

z/OS Infoprint Server Implementation

Customize Infoprint Server

Determine which components and functions you need

Sample user exits are provided



Paul Rogers
Juha Vainikainen

Redbooks



International Technical Support Organization

z/OS Infoprint Server Implementation

September 2001

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notice” on page ix.

First Edition (September 2001)

This edition applies to z/OS Version 1 Release 1, Program Number 5694-A01; to Infoprint Transforms Version 1 Release 1 Modification Level 1, Program 5697-F51; and to all subsequent releases and modifications.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.

Note to U.S Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Contents	iii
Preface	ix
The team that wrote this redbook.	ix
Special notice.	ix
IBM Trademarks.	x
Comments welcome.	x
Chapter 1. Infoprint Server overview.	1
1.1 Infoprint Server introduction	2
1.2 LPR to LPD protocol	2
1.3 Using the JES spool	4
1.3.1 Printing data from the JES spool	5
1.4 Benefits of the Infoprint Server	7
1.5 Infoprint Server components	8
1.5.1 Printer Inventory	8
1.5.2 Print Interface	9
1.5.3 z/OS UNIX print commands	11
1.5.4 Infoprint Server Transforms	12
1.5.5 NetSpool	13
1.6 IP PrintWay	14
1.7 Simple Network Management Protocol (SNMP)	15
1.8 Windows workstation support	16
1.8.1 Printing from Windows to z/OS	17
1.9 Internet Printing Protocol (IPP)	17
Chapter 2. Infoprint Server customization	21
2.1 Printer Inventory Manager.	22
2.1.1 Printer Inventory	23
2.1.2 ISPF application customization	23
2.2 Infoprint Server security	24
2.2.1 Define Infoprint Server administrators and operators	24
2.2.2 Define resource profile AOPADMIN in the FACILITY class	25
2.2.3 Create user environment	26
2.3 Run the aopsetup shell script	27
2.4 Operator control of starting Infoprint Server	27
2.4.1 Define a user ID for the AOPSTART and AOPSTOP JCL procedures	28
2.5 UNIX System Services file structure	31
2.5.1 Customize the HFS directories	32
2.6 Infoprint Server configuration files.	33
2.6.1 Customization of configuration file aopd.conf	33
2.7 Setting Infoprint Server environment variables	36
2.7.1 Customization of UNIX shell print commands	37
2.8 Infoprint Server ISPF application customization	38
2.8.1 Define Infoprint Server Printer Inventory Manager to ISPF	38
2.8.2 The AOPINIT exec	39
2.8.3 Primary ISPF panel for Infoprint Server	40
2.9 Starting the Printer Inventory Manager	41
2.10 TCP/IP customization	42

2.10.1	TCP/IP LPD and Print Interface LPD	42
2.10.2	TCP/IP PROFILE considerations	43
2.11	Starting and stopping Infoprint Server.	44
2.11.1	Infoprint Server start considerations	45
2.11.2	Region size considerations for the IPPD daemon.	46
2.11.3	IPPD daemon considerations	46
2.11.4	AOPJAUTH SAMPLIB member	47
2.11.5	AOPSTART JCL procedure	48
2.11.6	AOPSTOP JCL procedure	48
2.11.7	Starting Infoprint daemons from /etc/rc.	49
2.12	Print Interface customization.	50
2.12.1	JES blank truncation considerations.	50
2.12.2	AOPPRINT JCL procedure customization	51
2.12.3	End-user (lpr) considerations	52
2.12.4	Print Interface LPR/LPD considerations	54
2.13	AOPBATCH utility program.	54
2.13.1	Using AOPBATCH	55
2.14	Starting Infoprint Server	56
Chapter 3. Defining printers in the Printer Inventory.		59
3.1	Printer Inventory	60
3.2	Printer Inventory ISPF application.	61
3.2.1	Printer definitions	61
3.2.2	Other functions	61
3.3	Add Printer definitions.	63
3.3.1	Printer definition considerations	64
3.4	Defining IP PrintWay printers	65
3.4.1	Defining IP PrintWay printers with LPR protocol.	66
3.4.2	IP PrintWay printers with direct socket protocol	75
3.4.3	Defining IP PrintWay printers with IPP protocol	76
3.4.4	IP PrintWay printers with VTAM protocol	77
3.4.5	FSS definitions	78
3.5	Defining a PSF for OS/390 printer	81
3.5.1	Using transforms in the printer definition.	82
3.5.2	Customize PSF for OS/390 for sharing network printers	83
3.6	Defining a general printer	86
3.7	Defining components	87
3.7.1	Creating a component name definition	87
3.8	Display defined printers.	91
3.8.1	Display defined printer using PIDU.	91
Chapter 4. Implementing IP PrintWay		93
4.1	IP PrintWay	94
4.1.1	IP PrintWay TCP/IP considerations	94
4.2	IP PrintWay functional subsystem.	95
4.3	IP PrintWay customization	97
4.3.1	Defining the IP PrintWay functional subsystem in JES2.	97
4.3.2	Defining the IP PrintWay functional subsystem in JES3.	99
4.3.3	IP Printway FSA work-selection criteria	101
4.3.4	APF-authorize SYS1.IMAGELIB.	103
4.3.5	Allocate and initialize the transmission queue data set.	103
4.3.6	Allocating and initializing the IP PrintWay message log data set	107
4.3.7	Create an IP PrintWay startup procedure	109

