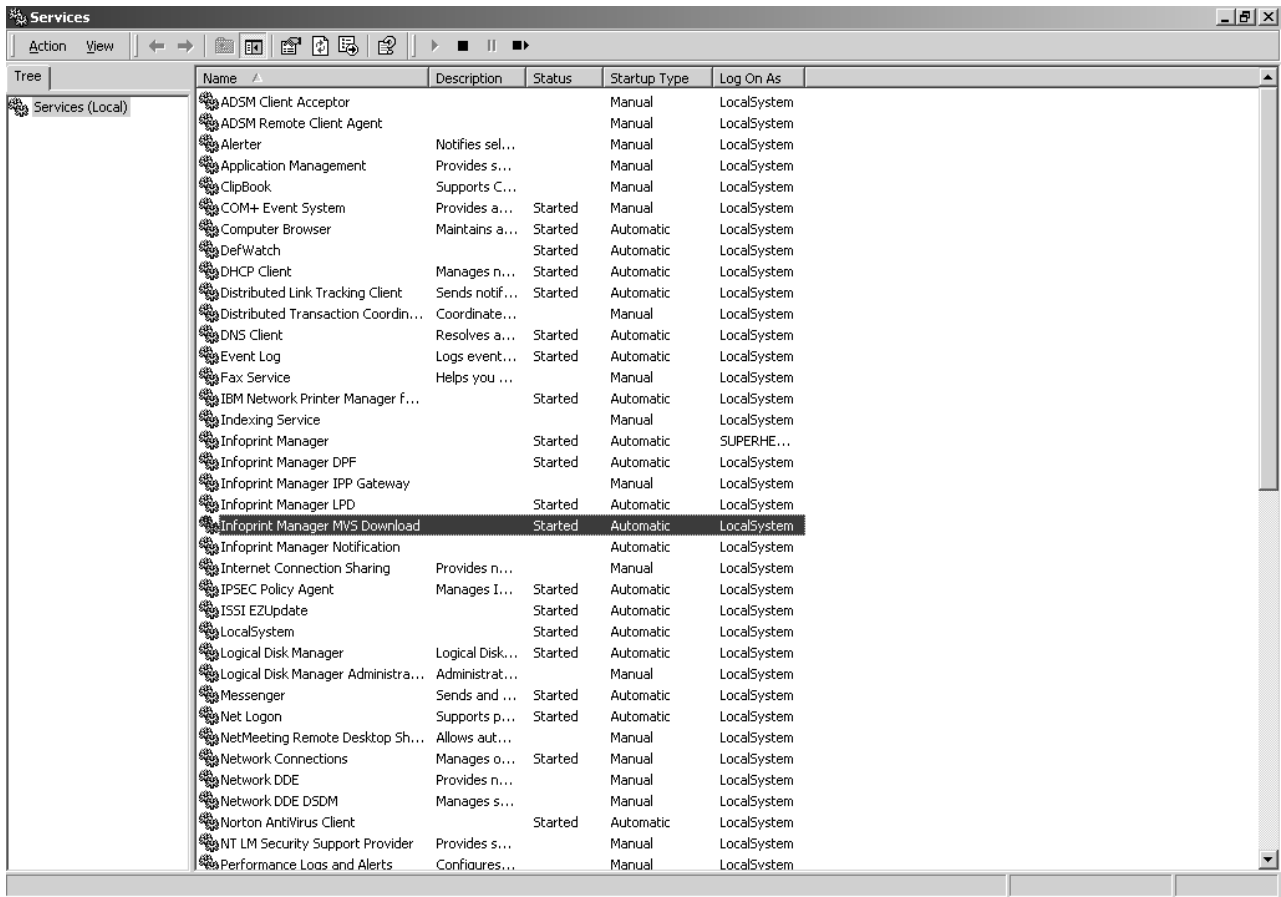


Figure 40. Windows 2000 — Services dialog: Infoprint Manager MVS Download

3. On the **Infoprint Manager MVS Download Properties (Local Computer)** dialog, click the **Log On** tab, and click the **This account** radio button.

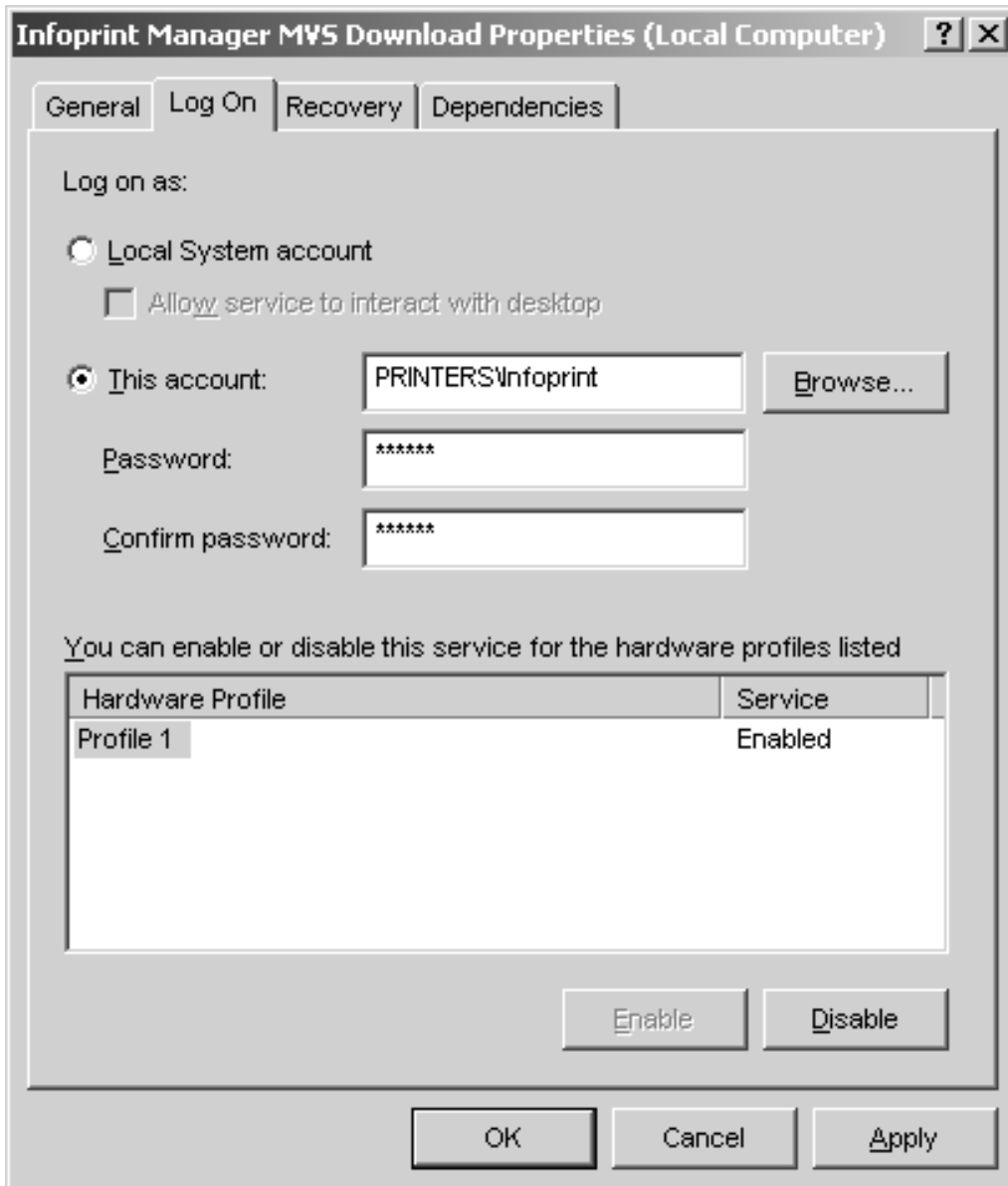


Figure 41. Windows 2000 — Infoprint Manager MVS Download Properties (Local Computer)

4. Enter the account information for the same user you authorized in step 1 on page 66..
5. Click OK.

Telling Infoprint Manager where your remote resources are located

Once you have made the resources available, you have to make sure the Infoprint Manager destinations that will be using them know where they are. Follow the steps below to complete this configuration.

1. On the system that Infoprint Manager runs on, map a network drive to the directory that your resources reside in.
 - a. Open Windows NT/2000 Explorer.
 - b. Select **Tools --> Map Network Drive**.

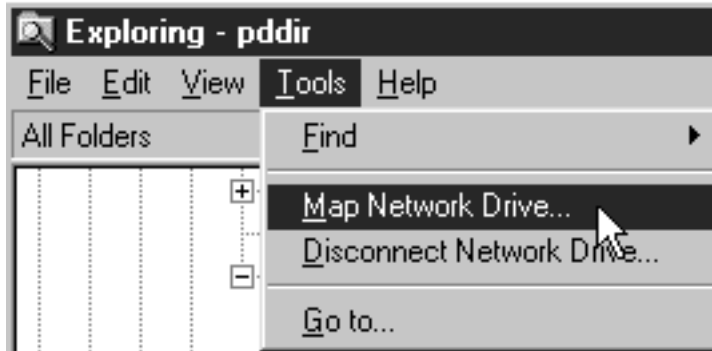


Figure 42. Map network drive

- c. Specify the user and password to use.

Windows NT

- 1) In the **Map Network Drive** dialog,
 - a) In the **Drive** field, select a drive letter that you are not currently using.
 - b) In the **Path** field, type the Universal Naming Convention (UNC) name of the directory you shared on the system that your resources are stored on in this format: `\\resourcesystem\sharename`.

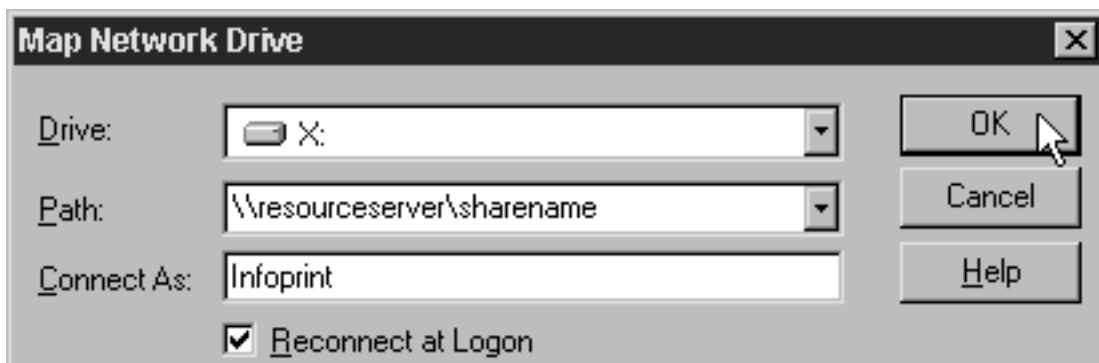


Figure 43. Map to...

- 2) Type the user that you created in step 1 on page 55 in the **Connect as** field, and select **Reconnect at logon**.
- 3) Click **OK**. A dialog will open and will ask you to enter the password for this user.
- 4) Type the password that you set for this user in step 1 on page 55, and click **OK**.

Windows 2000

- 1) In the **Map Network Drive** dialog, select a drive letter that you are not currently using from the drop-down menu in the **Drive** field.

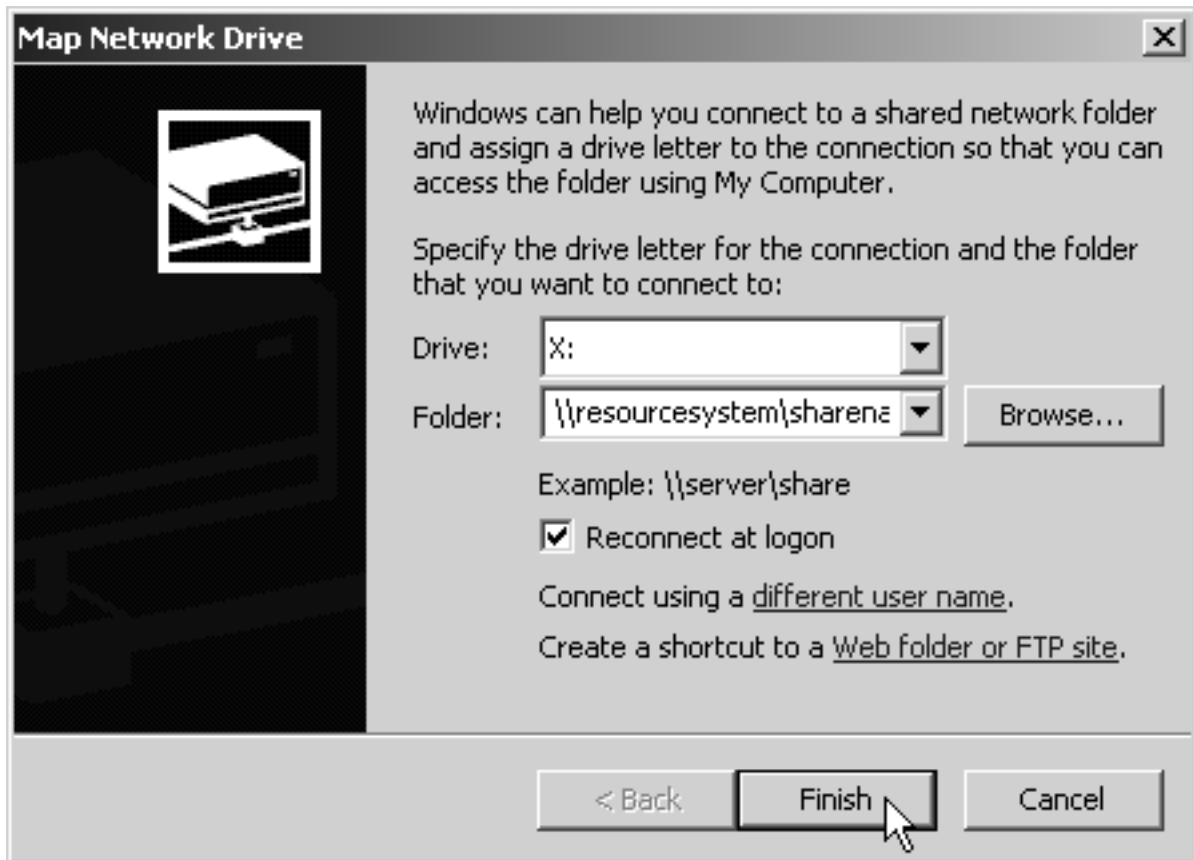


Figure 44. Map Network Drive

- 2) In the **Folder** field, type the Universal Naming Convention (UNC) name of the directory you shared on the system that your resources are stored on server in this format: `\\resourcesystem\sharename`.
- 3) Select **Reconnect at logon**.
- 4) Click the **Connect using a different user name** link.
- 5) In the **Connect As** dialog, type the user name and password that you created in step 1 on page 66.