4.3.8	Setting IP PrintWay LE runtime options	111
4.4	IP PrintWay sample exits	112
4.4.1	Installing exits	112
4.4.2	Activating IP PrintWay exits	113
4.5	IP PrintWay port ranges	113
4.5.1	LPR to LPD considerations	114
4.5.2	IP PrintWay procedure	114
4.6	Printing to IP PrintWay printers with JCL	115
4.6.1	Users using IP PrintWay printers	117
4.6.2	Define the default printer in Printer Inventory	118
Chapter 5.	Implementing NetSpool	123
5.1	NetSpool	124
5.2	NetSpool customization	124
5.2.1	Create NetSpool VTAM APPL statements	125
5.2.2	Define NetSpool printers	126
5.2.3	Allocate and initialize the NetSpool message log	128
5.2.4	Create a NetSpool startup procedure	129
5.2.5	NetSpool Exit Programs	131
5.3	Starting NetSpool	132
Chapter 6.	Infoprint Server client printing	133
6.1	Sending print to Print Interface	134
6.1.1	Print Interface receives data sets	134
6.1.2	Viewing the data set on spool	135
6.2	Infoprint Server client programs	137
6.2.1	Download of client programs	138
6.2.2	Infoprint Port Monitor	138
6.2.3	AFP Printer Driver for Windows	144
6.3	Workstation Users using the lpr command	150
6.3.1	Using the lpr command	150
6.4	Internet Printing Protocol (IPP)	151
6.4.1	IBM IPP client	152
6.4.2	Defining a workstation IPP printer	153
6.5	DFS and SMB support	159
6.5.1	z/OS DFS Server	160
6.5.2	Accessing printers	161
6.5.3	User requests for files and printing	162
6.6	Batch job submission	164
6.6.1	Batch jobs using the AOPPRINT procedure	164
6.6.2	Batch job submission to IP PrintWay	166
6.7	z/OS UNIX users using the lp command	170
6.7.1	z/OS UNIX print commands	170
6.8	TSO/E and LPR protocol	172
6.8.1	TSO/E ISPF command line	173
Chapter 7.	Infoprint Server Transforms	175
7.1	Infoprint Server Transforms	176
7.1.1	Print Interface and transforms	176
7.1.2	Enable the Infoprint Server Transforms	177
7.2	Transform Manager customization	178
7.2.1	Edit the aopd.conf file	178
7.2.2	Create the transform configuration file (aopxfd.conf)	178
7.3	Data stream transforms customization	179

7.3.1 Transform daemon messages	179
7.4 Establish security for the ps2afd daemon	180
7.5 Transform classes	181
7.5.1 Using transform classes	182
7.5.2 Format of a transform entry	184
7.6 Customize the PCL to AFP transform	186
7.6.1 Specify transform classes for the PCL to AFP transform	186
7.7 Customize the PostScript to AFP and PDF to AFP transform	188
7.7.1 Storage requirements	189
7.7.2 Edit the transform configuration file (aopxfd.conf)	189
7.7.3 Format of a PostScript and PDF to AFP transform entry	190
7.7.4 Customize PS2AFP transform resource files	191
7.8 Customize the SAP to AFP Transform	195
7.9 Customize the AFP to PCL, PDF, and PostScript transforms	195
7.9.1 Edit the transform configuration file (aopxfd.conf)	195
7.9.2 Transform classes	200
7.9.3 AFP to PCL transform examples	208
7.9.4 Customizing the font-mapping table	209
Chapter 8. Printer Inventory Definition Utility	213
8.1 pidu command for Printer Inventory management	214
8.2 The pidu command	214
8.2.1 Using the pidu command	215
8.3 PIDU commands	216
8.3.1 PIDU command object classes	217
8.3.2 Using PIDU commands	218
8.4 Running the pidu command as a batch job	223
8.5 Adding a printer definition summary	224
8.6 Backing up and restoring the Printer Inventory	225
8.6.1 PIDU export command	225
Chapter 9. Infoprint Server and SAP R/3	229
9.1 Printing on SAP/R3 systems	230
9.2 Implementing printers in z/OS Infoprint Server	230
9.2.1 Implementing the PSF printer	231
9.2.2 Implementing the IP PrintWay printer	232
9.3 Setting up printing in SAP R/3	234
9.3.1 Setting up Infoprint Server as an OMS	234
9.3.2 Host spool access method L	236
9.3.3 Host spool access method E	238
Appendix A. Dynamic IP PrintWay message exit sample	243
ANFUXMSG - dynamic exit stub	244
UXMSGANF - IP PrintWay ANFUXMSG exit proper	247
Appendix B. Sample IP PrintWay user exits	251
Sample ANFUXBD2 begin data set exit	252
Sample header page from ANFUXBD2	276
Appendix C. Sample ANFUXRC2 user exits	277
Sample user exit	278
Sample ANFUCRC2 record exit	280
Special notices	287

Related publications	289
IBM Redbooks	289
Other resources	289
Referenced Web sites	290
How to get IBM Redbooks	290
IBM Redbooks collections	290
Index	291

Preface

This IBM Redbook describes how to customize the Infoprint Server, an element of z/OS Version 1 Release 1, and Infoprint Server Transforms Version 1 Release 1 Modification Level 1, a separate IBM program product (5697-F51).

It is intended for system programmers and administrators responsible for customizing Infoprint Server for their installation. The reader should be familiar with z/OS UNIX System Services, TCP/IP, the Job Entry Subsystem (JES), and z/OS Job Control Language (JCL).

In this book we introduce Infoprint Server, describe how the components fit into your system, list the functions each component provides, and identify the components you need to customize in order to use each function.

We then describe how to customize each component of Infoprint Server. Sample user exits are provided as an aid in customizing your own exits.

The team that wrote this redbook

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

Paul Rogers is a Consulting IT Specialist at the International Technical Support Organization Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of OS/390 and z/OS. Before joining the ITSO 13 years ago, he worked in the IBM Installation Support Center (ISC) in Greenford, England as OS/390 and JES support for IBM EMEA.



Yuha Vainikainen is a Senior I/T Specialist in IBM Finland. He has extensive knowledge of all aspects of OS/390 and z/OS and all related subsystems. He has worked with Infoprint Server extensively since OS/390 Release 5.

Special notice

This publication is intended to help systems programmers install and customize the Infoprint Server. The information in this publication is not intended as the specification of any programming interfaces that are provided by Infoprint server and Infoprint Server Transforms. See the PUBLICATIONS section of the IBM Programming Announcement for Infoprint Server for more information about what publications are considered to be product documentation.