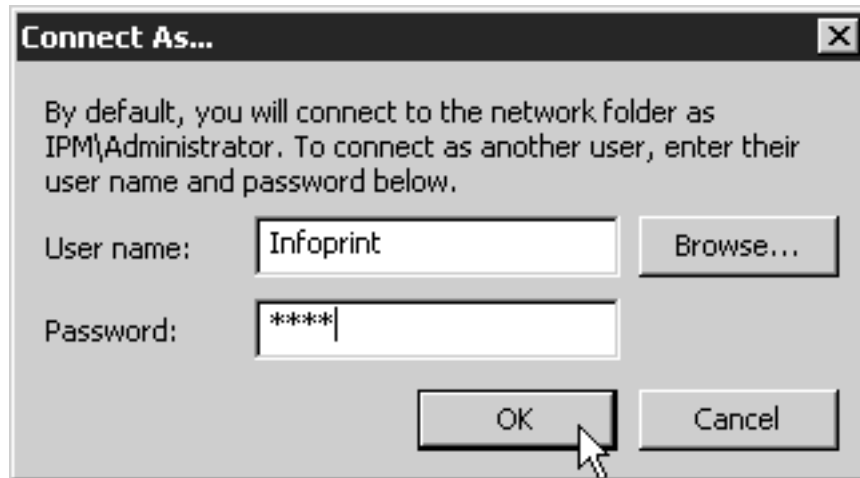


Figure 45. Connect As

- 6) Click **OK** in the **Connect As** dialog.
- 7) Click **Finish** in the **Map Network Drive** dialog.
2. Start the Infoprint Administration GUI.
3. Select one of the printers that needs to access these resources and right-click it.

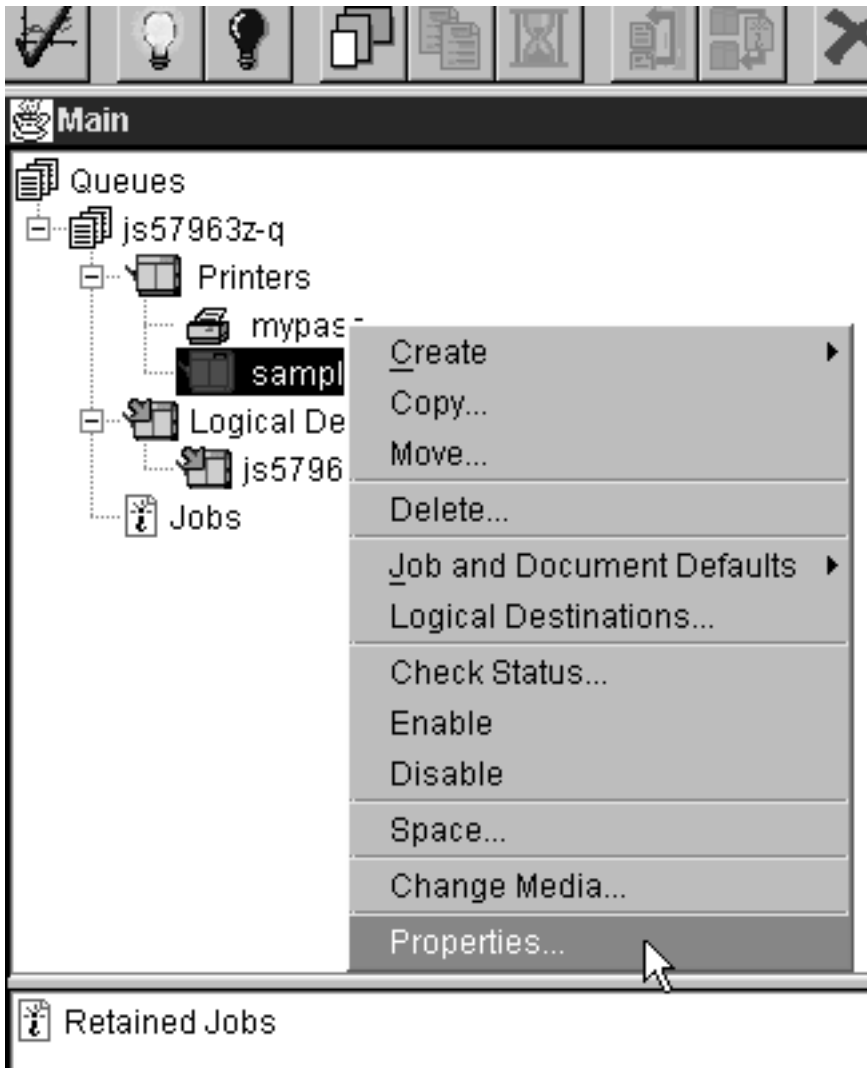


Figure 46. Administration GUI: Printer right-click menu

4. Select **Properties** from the pop-up menu.
5. In the Properties notebook for this printer, click the **AFP Resources** tab.

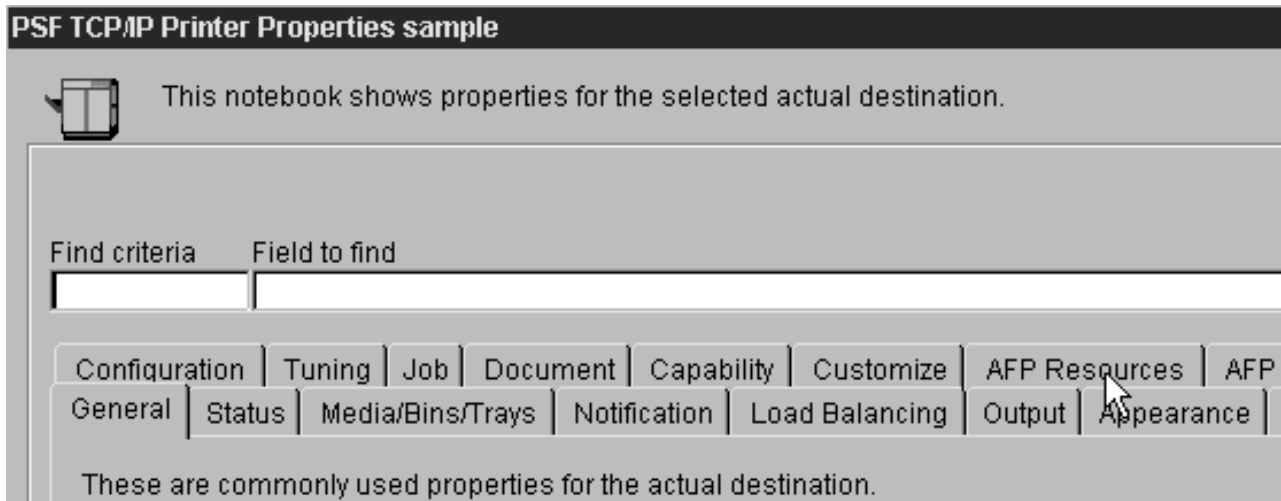


Figure 47. Administration GUI: Printer Properties Notebook

6. Find the correct field for your situation. For example, if your directory only holds fonts, find the **Location of fonts** field; if it holds only overlays, find the **Location of overlays** field. However, if the directory holds a variety of resources, find the **Location of resources** field.

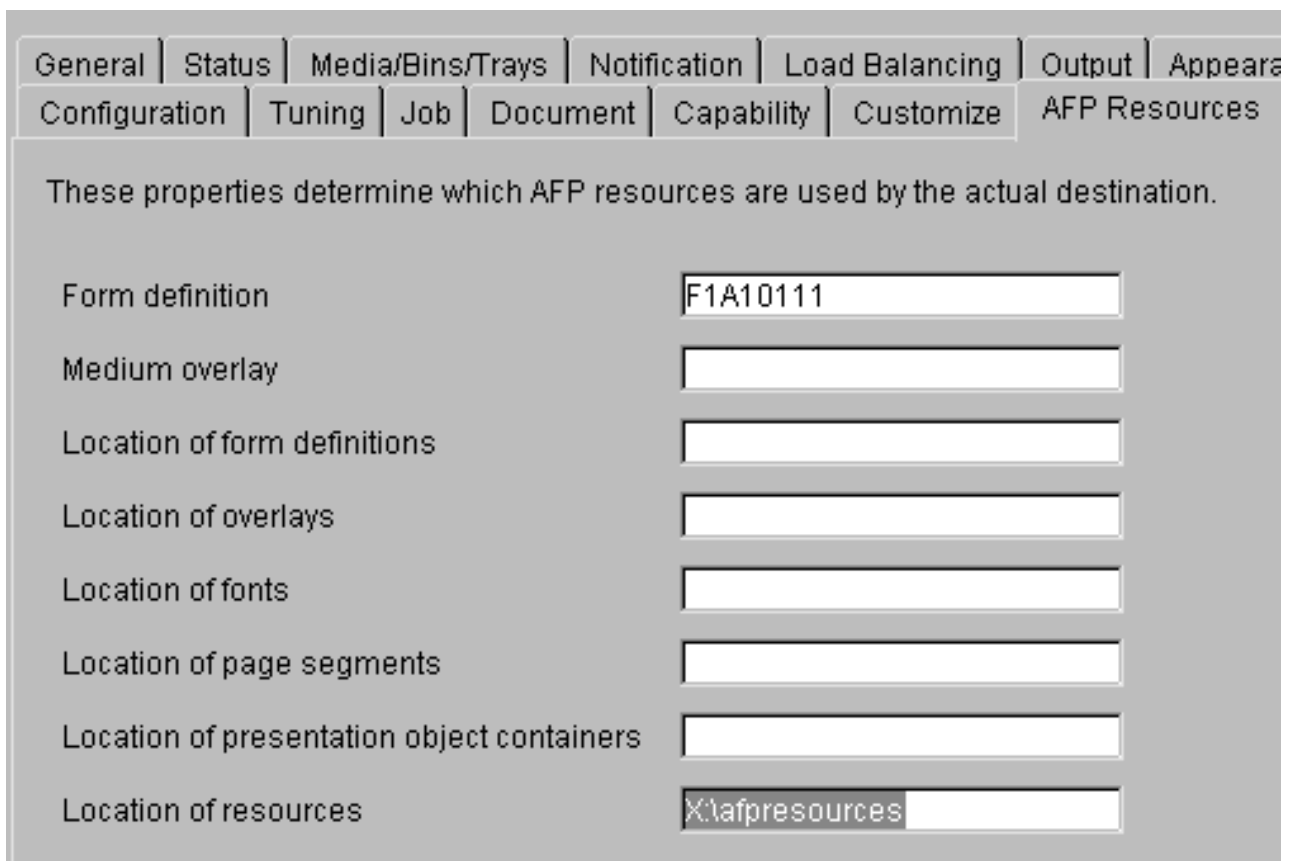


Figure 48. Administration GUI: Printer Properties notebook- AFP Resources tab

7. In the correct field, type the letter of the drive you mapped in step 1c on page 83 and the share name of the directory that contains the resources. For example: x:\afpresources.
8. Click **OK** to close the properties notebook and make the settings take effect.

Chapter 12. Setting up SNMP

About SNMP communication

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

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

Smart-defaulting actual destination attributes

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

- Does **not** explicitly set the attribute when creating or modifying the physical printer
- Resets the attribute to the Infoprint Manager default

Before you create the actual destination and let the server smart-default one or more actual destination attributes for a printer, make sure that the printer is turned on and connected to the network.

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

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

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

Detecting, reporting, and recovering from printer problems

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

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

In addition, the server can more accurately pinpoint the specific problem that caused a condition in an SNMP printer. The conditions are:

- Needs attention:
 - Cover open
 - Offline
 - Out of paper
 - Output tray full
 - Output tray problem
 - Paper jam
- Needs key operator:
 - Needs service
 - Out of toner
 - Output tray missing
- Warning (printing continues but a user needs to perform an action soon):
 - Low on paper
 - Low on toner
 - Needs service soon
 - Tray empty (printing continues with a linked tray)

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

Setting up and using SNMP

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

This section explains:

- Determining if a printer works with SNMP
- Using the GUI
- Procedures

Determining if a printer works with SNMP

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

To determine if a printer works with SNMP:

1. Set the actual destination attribute **destination-tcpip-internet-address** to the IP address of the printer to indicate that Infoprint Manager should try to communicate with the printer using SNMP. The server sets the actual destination attribute **use-snmp** to "true".
2. If the server cannot communicate with the printer using SNMP, it **pings** the printer. If the ping command:

- **Works**, the server concludes that the printer is not an SNMP printer. It sets **use-snmp** to “false”.
 - **Does not work**, the server concludes that the printer is turned off or is not currently connected to the network. It leaves **use-snmp** set to “true” and continues trying to establish an SNMP connection with the printer.
3. Check the actual destination attribute **snmp-active**. If it is “true”, the printer works with SNMP.

Using the GUI

To use the Infoprint Manager Administration GUI to check and set the attributes:

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

Procedures

Do these steps to check and set the attributes:

1. Check these actual destination attributes:
 - **snmp-community-name**
Indicates the name of the SNMP community that the printer belongs to. The default is “public”. The field on the SNMP tab is “Community name”.
 - **snmp-retry-count**
Indicates the number of times the server should retry an SNMP request if it fails. The higher the number, the more network traffic. Using a lower number can improve performance. The default is “2”. The field on the SNMP tab is “Number of times to retry”.
 - **snmp-timeout**
Indicates the number of seconds the server should wait for the printer to respond to an SNMP request. Using a lower number can improve performance. The default is “5”. The field on the SNMP tab is “Timeout”.
 - **use-snmp**
Indicates whether the server should try to use SNMP to communicate with the printer. The default is “true”. The field on the SNMP tab is “Use SNMP”.
2. If the defaults in Step 1 in this section are correct, set this actual destination attribute:
 - **destination-tcpip-internet-address**
Indicates the IP address of the SNMP printer. The field on the tab is “TCP/IP address”.
3. Check this actual destination attribute:
 - **snmp-active**
Indicates whether the server has been able to establish an SNMP session with the printer. The field on the SNMP tab is “SNMP is active”.
4. Set these server attributes:
 - **snmp-discovery-during-startup**
Indicates whether the server queries the printer MIB when Infoprint Manager starts, or whether the server gets the printer status when it periodically polls the printer. The default is “true”, but you might want to set it to “false” if

you have several SNMP printers and you want to make the server start faster. The field on the SNMP tab is “Discover printers at startup”.

- **snmp-normal-poll-interval**

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

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

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

Additional SNMP information

This section briefly explains the:

- Effects on performance with SNMP
- Additional GUI status window

Effects on performance with SNMP

The performance areas that SNMP can influence are:

- Server start-up
- Actual destination queries
- SNMP polling

Server start-up

If you set **snmp-discovery-during-startup** to “true”, the server start-up might be slow because the server queries the printer MIB for the most current information. Setting this attribute to “false” can improve performance. However, the printer status is not as current because the server only gets the printer status when it periodically polls the printer.

Actual destination queries

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

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

SNMP polling

The server periodically polls the SNMP printers to get their latest status. If a printer condition changes while the server is polling a printer (for example, a user closes an open cover), the server might temporarily miss one of the changes. However, during the next polling cycle, the server automatically picks up any changes that it missed.

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

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

| **Additional GUI status window**

| The Infoprint Manager Administration GUI can launch the Status window from
| IBM Network Printer Manager (NPM) for the Web. The window lets you see a
| detailed view of the status of an SNMP printer. It includes a graphic representation
| of the printer and its current status, such as whether it has a paper jam or a door
| open.

| To see the Status window from the GUI:

- | 1. Right-click or use the Printer pull-down menu to select an SNMP printer.
- | 2. Select **Check Status**.
- | 3. Select **More Information**.

Chapter 13. Creating a new resource-context object

You create **resource-context** objects from the **Command Prompt** window; you cannot create them through the **Infoprint Manager Administration GUI**. Use **-c resource-context** with the **pdcreate** command and specify the **context-address**, the server name, and the **resource-context** object name to create a new **resource-context** object. Specify a unique name that describes the **resource-context** object within the server. All **resource-context** objects within a server appear in the selection lists for the fields in the **PSF Resource** tab of the **Printer Properties** window. Two servers can each contain a **resource-context** object with the same name, but different values for their attributes.

For more information about the attributes used in creating resource-context objects, refer to the **Attributes for Resource Contexts** topic in **Chapter 7. Infoprint Object Attributes** of the *Infoprint Manager: Reference*.

Infoprint Manager uses **resource-context** object names as values for the lists associated with the fields in the **Printer Properties** window, or as a valid value you can enter in the fields of the **Modify Job and Document Defaults** notebook. Job submitters can also use them as values for attributes of a document within a job.

For example, to create the **resource-context** object **Resfont1**, which is contained in the Infoprint Manager server **Server1**, to identify the **\resource\fonts\group1** directory as the directory that contains AFP font resources, enter the command:

- `pdcreate -c resource-context`
- `-x "context-address=\resource\fonts\group1"`
- `Server1:Resfont1`

The search order for AFP™ resources

Using a resource-context object is an easy way to ensure that Infoprint Manager locates the AFP resources required to process a job. However, there are a number of ways to override the specifications of the resource-context object and direct Infoprint Manager to use different resources for a given job. The following describes the order in which IBM Infoprint Manager for Windows searches for AFP resources:

1. The path specified by the **resource-context-user** document attribute.
2. The paths specified by one of the following document attributes, depending on the type of resource:
 - **resource-context-font**
 - **resource-context-form-definition**
 - **resource-context-overlay**
 - **resource-context-page-definition**
 - **resource-context-page-segment**

Note: If one of these document attributes contains a value, Infoprint Manager ignores any location specified by the same attribute of a default document object.

3. The path specified by the **resource-context** document attribute.
4. The path specified by the **PSFPATH** environment variable.

5. The paths specified by the resource-context attributes of the actual destination. These attributes have the same names as the document attributes identified in step 2 on page 95.
6. For fonts, the directory where the IBM AFP Font Collection: Fonts for AIX and Windows is installed.

Once a resource is located, the search ends, and the located resource is used to process the job. If Infoprint Manager ultimately cannot find a required resource, it prints an error message advising you of the missing resource. Infoprint Manager reports that the job completed successfully.

The preceding information may make planning for resource-context objects in your Infoprint Manager environment appear complicated. But in most cases, it is actually quite simple. If the only AFP resources you use are the AFP resources that shipped with Infoprint Manager, there is no need to use resource-context objects. All of your resources are located automatically because these resources reside in the **reslib**, **fontlib**, and AFP Font Collection directories described above. If, on the other hand, you have custom AFP resources, make sure that you *do not* store them in the **reslib**, **fontlib**, or AFP Font Collection directories. These directories are controlled by Infoprint Manager, and any custom resources stored in these locations may be corrupted by installing, uninstalling, or servicing Infoprint Manager software. Store any custom AFP resources in a directory outside the Infoprint Manager install path and use resource-context objects to provide access to them.

For more information on resource-context objects, refer to the **Creating a New Resource-Context Object** Administrator procedure in *Infoprint Manager for AIX: Procedures* or on the IBM Printing Systems website at <http://www.ibm.com/printers>. On the Printing Systems homepage, click Infoprint Manager under **Resources for**, then click Infoprint Manager for AIX on the Infoprint Manager Family homepage. Select **Product Library** from the list in the right hand navigation bar, then click the link for **Administrator Procedures**.

Chapter 14. Creating a PSF other-driver printer

Create a PSF other-driver actual destination to represent a PCL printer that receives data streams transformed by PSF through a Windows port monitor.

By setting the **document-formats-ripped-at-destination** actual destination attribute, you can specify particular **document-formats** (such as PostScript and PCL) that will not be transformed by Infoprint Manager. The only exception that would cause Infoprint to transform these **document-formats** occurs when certain job and document attributes have been specified with the job. For example, if you submit a PostScript job that contains the document attributes **form-definition** and **output-format**, Infoprint must transform this job to honor these attributes. For a complete list of the attributes that force Infoprint to transform the job, see “Document formats that cause a job to be transformed” on page 101, “Job attributes that cause a job to be transformed” on page 102, and “PSF header sheet attributes that cause a job to be transformed” on page 103.

Using this attribute allows you to use the same PSF other-driver actual destination for all jobs submitted to that printer, while eliminating the performance overhead of having PCL and PostScript documents transformed into Advanced Function Presentation data, and then into PCL image data.

Before creating a PSF other-driver printer, complete the following worksheet.

Table 9. PSF Other-Driver Printer Worksheet

Wizard Prompt	Description	Value
Name	The name for the actual destination object (8 character limit). Note: This name is case-sensitive.	
Server	The server that controls this actual destination.	

Table 9. PSF Other-Driver Printer Worksheet (continued)

Wizard Prompt	Description	Value
Model	<p>On either a Windows or an AIX server, select the printer model from the pull-down list in the Infoprint Manager Administration GUI.</p> <p>If you are on a Windows server and the printer you are working with is not in the pull-down list, enter the name as it appears on the printer.</p> <p>If you are on an AIX server and the printer you are working with is not in the pull-down list, look in the /usr/lib/lpd/pio/predef directory to find the valid values.</p> <p>Note: While the predef value is not required for IBM Infoprint printers, the predef value is required for all other printers.</p>	
SNMP TCP/IP address	The dotted-decimal IP address used to access this printer. This value is used only for SNMP printers.	
Device or Windows port name	The name of the output device that the selected actual destination represents.	On an AIX system, leave this column blank.
Printer command	The printer command that submits a job to this remote destination. Include the full path name for the print driver that you are using for your print device.	On a Windows system, leave this column blank.
Data stream	The data stream expected by the printing device (for example, PCL4 or PPDS).	
Logical destination	The logical destination that directs jobs to this actual destination.	
Logical destination's server	The server that contains the logical destination.	
Queue	The queue that holds jobs destined for this actual destination.	

To create this type of printer, use the **Create Printer Wizard** in the Infoprint Manager Administration GUI. (On the menu bar, click **Printers**—>**Create**—>**PSF**—>**Other**.) Referring to the worksheet above, provide the

information as prompted by the wizard. For assistance, refer to the Infoprint Manager Administration GUI online help.

If you did not enable the printer with the **Create Printer Wizard**, use the Infoprint Manager Administration GUI to enable it.

Chapter 15. Using a PSF other-driver for printing all jobs from a single actual destination

By setting the **document-formats-ripped-at-destination** actual destination attribute, you can specify particular **document-formats** (such as PostScript and PCL) that will not be transformed unless an attribute that requires the transform is specified. You should specify any **document-formats** that are supported by your particular printer. Using this attribute allows you to use the same PSF other-driver actual destination for all jobs submitted to that printer, while eliminating the performance overhead of having PCL and PostScript documents transformed twice.

Configuring the printer

First, follow the directions under Creating a PSF Other-Driver Printer. Then use the following procedure to prepare your printer to accept jobs from a variety of data types.

1. From the main menu, select **Printer ->Properties**.
2. Click the **Document** tab.
3. Highlight both **PCL** and **PostScript** (or only one of the two, depending upon what data types your actual destination supports natively) from the **Formats to RIP at destination** list of Possible Values, and click the **Add** button.

Note: You can specify other values besides **PCL** and **PostScript** because you may have a destination that can accept them directly. For example, your printer may accept **ASCII**.

Once you have completed this task, you are ready to submit jobs to this printer.

Document formats that cause a job to be transformed

The following document attributes negate the **Document Formats Ripped at Destination** setting and force jobs to be transformed. For more information on these attributes, refer to the *Infoprint Manager: Reference*.

- base-printer
- carriage-control-type
- chars
- color-mapping-table
- convert-to-ebcdic
- data-fidelity-problem-reported
- default-printer-resolution
- document-finishing
- font-fidelity-action
- font-processing-messages
- font-resolution
- form-definition
- halftone
- image-fit
- image-length

- image-out-format
- image-width
- input-exit
- input-tray-select
- jog-between-job-copies
- maximum-messages-printed
- maximum-transform-pages-ahead
- mvs-segment-id
- new-line-option
- number-up
- other-transform-options
- output-appearance
- output-bin
- output-format
- overlay
- overlay-back
- overlay-front
- page-definition
- page-media-select
- page-select
- resource-context
- resource-context-font
- resource-context-form-definition
- resource-context-overlay
- resource-context-page-definition
- resource-context-page-segment
- resource-context-user
- resource-exit
- scanner-correction
- screen-frequency
- segment-file-size
- shared-formdef
- shift-out-shift-in
- start-on-new-sheet
- table-reference-characters
- transform-output-file-name
- x-image-shift
- x-image-shift-back
- y-image-shift
- y-image-shift-back

Job attributes that cause a job to be transformed

The following job attributes negate the **Document Formats Ripped at Destination** setting and force the jobs to be transformed. For more information on these attributes, refer to the *Infoprint Manager: Reference*.

- delete-segment-list
- job-finishing
- job-rip-action
- optimize-for-multiple-copies
- tape-exit
- tape-format
- tape-labeled
- tape-max-block-size
- tape-rewind-before
- tape-rewind-unload

PSF header sheet attributes that cause a job to be transformed

These attributes could effect header sheet generation, so they also negate the **Document Formats Ripped at Destination** setting and force jobs to be transformed. For more information on these attributes, refer to the *Infoprint Manager: Reference*.

- account-text
- address1-text
- address2-text
- address3-text
- address4-text
- building-text
- department-text
- mvs-class
- mvs-destination
- mvs-forms
- name-text
- node-id-text
- programmer-text
- room-text
- subject-text
- title-text
- user-id-text

Chapter 16. Using PSF DSS user-exit programs

This topic describes the sample PSF DSS user-exit programs supplied with Infoprint Manager and shows how you can create your own PSF DSS user-exit programs. These user-exit programs apply to printers connected to Infoprint Manager through the PSF device support subsystem (DSS).

This chapter contains the following topics:

- “Supported types of PSF DSS user-exits”
- “Sample PSF DSS user-exit programs” on page 106
- “Creating and using your own PSF DSS user-exit programs” on page 107
- “Compiling and installing the user-exit program” on page 107
- “Header page, trailer page, and separator page user-exit program inputs and outputs” on page 108
- “Accounting, post-print accounting, and audit user-exit program inputs and outputs” on page 112
- “Input data user-exit program inputs and outputs” on page 115
- “Output data user-exit program inputs and outputs” on page 116
- “Structure of a user-exit program” on page 117
- “User-exit programs for the line data transform” on page 118
- “Attributes of the line data input file” on page 124
- “Using the iconv command to convert code pages” on page 125

Supported types of PSF DSS user-exits

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

Infoprint Manager allows eight different types of PSF DSS user exits. Infoprint Manager calls the exits in the following order:

1. Header (Start) page user exit
2. Separator page user exit
3. Accounting user exit
4. Audit user exit
5. Input data user exit
6. Output data user exit
7. Trailer (End) page user exit
8. Post-print accounting user exit

Note: Refer to Chapter 20, “Setting up to use MVS Download” on page 151 for information about the sample user-exit program provided for MVS Download.

Sample PSF DSS user-exit programs

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

If you choose to modify the existing user-exit programs, the source code for each one is located in the *install_path*\exits directory. The *install_path* is the directory that Infoprint Manager is installed in. If you don't know the install path, you can find it in the **Management Console**. Open the Management Console, click **Edit -> Change Service Configuration** and find the **Install path** field.

Table 1 lists the sample user exit programs supplied with Infoprint Manager.