IBM Trademarks

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

AFP	Hiperspace
AIX	IBM.COM
CICS	OS/2
e (logo)® 	NetSpool
eNetwork	OS/390
FICON	Print Services Facility
IBM ®	PrintWay
IMS	RACF
IMS/ESA	Redbooks
Infoprint	Redbooks Logo 
Intelligent Printer Data Stream	RETAIN
IP PrintWay	S/390
IPDS	SecureWay
Language Environment	VTAM
Advanced Function Printing	z/OS
AS/400	zSeries
ESCON	

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review redbook form found at ibm.com/redbooks
- ▶ Send your comments in an Internet note to redbook@us.ibm.com
- ▶ Mail your comments to address on Page ii



Infoprint Server overview

This chapter provides an overview of the functions and implementation of Infoprint Server. All the functions are discussed in more detail in later chapters.

This chapter contains the following:

- ▶ “Infoprint Server introduction” on page 2
- ▶ “Benefits of the Infoprint Server” on page 7
- ▶ “LPR to LPD protocol” on page 2
- ▶ “Infoprint Server components” on page 8
- ▶ “IP PrintWay” on page 14
- ▶ “Simple Network Management Protocol (SNMP)” on page 15
- ▶ “Windows workstation support” on page 16
- ▶ “Internet Printing Protocol (IPP)” on page 17

1.1 Infoprint Server introduction

Infoprint Server is an optional feature of OS/390 Version 2 Release 8 and higher, and z/OS Version 1 Release 1 and higher. Infoprint Server is a UNIX application that uses OS/390 UNIX System Services in OS/390 systems and z/OS UNIX System Services in z/OS systems. This feature is the basis for a total print serving solution for the OS/390 or z/OS environment in a TCP/IP network. Infoprint Server lets users submit print requests from remote workstations in a TCP/IP network, from UNIX System Services applications, from batch applications, and from VTAM applications, such as CICS or IMS applications. It allows you to consolidate your print workload from the servers onto a central z/OS print server as shown in Figure 1-1.