Table 10. Sample user-exit programs

Type of user exit	File Name	Description
Header Page	ainuxhdr.c	Generates brief style start sheet.
	ainuxhdr2.c	Generates full style start sheet.
	ainuxhdrp.c	Generates start sheet with job ticket information.
	ainuxhdrx.c	Generates start sheet without vertical lines.
	pduxblkh.c	Generates a blank start sheet.
Separator Page	ainuxsep.c	Generates brief style separator sheet
	ainuxsep2.c	Generates full style separator sheet.
	ainuxsepp.c	Generates separator sheet with job ticket information.
	ainuxsepx.c	Generates separator sheet without vertical lines.
	pduxblks.c	Generates a blank separator sheet.
Trailer Page	ainuxtlr.c	Generates brief style end sheet.
	ainuxtlr2.c	Generates full style end sheet.
	ainuxtlrp.c	Generates start sheet with job ticket information.
	ainuxtlrx.c	Generates end sheet without vertical lines.
	pduxblkt.c	Generates a blank end sheet.

Table 10. Sample user-exit programs (continued)

Type of user exit	File Name	Description
Accounting	ainuxacc.c	Generates brief style accounting sheet.
	ainuxacc2.c	Generates full style accounting sheet.
	ainacclog.c	Generates accounting log information.
Post-print Accounting	ainuxjobcompletion.c	Generates accounting log information on printed ("stacked") sheets
Auditing	ainuxaud.c	Generates brief style audit sheet.
	ainuxaud2.c	Generates full style audit sheet.
	ainuxaudp.c	Generates audit sheet with job ticket information.
	ainaudlog.c	Generates audit log information.
Accounting log	ainuxaccp.c	Generates accounting sheet log with job ticket information.
Input Data	ainuxind.c	Provides template for writing your own user-exit program.
Output Data	ainuxout.c	Provides template for writing your own user-exit program.

Note: Refer to "User-exit programs for the line data transform" on page 118 for sample user-exit programs used with the line data transform.

Creating and using your own PSF DSS user-exit programs

You can create your own PSF DSS user-exit program at any time; however, you must perform additional tasks to produce a PSF DSS user-exit executable program. You must have the Microsoft Visual C++ Program, Version 5.0 or higher, installed on your Infoprint NT server in order to compile and build user-exit programs. Be sure to apply the latest service available.

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

- Copy one of the sample user-exit programs, rename it, and modify it using the Microsoft Visual C++ Program.
- Write your own user-exit program. You can include the variable data shown in the following sections, for example, USERID; but Infoprint cannot process variable data that is not shown in this topic.

Compiling and installing the user-exit program

You can only specify one user-exit program for each function name provided by Infoprint per actual destination.

The function name of the user exit must be one of the following (in uppercase), because that is the entry point in the module:

- ACCOUNTING
- AUDIT
- JOBCOMPLETION
- HEADER
- SEPARATOR
- TRAILER
- INDATA
- OUTDATA

Activating the user-exit program

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

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

To perform this task from the command line, use the click the Windows **Start** button and select **Programs—>Command Prompt** path and perform the following procedure.

1. To create the auxiliary-sheet object, type:

```
pdcreate -c aux-sh -x psf-exit-prog-name=fullpath\exit_name  
server1:auxiliary_sheet_name
```
2. Disable the actual destination (in this case, prt1).

```
pddisable prt1
```
3. Add this new auxiliary-sheet object (in this case, an audit exit) to the actual destination.

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

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

```
pdset -c auxiliary-sheet -x  
psf-post-print-accounting-name=ainuxjobcompletion servername:  
accounting-log
```

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

```
pdenable prt1
```

Header page, trailer page, and separator page user-exit program inputs and outputs

Because the Infoprint start page and trailer page user-exit programs require the same inputs and produce the same outputs, the program descriptions are similar. The Infoprint separator page user-exit program differs only in having access to the copy counter output information.

There is one invocation of the separator user-exit prior to every copy of the job data. This assumes that the copies were requested using the result-profile attribute rather than the copy-count attribute (for which the multiple copies are merged into one job file).

The source code (in the C programming language) for the sample header page user-exit programs, trailer page user-exit programs, and separator page user-exit programs are located in the *install_path*\exits directory. These sample programs generate an AFP data stream page.

The data structures for the header, trailer, and separator page user-exit programs are included with the source code in the *install_path*\exits\ainuexit.h file. The code for these structures is shown in “Structure of a user-exit program” on page 117.

The declaration of these exits is:

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

The **HEADER_EXITDATA**, **TRAILER_EXITDATA**, and **SEPARATOR_EXITDATA** input/output parameters contain all the input and output data needed to communicate between Infoprint and the user-exit programs.

The **HEADER_EXITDATA**, **TRAILER_EXITDATA**, and **SEPARATOR_EXITDATA** structures include these fields.

To build the executable program, you must use the Microsoft Visual C++ Program, Version 5.0 pull-down menu.

Fields that provide input information

UserID

The system user ID of the person who submitted the job.

NodeID

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

JobName

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

Job submitters can also specify the job name with the Infoprint job attribute `job-name=name` or `name=name` when they submit a job. If the job name is not specified for files defined as PCL, PostScript, line data, unformatted ASCII, double-byte character set (DBCS) ASCII, or PDF, when the job is submitted, Infoprint creates a temporary file for the transform output. Infoprint uses the temporary file name for the job name.

SpoolID

The job ID, which is an integer. Infoprint Manager sets the **SpoolID**.

PrinterName

The name of the Infoprint destination.

hab This field is not used.

Date The date the job is printed in MM/DD/YY format.

ExtDate

The date the job is printed in MM/DD/YYYY format.

Time The time the job is printed in HH:MM:SS format.

Distribution

The distribution information provided when the job is submitted.

Other input values

The following fields are used by some of the sample programs to provide additional input information. This information can be specified from:

- migrated jobs
- MVS Download jobs
- fields set in the Infoprint Manager Administration GUI **default-document** notebook on the **PSF Information** tab

The field length limits provided below are based on the currently shipped user-exits. Additional field length is normally provided to the user-exits, but would require modification of the exits to display the additional data.

Account

Identifies the account information provided when the job is submitted. The **Account** field value can be any null-terminated (X'00') character string. You should limit the string to 20 characters or less. Job submitters can specify this same information through the **account-text** document attribute.

Address1

Identifies the first line of address information provided when the job is submitted. The **Address1** field value can be any null-terminated (X'00') character string. You should limit the string to 57 characters or less. Job submitters can specify this same information through the **address1-text** document attribute.

Address2

Identifies the second line of address information provided when the job is submitted. The **Address2** field value can be any null-terminated (X'00') character string. You should limit the string to 57 characters or less. Job submitters can specify this same information through the **address2-text** document attribute.

Address3

Identifies the third line of address information provided when the job is submitted. The **Address3** field value can be any null-terminated (X'00') character string. You should limit the string to 57 characters or less. Job submitters can specify this same information through the **address3-text** document attribute.

Address4

Identifies the fourth line of address information provided when the job is submitted. The **Address4** field value can be any null-terminated (X'00') character string. You should limit the string to 57 characters or less. Job submitters can specify this same information through the **address4-text** document attribute.

Building

Identifies the building information provided when the job is submitted. The **Building** field value can be any null-terminated (X'00') character string. You should limit the string to 24 or less characters. Note that job submitters can specify this same information through the **building-text** document attribute.

Department

Identifies the department information provided when the job is submitted. The **Department** field value can be any null-terminated (X'00') character

string. You should limit the string to 24 characters or less. Job submitters can specify this same information through the **department-text** document attribute.

Name Identifies the name information provided when the job is submitted. The **Name** field value can be any null-terminated (X'00') character string. You should limit the string to 24 characters or less. Job submitters can specify this same information through the **name-text** document attribute.

Nodeid

Identifies the nodeid information provided when the job is submitted. The **Nodeid** field value can be any null-terminated (X'00') character string. You should limit the string to 10 characters or less. Job submitters can specify this same information through the **node-id-text** document attribute.

Passthru

Identifies any other information provided when the job is submitted that will be passed through the user exit to the backend program. Job submitters can also specify the account information with the Infoprint attribute **printer-pass-through='-pa=class=A...'**

The following **Passthru** flags are extracted by the "full" user-exits. Extracting other values from the **Passthru** variable would require modifications to the supplied user-exit programs.

class Identifies the one-character class attribute.

destination

Identifies the one- to eight-character destination attribute.

forms Identifies the one- to eight-character forms attribute.

segmentid

Identifies the one- to ten-character segmentation identifier.

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

Programmer

Identifies the programmer information provided when the job is submitted. The **Programmer** field value can be any null-terminated (X'00') character string. You should limit the string to 24 characters or less. Job submitters can specify this same information through the **programmer-text** document attribute.

Room Identifies the room information provided when the job is submitted. The **Room** field value can be any null-terminated (X'00') character string. You should limit the string to 24 characters or less. Job submitters can specify this same information through the **room-text** Infoprint document attribute.

Title Identifies the title information provided when the job is submitted. The **Title** field value can be any null-terminated (X'00') character string. You should limit the string to 24 characters or less. Job submitters can specify this same information through the **title-text** document attribute.

Userid

Identifies the userid information provided when the job is submitted. The **Userid** field value can be any null-terminated (X'00') character string. You should limit the string to 10 characters or less. Job submitters can specify this same information through the **user-id-text** document attribute.

Fields that provide output information

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

PagePointer

Points to a buffer containing the header page or trailer page data produced by the exit.

PageSize

Indicates the size of the header page or trailer page data returned by the exit. This field is initially set to 0 before the exit is called. If no data is generated by the exit, this field remains set at 0.

PageType

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

Valid values are:

0 Advanced Function Presentation (AFP) data stream.

1 ASCII data.

Job Completion

Indicates the status returned by the exit. This field is initially set to 0 before the exit is called.

Accounting, post-print accounting, and audit user-exit program inputs and outputs

The Infoprint accounting, post-print accounting, and audit user-exit programs require the same inputs and produce the same outputs so the program descriptions are similar.

The source code (in the C programming language) for the sample separator page user-exit programs listed in “Supported types of PSF DSS user-exits” on page 105 are located in the *install_path*\exits directory.

These sample programs generate an ASCII data stream page or a log file. The sample programs store the data into a file you can access with the following executable reporting utilities:

ainurpt1

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

ainurpt2

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

ainurpt3

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

ainurpt4

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

ainurpt5

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

ainurpt6

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

ainurpt7

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

ainurpt8

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

ainurpt9

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

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

Note: You must activate these exits to generate the reports. For more information on activating these exits, see "Activating the user-exit program" on page 108.

The data structures for the accounting and audit user-exit programs are included with the source code in the *install_path*\exits\ainuexit.h and *install_path*\exits\ainurpt.h files. The code for these structures is shown in "Structure of a user-exit program" on page 117.

The declarations of these exits are:

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

The **ACCOUNTING_EXITDATA** and the **AUDIT_EXITDATA** input and output parameters contain all the input and output data needed to communicate between Infoprint and the user-exit programs.

The **ACCOUNTING_EXITDATA** and the **AUDIT_EXITDATA** structures include the following fields.

To build the executable program, you must use the Microsoft Visual C++ Program, Version 5.0 pull-down menu.

Fields that provide input information

UserID

The system user ID of the person who submitted the job.

NodeID

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

JobName

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

Job submitters can also specify the job name with the Infoprint job attribute **job-name=name** or the alias **name=** when they submit a job. If the job name is not specified for files defined as PCL, PostScript, line data, unformatted ASCII, double-byte character set (DBCS) ASCII, or PDF, files when the job is submitted, Infoprint creates a temporary file for the transform output. Infoprint uses the temporary file name for the job name.

SpoolID

The job ID, which is an integer. Infoprint sets the **SpoolID**.

PrinterName

The name of the Infoprint destination.

hab This function is not used.

Pages Printed (Projected)

Shows the total number of impressions produced for this job.

Bin One Sheets Printed (Projected)

Shows the total number of sheets selected from the primary bin.

Bin Two Sheets Printed (Projected)

Shows the total number of sheets selected from the secondary bin.

Number of Fonts used

Shows the total number of fonts used in this job.

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

Number of Overlays used

Shows the total number of overlays used in this job.

Number of Segments used

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

ExtStart Date

Identifies the four-digit date the job started processing.

Start Date

The two-digit date the job started processing.

Start Time

The time the job started processing.

ExtStop Date

The four-digit date the job finished processing.

Stop Date

The two-digit date the job finished processing.

Stop Time

The time the job finished processing.

Optional fields that can provide other input values for the accounting and auditing user exit are listed in "Other input values" on page 110.

Fields that provide output information

PagePointer

Points to a buffer containing the accounting or audit page data produced by the exit.

PageSize

Indicates the size of the accounting or audit page data returned by the exit. This field is initially set to 0 before the exit is called. If no data is generated by the exit, this field remains set at 0.

PageType

Indicates the type of accounting or audit page data (if any) that the exit generates. The field is initially set to 0 (AFP data stream) before the exit is called.

Valid values are:

- 0 Advanced Function Presentation (AFP) data stream
- 1 ASCII data

Input data user-exit program inputs and outputs

The Infoprint Manager input data user-exit program is used to monitor the print data stream that is coming into Infoprint Manager. This exit is called at the beginning of the job after the header (start) page.

The source code (in the C programming language) for the sample input data user-exit program listed in “Supported types of PSF DSS user-exits” on page 105 is in the *install_path*\exits\ainuxind.c file. This sample program consists of a return code and performs no function.

The data structures for the output data user-exit program are included with the source code in the *install_path*\exits\ainuexit.h file. The code for these structures is shown in “Structure of a user-exit program” on page 117.

The declaration of this exit is:

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

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

The `INDATA_EXITDATA` structure includes the following fields.

To build the executable program, you must use the Microsoft Visual C++ Program, Version 5.0 pull-down menu.

Fields that provide input information

UserID

The system user ID of the person who submitted the job.

NodeID

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

JobName

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

Job submitters can also specify the job name with the Infoprint job attribute **job-name=name** when they submit a job. If the job name is not specified for files defined as PCL, PostScript, line data, unformatted ASCII, double-byte character set (DBCS) ASCII, or PDF, files when the job is submitted, Infoprint creates a temporary file for the transform output. Infoprint uses the temporary file name for the job name.

SpoolID

The job ID, which is an integer. Infoprint Manager sets the **SpoolID**.

PrinterName

The name of the Infoprint destination.

hab This function is not used.

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

DataOffset

Indicates the offset into the source file.

DataSize

Indicates the number of bytes in the buffer.

DataPointer

Points to a buffer containing the incoming print data stream.

DataType

Indicates the type of source data the exit receives.

Valid values are:

0 Advanced Function Presentation (AFP) data stream.

1 ASCII data.

Output data user-exit program inputs and outputs

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

The source code (in the C programming language) for the sample input data user-exit program listed in "Supported types of PSF DSS user-exits" on page 105 is in the *install_path*\exits\ainuxout.c file. This sample program consists of a return code and performs no function.

The data structures for the output data user-exit program are included with the source code in the *install_path*\exits\ainuexit.h file. The code for these structures is shown in "Structure of a user-exit program" on page 117.

The declaration of this exit is: void OUTDATA (OUTDATA_EXITDATA \exitdata)

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

To build the executable program, you must use the Microsoft Visual C++ Program, Version 5.0 pull-down menu.

The **OUTDATA_EXITDATA** structure includes the following fields.

Fields that provide input information

UserID

The system user ID of the person who submitted the job.

NodeID

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

JobName

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

Job submitters can also specify the job name with the Infoprint job attribute **job-name=name** when they submit a job. If the job name is not specified for files defined as PCL, PostScript, line data, unformatted ASCII, double-byte character set (DBCS) ASCII, or PDF, files when the job is submitted, Infoprint creates a temporary file for the transform output. Infoprint uses the temporary file name for the job name.

SpoolID

The job ID, which is an integer. Infoprint sets the **SpoolID**.

PrinterName

The name of the Infoprint destination.

hab This function is not used.

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

DataSize

Indicates the number of bytes in the buffer.

DataPointer

Points to a buffer containing the incoming print data stream.

Note: The output information fields used by some of the sample programs (“Other input values” on page 110) are available through the alternate sample user exits. Infoprint does not provide an alternate Output Data User Exit.

Structure of a user-exit program

All input and output variables used in Infoprint user exits are defined through either of the following two files:

- *install_path*\exits\ainuexit.h or
- *install_path*\exits\ainurpt.h

The ainuexit.h file contains definitions for the header, trailer, separator, accounting, audit, and post-print accounting exits, while the ainurpt.h file formats data for the following executable reporting utilities:

- Accounting Log – (**ainurpt1; ainurpt2; ainurpt3**)
- Audit Log – (**ainurpt4; ainurpt5; ainurpt6**)
- Post-print Accounting Log – (**ainurpt7; ainurpt8; ainurpt9**)

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

User-exit programs for the line data transform

Infoprint provides several sample user-exit programs for the line-data transform. Use of the user exits is optional. You specify the names of the exit programs with the **inpxit**, **indxexit**, **outexit**, and **resexit** keywords.

Infoprint provides the following sample programs:

install_path\exits\acif\apkinp.c
Input-record user exit

install_path\exits\acif\apkout.c
Output-record user exit

install_path\exits\acif\apkres.c
Resource exit

In addition, Infoprint provides the following input-record user exits programs to translate input data streams:

- apka2e
- asciinp.c
- asciinpe.c

apka2e

install_path\exits\acif\apka2e.c
Converts ASCII stream data to EBCDIC stream data.

The **apka2e** input-record exit program translates data that is encoded in ASCII (code set IBM-850) into EBCDIC (code set IBM-037) encoded data. You should use this exit when your job requires fonts such as GT12, which has only EBCDIC code points defined.

When you apply PTF UR52401 (APAR IR43438), this program also converts the ASCII code page file (ibm-850) into a corresponding EBCDIC file (ibm-037) referring to files that reside in the *install_path\uconvtab* directory. You can use the Microsoft Visual C++ Program, Version 5.0 and edit this exit to convert any code page from ASCII to EBCDIC. To perform this conversion from a Command Prompt window, see "Using the iconv command to convert code pages" on page 125.

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

```
inpxit=install_path\bin\apka2e  
cc=yes  
cctype=z
```

asciinp.c

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

The **asciinp** input-record exit program transforms an ASCII data stream into a record format that contains a carriage control character in byte 0 of every record. If

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

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

```
inpexit=install_path\bin\asciinp
cc=yes
cctype=z
```

asciinpe.c

install_path\exits\acif\asciinpe.c

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

The **asciinpe** input-record exit program combines both of the user input-record exits described above. To execute, specify `inpexit=install_path\bin\asciinpe` as the exit program in the keyword file and follow the directions specified for both **apka2e** and **asciinp**.

When you apply PTF UR52401 (APAR IR43438), this program also converts the ASCII code page file (ibm-850) into a corresponding EBCDIC file (ibm-037) referring to files that reside in the *install_path*\uconvtab\ directory. You can use the Microsoft Visual C++ Program, Version 5.0 and edit this exit to convert any code page from ASCII to EBCDIC. To perform this conversion from a Command Prompt window, refer to "Using the iconv command to convert code pages" on page 125.

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

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

In addition to the information above, refer to the prolog of the **asciinpe.c** source file that is provided with Infoprint Manager in the *install_path*\exits\acif directory for more information on using and modifying these programs.

The C language header file for all ACIF exit programs is provided in *install_path*\exits\acif\apkexits.h along with the build rules for the ACIF user exits in *install_path*\exits\acif\Makefile.

To build the executable program, you must use the Microsoft Visual C++ Program, Version 5.0 pull-down menu.

For more information about compiling user-exit programs, refer to "Compiling and installing the user-exit program" on page 107.

Input record exit

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

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

The following example provides a sample C language header that describes the control block that is passed to the exit program:

```
typedef struct_INPEXIT_PARMS /* Input record exit Parameters */
{
char *work; /*Address of 16-byte static work area */
PFATTR *pfattr; /*Address of print file attribute information */
char record; /*Address of the input record */
void reserved1; /*Reserved for future use */
unsigned short recordln; /*Length of the input record */
unsigned short reserved2; /*Reserved for future use */
char *work; /*Add, delete or process the record*/
char *eof; /*EOF indicator */
}
INPEXIT_PARMS
```

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

work (Bytes 1–4)

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

pfattr (Bytes 5–8)

A pointer to the print file attribute data structure. Refer to “Attributes of the line data input file” on page 124 for more information on the format of this data structure and the information it contains.

record (Bytes 9–12)

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

reserved1 (Bytes 13–16)

These bytes are reserved for future use.

recordln (Bytes 17–18)

Specifies the number of bytes (length) of the input record. If the input record is modified, this parameter must also be updated to reflect the actual length of the record.

reserved2 (Bytes 19–20)

These bytes are reserved for future use.

request (Byte 21)

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

X'00' Specifies that the line-data transform should process the record.

X'01' Specifies that the line-data transform should not process the record.

X'02' Specifies that the line-data transform should process the record and then return control to the exit program to allow it to insert the next record. The exit program can set this value to save the current record, insert a record, and then supply the saved record at the next call. After the exit inserts the last record, the exit program must reset the request byte to X'00'.

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

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

eof (Byte 22)

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

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

Y Specifies that eof has been encountered.

N Specifies that eof has not been encountered.

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

Output record exit

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

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

The following example contains a sample C language header that describes the control block passed to the exit program:

```
typedef struct_OUTEXIT_PARAMS /* Output record exit Parameters */
{
char *work; /*Address of 16-byte static work area*/
PFATTR *pfattr; /*Address of print file attribute information */
char *record; /*Address of the output record*/
void reserved1; /*Reserved for future use */
unsigned short recordln; /*Length of the input record */
char request; /*Add, delete or process the record */
char *eof; /*EOF indicator */
}
OUTEXIT_PARAMS
```

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

work (Bytes 1–4)

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

pfattr (Bytes 5–8)

A pointer to the print file attribute data structure. Refer to “Attributes of the line data input file” on page 124 for more information on the format of this data structure and the information it contains.

record (Bytes 9–12)

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

recordln (Bytes 13–14)

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

request (Byte 15)

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

X'00' Specifies that the line-data transform should process the record.

X'01' Specifies that the line-data transform should not process the record.

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

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

eof(Byte 16)

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

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

Y Specifies that the last record has been written.

N Specifies that the last record has not been written.

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

Resource exit

The line-data transform provides an exit that enables you to filter (exclude) resources. This exit is useful in controlling resources at the file-name level. For

example, assume you only wanted to use those fonts that are not shipped with Infoprint Manager. You could code this exit program to contain a table of all fonts shipped with Infoprint and filter those from the resource file. Security is another consideration for using this exit because you could prevent certain named resources from being included. The program invoked at this exit is defined by the `resexit` keyword of the `line2afp` command.

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

The following example contains a sample C language header that describes the control block passed to the exit program:

```
typedef struct_RESEXIT_PARMS /* Resource record exit Parameters */
{
char *work; /*Address of 16-byte static work area */
PFATTR *pfattr; /*Address of print file attribute information */
char *resname[8]; /*Name of the requested resource */
char restype; /*Type of requested resource */
char request; /*Ignore or process the request */
char *eof; /*Last call indicator */
}
RESEXIT_PARMS
```

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

work (Bytes 1–4)

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

pfattr (Bytes 5–8)

A pointer to the print file attribute data structure. Refer to “Attributes of the line data input file” on page 124 for more information on the format of this data structure and the information presented.

resname (Bytes 9–16)

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

restype (Byte 17)

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

- X'03'** Specifies a GOCA (graphics) object.
- X'05'** Specifies a BCOCA (barcode) object.
- X'06'** Specifies an IOCA (IO image) object.
- X'40'** Specifies a font character set.
- X'41'** Specifies a code page.
- X'FB'** Specifies a page segment.
- X'FC'** Specifies an overlay.

The line-data transform does not call this exit for the following resource types:

Page definition

The page definition (**pagedef** keyword) is a required resource for transforming a line-data file.

Form definition

The form definition (**formdef** keyword) is a required resource for printing a transformed line-data file.

Coded fonts

The line-data transform must process coded fonts to determine the names of the code pages and font character sets they reference. This is necessary in creating Map Coded Font-2 (MCF-2) structured fields.

request (Byte 18)

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

X'00' Specifies that the line-data transform should process the resource.

X'01' Specifies that the line-data transform should ignore the resource.

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

eof (Byte 19)

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

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

Y Specifies that the last record has been written.

N Specifies that the last record has not been written.

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

Non-zero return codes

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

Attributes of the line data input file

Note: This data structure is provided for informational purposes only.

The line-data transform provides information about the attributes of the line-data input file in a data structure available to the line-data transform user exits. The following example shows the format of this data structure.

```

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

```

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

cc (Bytes 1–3)

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

cctype (Byte 4)

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

chars (Bytes 5–24)

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

formdef (Bytes 25–32)

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

pagedef (Bytes 33–40)

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

prmode (Bytes 41–48)

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

trc (Bytes 49–51)

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

Using the **iconv** command to convert code pages

From a Command Prompt window, you can invoke the **iconv** command to convert a code page from one format to another (for example, to convert ASCII to EBCDIC). This command is useful when the line data code pages must be changed so they match the code pages for the available font resources for a job. These are often applied for job resources (such as page definitions and form definitions) that have been created on either an AIX or an MVS operating system and sent to Windows NT or Windows 2000.

Use the following syntax to invoke this command:

```
iconv -f nnn -t nnn infile [> outfile]
```

-f The existing code page that needs to be changed, using the format of *nnn* where *nnn* represents the particular code page. The space between the flag and its value is optional.

-t The code page that is the result of the command, using the format of *nnn* where *nnn* represents a three-to-five number alphanumeric code page name in the format *-nnn*, *nnnn*, or *nnnnn*. The space between the flag and its value is optional.

infile The input text file name to which you want to convert. If you do not specify a file name, the **iconv** command expects input from standard input (STDIN).

outfile

The output file where the result of the **iconv** transform is stored. Use an output file to pass the **iconv** output to a customized program designed to process this output. If you do not specify a file name, the output goes to standard out (STDOUT).

Example 1

To convert an ASCII code page to an EBCDIC code page, open a Command Prompt window and type the following:

```
iconv -f 850 -t 500 input.txt > output.txt
```

Example 2

To convert a Japanese ASCII code page to a Japanese EBCDIC code page, open a Command Prompt window and type the following:

```
iconv -f 932 -t 939 input_jpn.txt > output_jpn.txt
```

Chapter 17. Creating and managing auxiliary sheets and PSF DSS user-exit programs

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

This section contains the following topics:

- “Preparing to work with auxiliary sheets”
- “Creating and configuring auxiliary sheet objects” on page 134

Preparing to work with auxiliary sheets

The following topics describe auxiliary-sheet terms, default auxiliary sheets, and PSF DSS user-exit programs supplied by Infoprint Manager, and provides examples of printed auxiliary sheets:

- “Terms relating to auxiliary sheets”
- “Infoprint default auxiliary-sheet objects” on page 128
- “Sequence of user-exit programs” on page 132
- “Directory locations of user-exit programs supplied with Infoprint Manager” on page 133

Terms relating to auxiliary sheets

To understand auxiliary sheets and how Infoprint Manager generates the information that appears on them, you need to understand the following terms:

Auxiliary sheet

Represents a particular sheet of paper, blank or otherwise, that can precede a job, separate copies of a job, or follow a job. PSF printers print auxiliary sheets. Both Internet Print Protocol (IPP) and Passthrough printers also print auxiliary sheets, but they do not use auxiliary sheet objects to activate this support.

Auxiliary-sheet object

Represents a single user-exit program that you can specify for Infoprint PSF printers. User-exit programs process information that can be printed on an auxiliary sheet or recorded in an accounting or audit log. Both the actual destination (PSF printer) and the auxiliary-sheet objects must reside in the same Infoprint server.

Some auxiliary sheet objects supplied with Infoprint Manager generate a file that you can use with an executable reporting utility to view job statistics. Refer to Chapter 18, “Gathering accounting and auditing data about print jobs” on page 137 for more information on viewing statistics.

Auxiliary-sheet object name

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

User exit

A point during print processing when you can tell Infoprint Manager to run a user-exit program, then return control of processing to Infoprint

Manager after the user-exit program ends. Infoprint Manager provides data at each exit point that can serve as input to the user-exit program. Infoprint Manager can use the output from the user-exit program for Infoprint Manager processing; for example, a user-exit program can pass instructions about how to format a start sheet to Infoprint Manager. Infoprint Manager provides user exits for the following types of auxiliary sheets:

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

User-exit program

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

Form definition

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

For more information about both form definitions, refer to the *Infoprint Manager: Reference*.

Edge marks

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

Infoprint default auxiliary-sheet objects

Infoprint Manager provides six default auxiliary-sheet objects, which it creates when you first create a server. Infoprint Manager automatically identifies these as supported values for PSF actual destinations. The default auxiliary-sheet objects with their associated user-exit programs and form definitions are shown in Table 11.

Table 11. Default auxiliary sheet objects

Auxiliary sheet object	Form definition	Description	User exit program	Additional Information
none	n/a	Infoprint generates no auxiliary sheet.	n/a	n/a

Table 11. Default auxiliary sheet objects (continued)

Auxiliary sheet object	Form definition	Description	User exit program	Additional Information
brief	F1A10110	Start sheet	ainuxhdr	
		Separator sheet	ainuxsep	
		End sheet	ainuxtlr	
		Accounting exit	ainuxacc	
		Audit exit	ainuxaud	
full	F1A10110	Start sheet	ainuxhdr2	
		Separator sheet	ainuxsep2	
		End sheet	ainuxtlr2	
		Accounting exit	ainuxacc2	
		Audit exit	ainuxaud2	
blank	F1A10110	Start sheet	pduxblkh	Blank sheet of paper
		Separator sheet	pduxblks	
		End sheet	pduxblkt	
accounting-log	n/a	Accounting exit	ainacclog	Writes accounting information to the <i>install_path \var\psf\accounting.log</i> file. Does not produce a printed sheet.
		Audit exit	ainaudlog	Writes audit information to the <i>install_path \var\psf\audit.log</i> file. Does not produce a printed sheet.
job-ticket	n/a	Start sheet	ainuxhdrp	
		Separator sheet	ainuxsepp	
		End sheet	ainuxtlrp	
		Accounting exit	ainuxaccp	Writes accounting information to the <i>install_path \var\psf\podaccount.log</i> Does not produce a printed sheet.
		Audit exit	ainuxaudp	

You can create your own auxiliary-sheet objects using the Infoprint **pdcreate** command from a Command Prompt window. Refer to “Creating new auxiliary-sheet objects using the pdcreate command” on page 134.

Examples of Brief-Style and Full-Style Auxiliary Sheets

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

Start Sheet Examples: In Figure 49, the user-exit program **ainuxhdr** generated the brief style of the start sheet and user-exit program **ainuxhdr2** generated the full style of the start sheet.

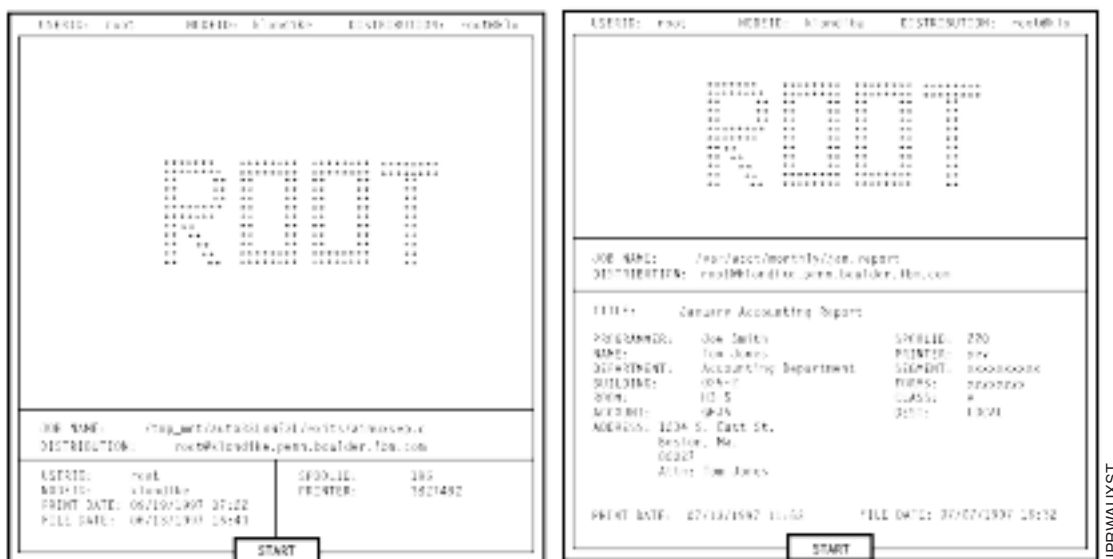


Figure 49. Brief and Full Styles of Start Sheets

Separator Sheet Examples: In Figure 50, the user-exit program **ainuxsep** generated the brief style of the separator sheet and user-exit program **ainuxsep2** generated the full style of the separator sheet.

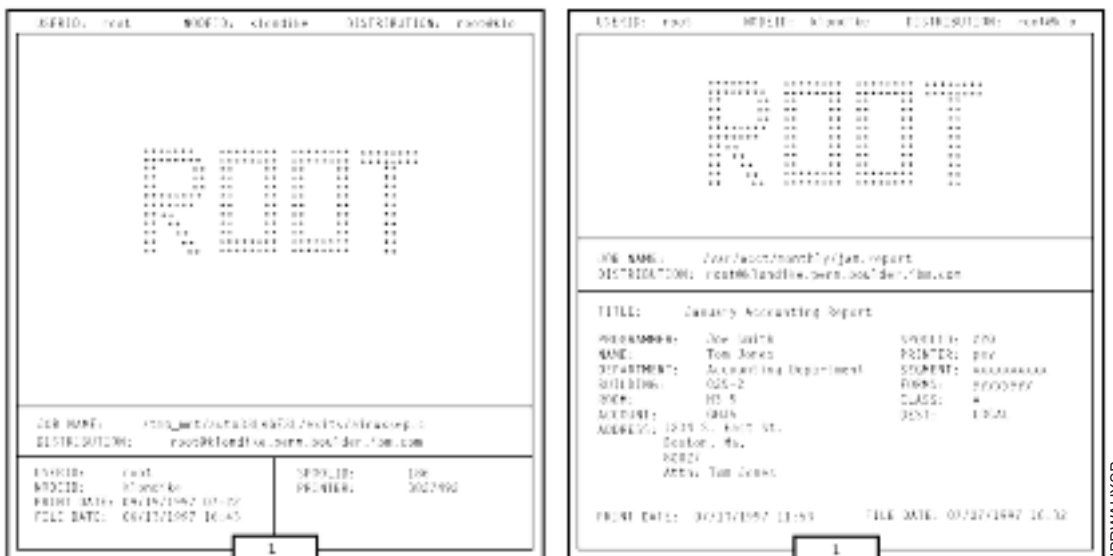


Figure 50. Brief and Full Styles of Separator Sheets

End Sheet Examples: In Figure 51 on page 131, the user-exit program **ainuxtr** generated the brief style of the end sheet and user-exit program **ainuxtr2** generated the full style of the end sheet.

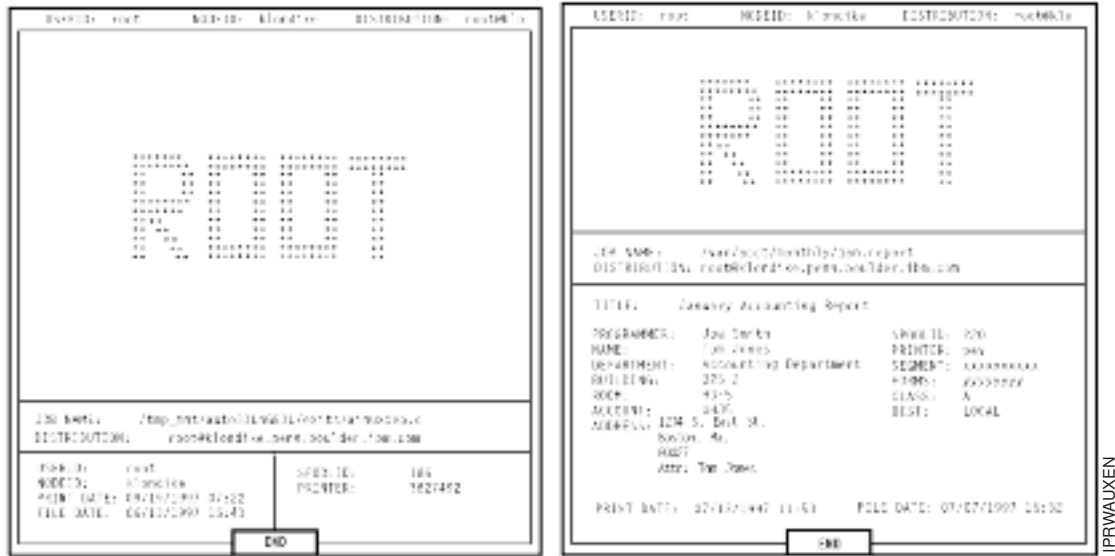


Figure 51. Brief and Full Styles of End Sheets

Accounting Sheet Examples: In Figure 52, the user-exit program **ainuxacc** generated the brief style of the accounting-log sheet and user-exit program **ainuxacc2** generated the full style of the accounting-log sheet.

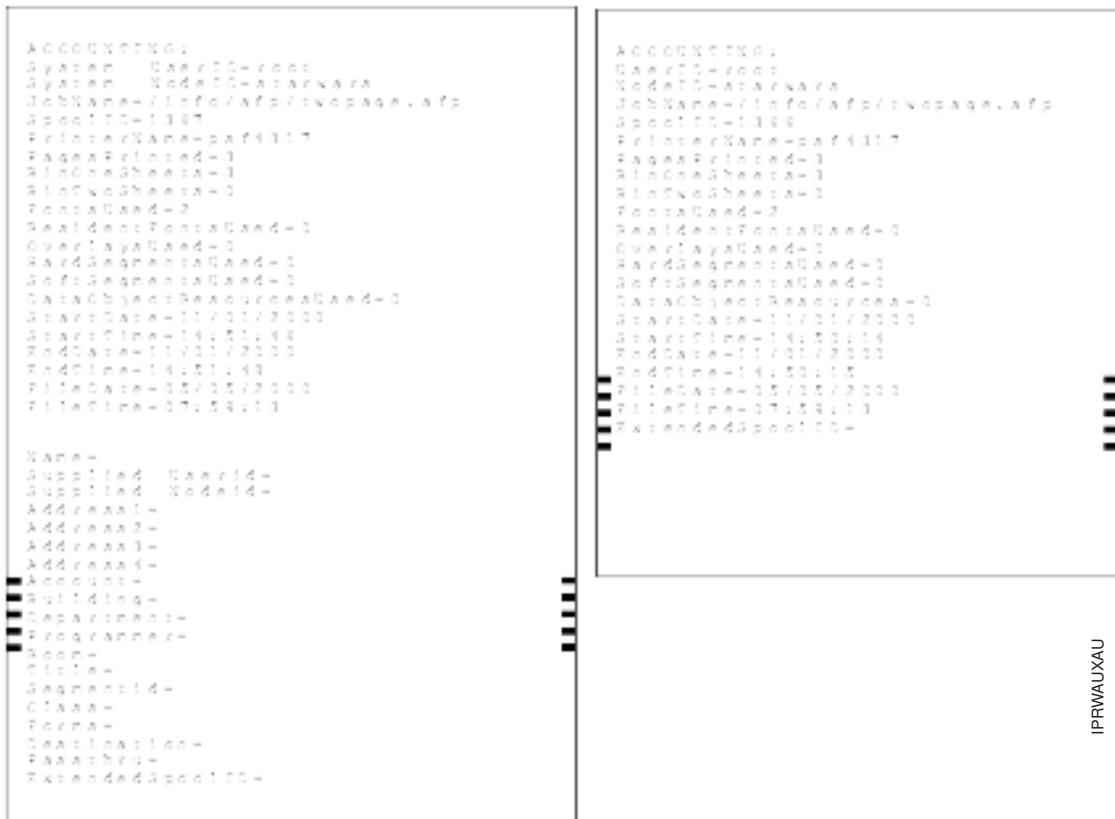


Figure 52. Brief and Full Styles of Accounting-Log Sheets

Audit Sheet Examples: In Figure 53, the user-exit program **ainuxaud** generated the brief style of the audit sheet and the user-exit program **ainuxaud2** generated the full style of the audit sheet.

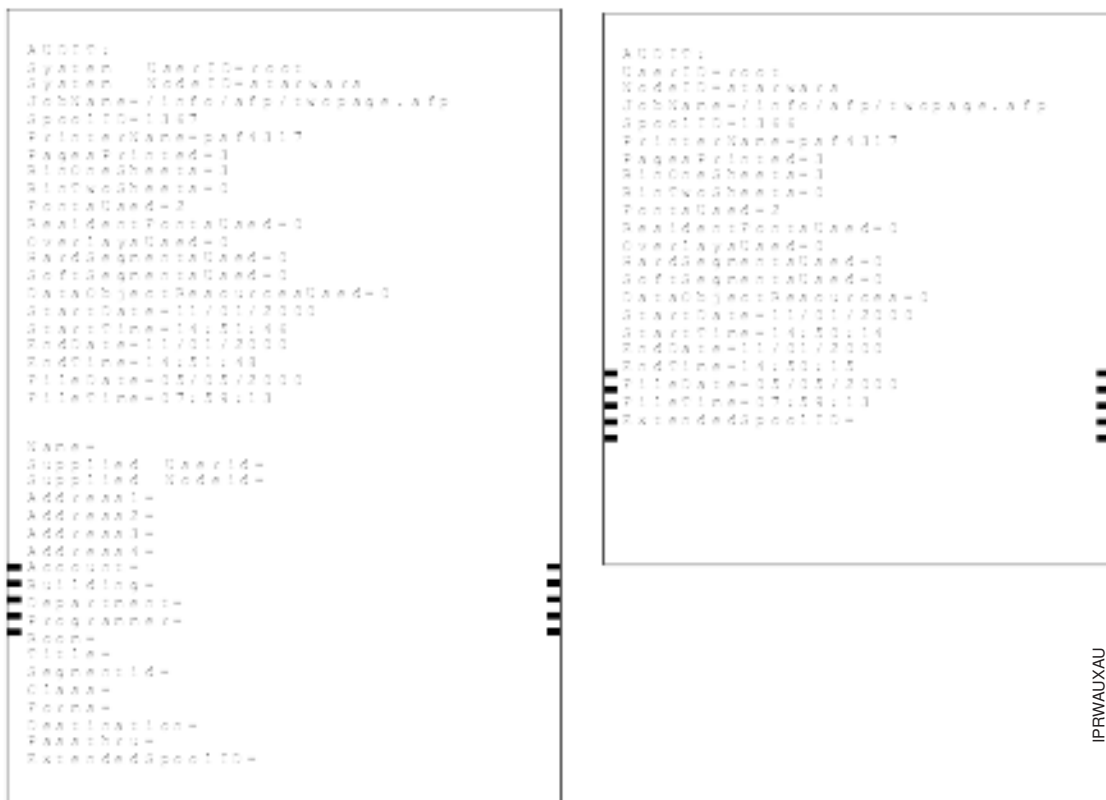


Figure 53. Brief and Full Styles of Audit Sheets

Sequence of user-exit programs

The following figure shows the order in which Infoprint calls the user-exit programs during job processing on a printer. You cannot change this order.

Note: Separator sheets are printed only between copies of a job, not between documents within a job.

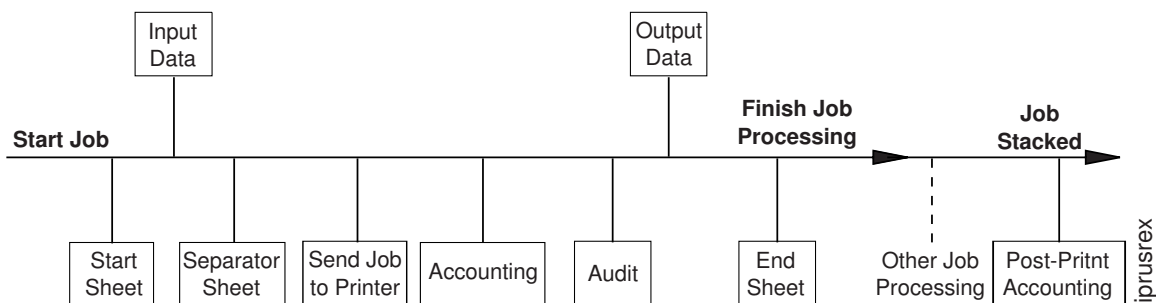


Figure 54. Infoprint Manager user-exit program order at the printer

Directory locations of user-exit programs supplied with Infoprint Manager

When you create the auxiliary-sheet object, you specify the name of the user-exit program and other options, such as the form definition and whether the program produces printed output. You identify which auxiliary-sheet objects you want associated with each actual destination using the GUI attributes notebook for each actual destination. Infoprint Manager runs the referenced user-exit programs when you submit a job to that printer.

Table 12. Locations of Infoprint Manager user exits

Program	Description
<i>Start Sheets</i>	
<code>install_path\bin\pduxblkh.dll</code>	Prints a blank start sheet.
<code>install_path\bin\ainuxhdr.dll</code>	Prints a brief-style start sheet.
<code>install_path\bin\ainuxhdr2.dll</code>	Prints a full-style start sheet.
<code>install_path\bin\ainuxhdrp.dll</code>	Prints job ticket information on the start sheet.
<i>Separator Sheets</i>	
<code>install_path\bin\pduxblks.dll</code>	Prints a blank separator sheet.
<code>install_path\bin\ainuxsep.dll</code>	Prints a brief-style separator sheet.
<code>install_path\bin\ainuxsep2.dll</code>	Prints a full-style separator sheet.
<code>install_path\bin\ainuxsepp.dll</code>	Prints job ticket information on the separator sheet.
<i>End Sheets</i>	
<code>install_path\bin\pduxblks.dll</code>	Prints a blank end sheet.
<code>install_path\bin\ainuxtlr.dll</code>	Prints a brief-style end sheet.
<code>install_path\bin\ainuxtlr2.dll</code>	Prints a full-style end sheet.
<code>install_path\bin\ainuxtlrp.dll</code>	Prints job ticket information on the end sheet.
<i>Accounting Sheets</i>	
<code>install_path\bin\ainuxacc.dll</code>	Prints a brief-style accounting sheet.
<code>install_path\bin\ainuxacc2.dll</code>	Prints a full-style accounting sheet.
<code>install_path\bin\ainuxaccp.dll</code>	Writes accounting information to <code>install_path\var\psf\podaccount.log</code> .
<code>install_path\bin\ainuacclog.dll</code>	Writes accounting information to <code>install_path\var\psf\accounting.log</code> .
<i>Post-print Accounting</i>	
<code>install_path\bin\ainuxjobcompletion.dll</code>	Writes accounting information to <code>install_path\var\psf\jobcompletion.log</code>
<i>Auditing</i>	
<code>install_path\bin\ainuxaud.dll</code>	Prints a brief-style audit sheet.
<code>install_path\bin\ainuxaud2.dll</code>	Prints a full-style audit sheet.
<code>install_path\bin\ainuxaudp.dll</code>	Prints job ticket information on the audit sheet.
<code>install_path\bin\ainaudlog.dll</code>	Writes audit information to <code>install_path\var\psf\audit.log</code> .

Table 12. Locations of Infoprint Manager user exits (continued)

Program	Description
<i>Input Data</i>	
<i>install_path\bin\ainuxind.dll</i>	Performs no tasks; use it as a model for writing your own user-exit program.
<i>Output Data</i>	
<i>install_path\bin\ainuxout.dll</i>	Performs no tasks; use it as a model for writing your own user-exit program.

In addition, you can use the source code of the sample user-exit programs to write your own user-exit programs. Infoprint Manager provides the source code for all of these exits in the *install_path\exits\psf* directory.

Creating and configuring auxiliary sheet objects

When you first create a server, Infoprint creates the default auxiliary-sheet objects described in “Infoprint default auxiliary-sheet objects” on page 128.

The following topics describe how you can create and configure auxiliary-sheet objects:

- “Creating new auxiliary-sheet objects using the `pdcreate` command”
- “Associating auxiliary-sheet objects with PSF printers” on page 135
- “Activating auxiliary sheets for accounting or auditing information” on page 135
- “Associating the post-print accounting user-exit program with an actual destination” on page 135
- “Associating an input data user-exit program with an actual destination” on page 135
- “Associating an output data user-exit program with an actual destination” on page 135

Creating new auxiliary-sheet objects using the `pdcreate` command

You create auxiliary-sheet objects from the **Command Prompt** window; you cannot create them through the Infoprint administrator’s GUI. Use the `pdcreate` command with the `-c auxiliary-sheet` flag and specify the server name and an auxiliary-sheet object name to create a new auxiliary-sheet object. Specify a name that uniquely identifies the auxiliary-sheet object within the server. Different servers can contain a auxiliary-sheet objects with the same name, but possibly with different values for their attributes.

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

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

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


Associating auxiliary-sheet objects with PSF printers

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

Activating auxiliary sheets for accounting or auditing information

Use the **Infoprint Manager Administration GUI** to specify auxiliary sheets to collect accounting and auditing information about jobs processed by a selected PSF printer. Start the GUI and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer —> Properties** to open the **Printer Properties** notebook. To complete this task, refer to the online help in the **Infoprint Manager Administration GUI**, starting with the topic *Using auxiliary-sheet objects*.

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

Use the **Infoprint Manager Administration GUI** to specify auxiliary sheets to collect post-print accounting information about jobs processed by a selected PSF printer. Start the GUI and select the printer that you want to associate with an auxiliary sheet. Then, click **Printer —> Properties** to open the **Printer Properties** notebook. To complete this task, refer to the online help in the **Infoprint Manager Administration GUI**, starting with the topic *Using auxiliary-sheet objects*. Then set the attribute of the auxiliary sheet object using the command shown in “Activating the user-exit program” on page 108.

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

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

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

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

Chapter 18. Gathering accounting and auditing data about print jobs

Infoprint Manager for Windows allows you to gather accounting information about print jobs from two sources:

- The Infoprint Manager server (applies to all destination support systems (DSSs)). For more information, see *Working with Infoprint Manager Accounting Information* below.

You should use accounting information from the Infoprint Manager if you:

- need to track accounting information for all types of Infoprint DSSs.
- want to take advantage of the **pdaccount** command to summarize data in a spread sheet program.

- The PSF auditing, accounting, and post-print accounting user-exit programs (applies only to the PSF DSSs). For more information, see *Working with PSF Accounting and Auditing User-Exit Programs*.

You should use information from these PSF auditing and accounting user-exit programs if you:

- use only PSF attached printers.
- need more detailed information than that provided by the **pdaccount** command.

Working with Infoprint Manager server accounting information

While all DSSs can store accounting information about the jobs processed, the information from some DSSs is more accurate than the information from others. All actual destination accounting logs are enabled through the server and actual destination **log-accounting-data** attribute. When you specify **log-accounting-data=true**, Infoprint stores data into server accounting logs (one for each destination), which you can either view directly through your editor of choice (such as Notepad) from the command-line or by specifying the **pdaccount** command. The **pdaccount** command allows you to obtain summary information in a comma-delimited format (Printer1,12997,1989787763). This information can be saved in a file, which can be imported into a spread sheet to calculate the cost of consumables, the usage rate of printers, or the cost of printing by individual user. The summary information includes both the total pages and total octets (the number of bytes) printed, either by destination or by user as requested for a specified time period. You can also obtain the complete accounting records (information for all print jobs) for a specified time period, instead of the summary data.

How does the **pdaccount** command collect Infoprint Manager accounting information?

To activate Infoprint Manager account log support, you must change the **log-accounting-data** value. By default, the Infoprint Manager value is **log-accounting-data=false**, and the actual destination default is to use the server setting. As a result, you must **log-accounting-data** attribute as follows:

- To activate accounting for all destinations on a server:
set the server's **log-accounting-data** attribute to **True** (if an actual destination hasn't specified this attribute, the server's **log-accounting-data** attribute is used).

- To activate accounting for a specific actual destination:
set the **log-accounting-data** attribute to **True** for that specific actual destination. If you have not set an actual destination, the value of the **log-accounting-data** attribute for the server is used.
- To deactivate accounting for all destinations on a server, :
set the **log-accounting-data** attribute to **False** . If an actual destination hasn't specified this attribute, the server's **log-accounting-data** attribute is used.
- To deactivate accounting for a specific actual destination:
set the **log-accounting-data** attribute to **False** for that specific actual destination. If an actual destination hasn't specified this attribute, the server's **log-accounting-data** attribute is used.

You may change the **log-accounting-data** attribute with either a **pdset** command or the Infoprint Manager Administration GUI. For more information on the Infoprint Manager Administration GUI, see the Chapter 6: "Using the Infoprint Interfaces" in *Infoprint Manager for Windows: Getting Started*.

The **pdaccount** command obtains the following accounting information from one or more Infoprint Manager servers:

- **actual destination name**
- **job owner**
- **global job-id**
- **submission time**
- **completion time**
- **pages-completed**
- **octets-completed**
- **data stream**
- **job name**

Note: You only get all these accounting options if you specify **pdaccount -t all** on the command line.

If you wanted to request summary information grouped by destinations on an Infoprint Manager server named Server1 from 8 AM 9/20/99 to 8 AM 9/27/99, specify the following from either the Windows **Command prompt** or **MS-DOS prompt** window:

```
pdaccount -t destination -s '08:00:00 09/20/99' -e '08:00:00 09/27/99'  
Server1
```

Infoprint displays the information in a format similar to the following:

```
Destination Name,Pages Completed,Octets Completed  
Printer1,12997,1989787763  
Printer2,2455,17676836  
Printer3,86673,189808083
```

where Printer1 specifies the actual destination, 12997 indicates the pages completed, and 1989787763 indicates the octets completed within the specified start and end times (in this example, one week).

For more information about using the **pdaccount** command, see *Infoprint Manager: Reference*.

How do you manage the Infoprint Manager accounting logs?

Once you specify `log-accounting-data=true` on either an Infoprint Manager server or an actual destination, these accounting logs start growing without bound. As part of your scheduled system maintenance, the accounting logs should be managed carefully to avoid potential performance problems. Periodically, the log files must be either deleted or moved to a different folder. If you do not perform either of these tasks, your server will run out of space on the disk drive, preventing you from printing any more jobs.

The Windows NT and Windows 2000 accounting logs reside in a subfolder of the server's folder. The following describes the subdirectory where the files reside:

```
install_path\var\pd\accounting.logs
```

Infoprint Manager for Windows provides one accounting log for each destination, named *actual_destination_name*.**accounting.log.v1**, where *actual_destination_name* specifies the actual destination where the job is processed.

To save a record of Infoprint Manager server accounting data, you need to move the existing log files to a new drive. Be sure that these logs are not being written to, then use the following procedure:

1. Create a new directory in the location where you want to archive the existing (**prt1.accounting.log.v1**) log file by moving to that location on your system (from the command line prompt, specify E:, then `cd \ntserver`).

To create the new directory, type: `mkdir 3qrt2001logs`

2. Go back to your C drive to the folder where the log file resides. Then access that location, for example **prt1.accounting.log.v1** on server1, by specifying:

```
cd install_path\var\pd\accounting.logs
```

3. Copy the **prt1.accounting.log.v1** file to a new directory on a different drive (E for this example) named **3qrt2001** by specifying:

```
copy prt1.accounting.log.v1 E:\ntserver\3qrt2001logs\
```

This command copies the existing file to the new directory.

4. Remove the original **prt1.accounting.log.v1** file from the *install_path*\var\pd\accounting.logs folder:

```
del prt1.accounting.log.v1
```

When the next job is processed on **prt1**, Infoprint creates a new **prt1.accounting.log.v1** file.

IBM recommends you move the data to another disk drive rather than deleting it from the system. This allows you to store accounting data and delete it from the system when you no longer need to refer to it. If you want to delete the **prt1.accounting.log.v1** file,

1. Specify `cd install_path\var\pd\accounting.logs`

This command moves you into the correct path.

2. Enter

```
del prt1.accounting.log.v1
```

When the MS-DOS command-line prompt returns, you know that the **prt1.accounting.log.v1** file has been deleted from the system.

3. Enter

```
dir
```

This command displays the contents of your directory for you to verify that the file has been deleted. When the next job is submitted for printing to that actual destination, Infoprint creates a new **prt1.accounting.log.v1** file and writes the data into that file.

The **pdaccount** command only finds and processes the information in the log files in the original location. Once you move or rename the files, you cannot use the **pdaccount** command to summarize the data. You may want to use the **pdaccount** command to find and process the information and summarize it, store the summaries in a different location, and then delete the logs.

Attention: Requesting information for **all** accounting records can consume a lot of the server's memory since all the data is read into memory to be sent to you. If you have a lot of data to retrieve, you should consider requesting it for smaller time periods and concatenating the returned data.

For example, if you want to retrieve all the accounting records for a one month period, you could issue four **pdaccount** commands like this:

```
pdaccount -t all -s '00:00:00 03/01/00' -e '00:00:00 03/08/00' Server A
>March1.accting.data
pdaccount -t all -s '00:00:00 03/08/00' -e '00:00:00 03/15/00' Server A
>March2.accting.data
pdaccount -t all -s '00:00:00 03/15/00' -e '00:00:00 03/22/00' Server A
>March3.accting.data
pdaccount -t all -s '00:00:00 03/22/00' -e '00:00:00 03/31/00' Server A
>March4.accting.data
```

Usage considerations by DSS for accounting information

This section contains more information about server accounting information by DSS. It also gives suggestions for working with the various DSSs when using the accounting information in Infoprint Manager.

In general, Infoprint Manager logs jobs that complete or are paused, cancelled, or are cleaned while being processed by the actual destination. Infoprint Manager **does not** log jobs that are not completed but are put back into the queue because the printer is shutdown when a printer problem is encountered.

PSF DSS with TCP/IP attachment type

- On jobs that complete successfully, the page count shown in the accounting log is the number of AFP pages stacked on the printer, including start pages, separator pages, and end pages.

Note: For n-up documents, up to *n* pages are counted for each sheet-side that prints.

- **Limitations:**
 - When printing a multiple-document job in which all the documents are ASCII, the accounting log only shows pages for the last document printed.
 - If a job is paused or cancelled, the page count shown in the accounting log may not be accurate.
 - On paused jobs, the octet count is always zero (0).
 - The number of Intelligent Printer Data Stream (IPDS) pages printed is based upon the number of **begin-page** and **end-page** structured field

pairs in the data. As a result, a **2-up** document printed duplex could provide a page count of four for a single sheet of paper.

PSF DSS with Other-Driver attachment type

- If you are using the IBM TCP/IP Port Monitor and the job completes successfully, then the page count shown in the accounting log is the number pages stacked at the printer. For example, a three-page document printed simplex records three pages in the accounting log and three pages stacked; a four-page document printed duplex reports four pages in the accounting log even through only 2 sheets are stacked. A duplexed four-page job with a header sheet provides a page count of four, because the auxiliary sheets are not included in the count, but both sides of each page are counted.
- If you are not using the IBM TCP/IP Port Monitor, the page count is not based on the number of pages stacked on the printer. Instead, the page count recorded in the accounting log is the number of pages PSF reports it has sent to the printer driver, including start pages, separator pages, and end pages.
- If you have set the **document-formats-ripped-at-destination** actual destination attribute for a PSF DSS with Other-Driver attachment type actual destination, the accounting is handled as if it had been submitted to a Passthrough DSS actual destination.
- **Limitations:** The page count and octet count reported in the accounting log for cancelled and paused jobs is not accurate.

PSF DSS with Command attachment type

The page count recorded in the accounting log for destinations using the command is the number of pages PSF generates and sends to the printer. It is not based on the number of pages received or printed by the printer. The page count does include start pages, separator pages, and end pages.

BSD DSS

The page count recorded in the accounting log for destinations using the BSD DSS is based on the following criteria:

- If the user specifies a **job-page-count** job attribute or a **page-count** document attribute with the print job, Infoprint Manager records that number in the accounting log.
- If the user has not specified **job-page-count** or **page-count**, and the data being printed is PCL or PostScript, Infoprint Manager attempts to calculate the page count and records the calculated page count in the accounting log.

Internet Printing Protocol (IPP) DSS

The page count recorded in the accounting log for destinations using the IPP DSS is based on the following criteria:

- Depending upon the capabilities of the your particular IPP printer, if the printer reports pages completed information through the Internet Printing Protocol and the job completes successfully, that information is used. The auxiliary sheet is not added to the count.
- If the user specifies a **job-page-count** job attribute or a **page-count** document attribute with the print job, Infoprint Manager records that number in the accounting log. The auxiliary sheet is not added to the count.
- If the user has not specified **job-page-count** or **page-count**, and the data being printed is PCL or PostScript, Infoprint Manager retrieves the page

count from the printer, if the particular printer supports that function. The auxiliary sheet is not included in the count.

- **Limitations:** Because all information is obtained through the Internet Printer Protocol, the attributes required for **pages-completed** might not be supported by a particular printer model.
- **Note:** For accurate results with this DSS, ensure that you set the sides default (whether simplex or duplex) in the printer to match the value on the **sides** actual destination attribute.

Passthrough DSS

The page count recorded in the accounting log for destinations using the Windows DSS is based on the following criteria:

- If you are using the IBM TCP/IP Port Monitor and the job completes successfully, then the page count shown in the accounting log is the number pages stacked at the printer. For example, a three-page document printed simplex records three pages in the accounting log and three pages stacked; a four-page document printed duplex reports four pages in the accounting log even through only 2 sheets are stacked. Auxiliary sheets are not included in the count.
- If you are using the IBM TCP/IP Port Monitor and the job does not complete successfully, or if you are not using the IBM TCP/IP Port Monitor and the user specifies a **job-page-count** job attribute or a **page-count** document attribute with the print job, Infoprint Manager records that number in the accounting log.

A duplexed four-page job with a header sheet provides a page count of four, because neither the auxiliary sheets nor the back sides of each page are included in the count.

- If the user has not specified **job-page-count** or **page-count**, and the data being printed is PCL or PostScript, Infoprint Manager attempts to calculate the page count and records the calculated page count in the accounting log. The auxiliary sheet is not included in the count.

Note: For accurate results with this DSS, ensure that you set the sides default (whether simplex or duplex) in the printer to match the value on the **sides** actual destination attribute.

Working with PSF accounting, post-print accounting, and auditing data about the job

The accounting, post-print accounting, and auditing PSF DSS user-exit programs record data about jobs. For each job printed, Infoprint Manager activates the PSF DSS user-exit programs specified by the actual destination's **accounting-exit** attribute values before processing the print job's trailer page. More than one Infoprint actual destination can use the same user-exit program to write into the same log file.

This section contains the following topics:

- "What do the accounting and audit PSF DSS user exits provide?" on page 143
- "What does the post-print accounting user exit provide?" on page 143
- "How do you format accounting, post-print accounting, and audit data for viewing?" on page 144
- "How do you manage the contents of the data files?" on page 145

What do the accounting and audit PSF DSS user exits provide?

The accounting and audit PSF DSS user-exit programs record statistical data on jobs processed by Infoprint Manager actual destinations. The following table shows the most common accounting and audit user-exit programs, the files where they record data, and the types of data that they record.

Program	Log File	Records
<i>install_path</i> \ bin\ainuxaccp	<i>install_path</i> \var\psf\podaccount.log	Job submitter, job ID, actual destination, submission machine (nodeid), pages processed (projected), sheets from bin 1, sheets from bin 2, end time, start date, start time, copies per job
<i>install_path</i> \ bin\ainacclog	<i>install_path</i> \var\psf\accounting.log	Job submitter, job name, actual destination, submission machine (nodeid), pages processed (projected), sheets from bin 1, sheets from bin 2, end time, start date, start time, fonts, resident fonts, overlays
<i>install_path</i> \ bin\ainaudlog	<i>install_path</i> \var\psf\audit.log	Job submitter, job name, actual destination, pages per job, pages per actual destination, jobs per output bin, start date, start time, fonts, resident fonts, overlays

Note: Both *install_path*\ bin\ainacclog and *install_path*\ bin\ainaudlog record the same information in different files.

This information is specified in flat ASCII format, with a single line for each job that is processed. You can use this information to determine billing procedures, to assess the workload of different output devices, or to determine how resources are used at your installation.

What does the post-print accounting user exit provide?

This sample user exit can provide information on the actual number of pages and sheets printed from input tray 1, input tray 2, and a combined total. This user exit also reports how many pages/sheets for the job were generated by Infoprint Manager and how many were user-generated, and it reports these results by input trays (up to 255). The complete list of information reported is shown in the following table.

Program	Log File	Records
<i>install_path</i> \ bin\ainuxjobcompletion	<i>install_path</i> \var\psf\jobcompletion.log	Job ID, job submitter, job name, actual destination, pages per job, copies requested, actual pages stacked per input tray 1/2 and total, processing start and stop date/time, job completion date/time, fonts, resident fonts, overlays

The **ainuxjobcompletion** exit is called when the printer reports back to Infoprint Manager that all of the pages for a job have printed. This includes a partial job (due to a datastream error or because the job was paused/cancelled or

forward-spaced) and also any extra pages produced as a result of a backspace or other recovery. For the last job that prints on a continuous forms printer, the exit is not called until an NPRO (non-print run out) is done, either by an operator or automatically by Infoprint Manager.

Only one record is written for a job that prints, unless a job is paused. In this case, a record is written for the part of the job that was printed up to the pause point; later, if the job is resumed, another record is written for the rest of the job.

Note: If Infoprint Manager does not get positive feedback from the printer that all pages have stacked, no record is written.

How do you format accounting, post-print accounting, and audit data for viewing?

To view this data, you can use one of the following executable reporting utilities that are provided with Infoprint Manager. These reporting utilities only work if you open a **Command Prompt** window, navigate to the *install_path*\bin directory, and run the program from the command line.

- Accounting utilities

ainupod1

Reports accounting data stored in **podaccount.log** in a summary format (one line per user), sorted by customer ID.

ainupod2

Reports accounting data stored in **podaccount.log**, sorted by job ID.

ainupod3

Reports detailed entries of times and pages printed for each job by the customer ID from **podaccount.log**.

ainurpt1

Reports accounting data stored in **accounting.log**, sorted by actual destination.

ainurpt2

Reports summary accounting data (one line per job submitted) stored in **accounting.log**, sorted by job submitter.

ainurpt3

Reports detailed entries of times and pages printed for a specific job submitter from **accounting.log**.

- Audit utilities

ainurpt4

Reports audit data stored in **audit.log**, sorted by actual destination.

ainurpt5

Reports summary audit data stored in **audit.log**, sorted by job submitter.

ainurpt6

Reports detailed entries of times and pages printed for a specific job submitter from **audit.log**.

- Post-print accounting utilities

ainurpt7

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

ainurpt8

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

ainurpt9

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

How do you manage the contents of the data files?

An installation with a regular schedule of printing jobs will accumulate accounting, post-print accounting, and audit data that must be managed carefully to avoid potential performance problems. Periodically, the log files must be either purged or moved to a different folder. If you do not perform either of these tasks, your installation may run out of space on the file system, preventing you from printing any more jobs. To avoid this problem, you should purge or move the accounting and audit logs as part of your system maintenance.

IBM recommends that you move the data to another file system rather than deleting it from the system. That way, you can store accounting data and delete it from the system when you no longer need to refer to it. The following procedure describes one way to copy the data to a new location.

1. On your Infoprint Manager server system, open the Management Console and select **Edit** → **Service Configuration**.
2. Make note of the directory that is listed in the **Install path** field.
3. Open Windows NT Explorer or Windows 2000 Explorer and navigate to the *install path*\bin directory.
4. In that directory, find the log file that you need to move, for example, **jobcompletion.log** and select it.
5. With the file selected, click **Edit** → **Copy**.
6. Navigate to the directory that you want to archive the log file in and select the directory.

Note: This directory may be on a different Windows system than the one you are currently working on.

7. With the directory selected, click **Edit** → **Paste**.
8. Verify that the log file appears in the new directory.
9. Navigate back to the *install path*\bin directory.
10. In that directory, find the log file that you need to move, for example, **jobcompletion.log** and select it.
11. With the file selected, click **File** → **Delete**.

If you want to delete the **jobcompletion.log** file without storing the data in another folder, use the procedure above, skipping steps five through ten.

When the next job is submitted for printing, Infoprint creates a new **jobcompletion.log** file and writes the data into that file.

Chapter 19. Using the Infoprint Manager line printer daemon (LPD)

The Infoprint Manager Line Printer Daemon (LPD) is a utility that allows you to submit print jobs from other operating systems using a Line Printer (LPR) client. The Infoprint Manager LPD replaces the LPD that is supplied with the Windows operating system (TCP/IP Print Services), however, it submits the files it receives directly to an Infoprint destination, not to a Windows printer. It also supports the -o options that are available with some LPR clients (-o options allow you to specify advanced Infoprint print options such as formdefs and pagedefs).

Only one LPD can run on your Infoprint Manager server system at a time. If you have TCP/IP Print Services installed, Infoprint Manager will disable it and enable the Infoprint Manager LPD. If you would prefer to use TCP/IP Print Services, use the Management Console to change the service configuration. Refer to the Management Console help for further instructions.

Important: If you have a different LPD installed, Infoprint Manager will **not** disable it, but will install and start the Infoprint Manager LPD. You must stop one of the LPDs before you try to submit a print job using an LPR client.

To submit print jobs to the Infoprint Manager LPD, you can use any LPR client. However, if you want to be able to specify advanced Infoprint options on the -o flag, you should use the **lprafp** sample code package. **lprafp** is an LPR client that is available as a free download from the IBM Printing Systems website. The package includes executable forms of the **lprafp** client for the Windows and AIX (version 4.3.3 and above) platforms. You can compile this code to use it on other platforms as well. IBM Printing Systems does not provide support for the package.

Refer to "Submitting print jobs through the Infoprint Manager LPD" on page 297 for information about downloading, configuring, and using the **lprafp** sample code package.