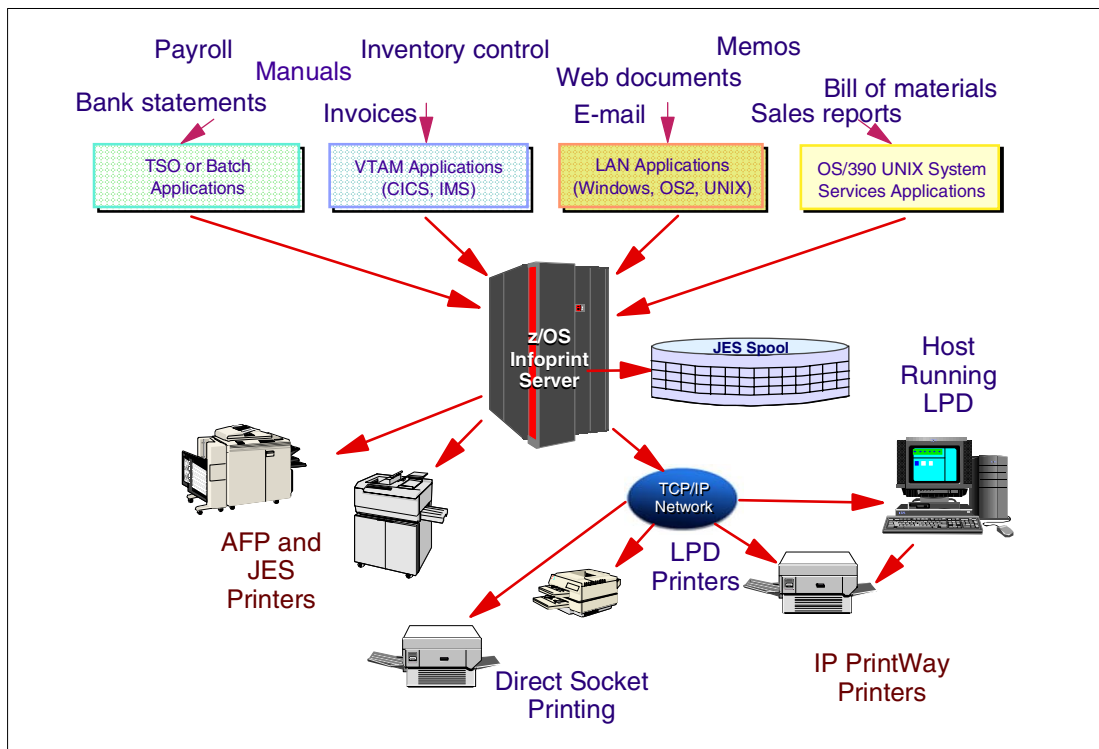


Figure 1-1 Consolidation of application print to JES spool

IP PrintWay is a component of Infoprint Server that transmits output data sets from the JES2 or JES3 spool to remote printers using IP PrintWay, as shown in Figure 1-1 on page 2, or to host systems in your TCP/IP network. A print server can be running on the host system. The remote printer or host system must support either the LPR/LPD protocol, the IPP protocol, or direct socket printing.

1.2 LPR to LPD protocol

One of the protocols that Infoprint Server uses to accept print requests and to send print data is the LPR to LPD protocol.

The Network Printing Working Group has defined the RFC 1179 Request for Comments for the line printer daemon (LPD) protocol. This RFC describes the existing print server protocol, which is widely used on the Internet for communicating between line printer daemons (both clients and servers). The protocol describes in detail how a line printer daemon client requests and controls printing.

LPR is a TCP-based protocol. The port on which a line printer daemon (LPD) listens is 515. The source port must be in the range 721 to 731, inclusive. An LPD responds to commands sent to its port.

Many UNIX operating systems provide line printer spooling with a collection of programs:

lpr	Assign to queue
lpq	Display the queue
lprm	Remove from queue
lpc	Control the queue

These programs interact with an autonomous process called the line printer daemon (LPD). Figure 1-2 illustrates the LPR to LPD protocol for printing.

Note: The lpr and lpq commands are supported on various platforms such as TSO/E, VM, AS/400, OS/2, AIX, and UNIX platforms.

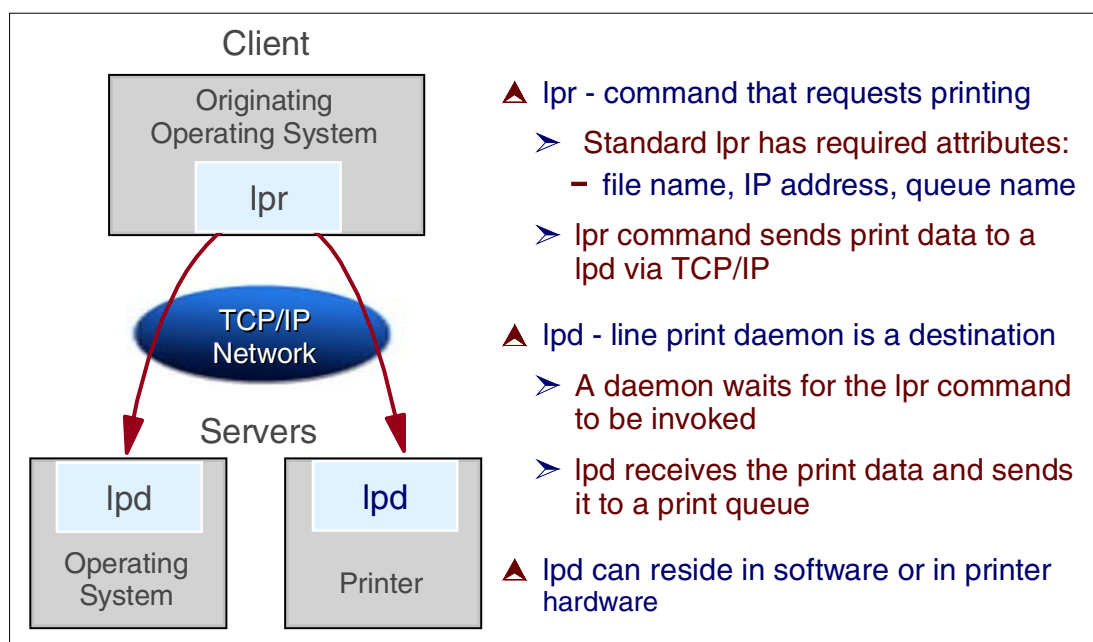


Figure 1-2 LPR to LPD protocol

The Infoprint Server implementation uses this protocol in the following two ways:

1. Workstation client users, using the **lpr** command, send print requests to a server, the Print Interface address space shown in Figure 1-3 on page 4, which acts as a lpd.
2. IP PrintWay, acting as a client, sends print data using the **lpr** command to an lpd, as shown in Figure 1-3 on page 4.

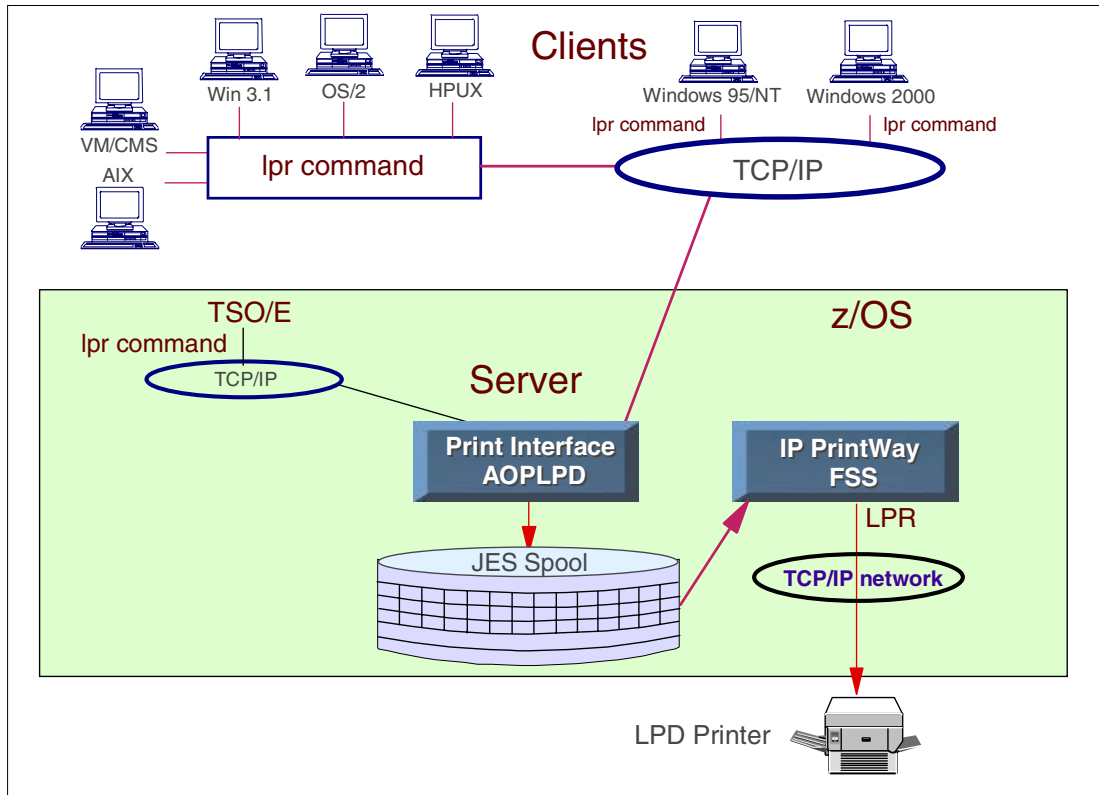


Figure 1-3 LPR to LPD protocol used by Infoprint Server

1.3 Using the JES spool

Infoprint Server uses the JES spool as a temporary application print data repository.

Print data may be received to the JES spool through any of the several print protocols supported by the Infoprint Server. Host applications can make print data available for Infoprint Server processing by writing it directly to the JES spool, as shown in Figure 1-4 on page 5.

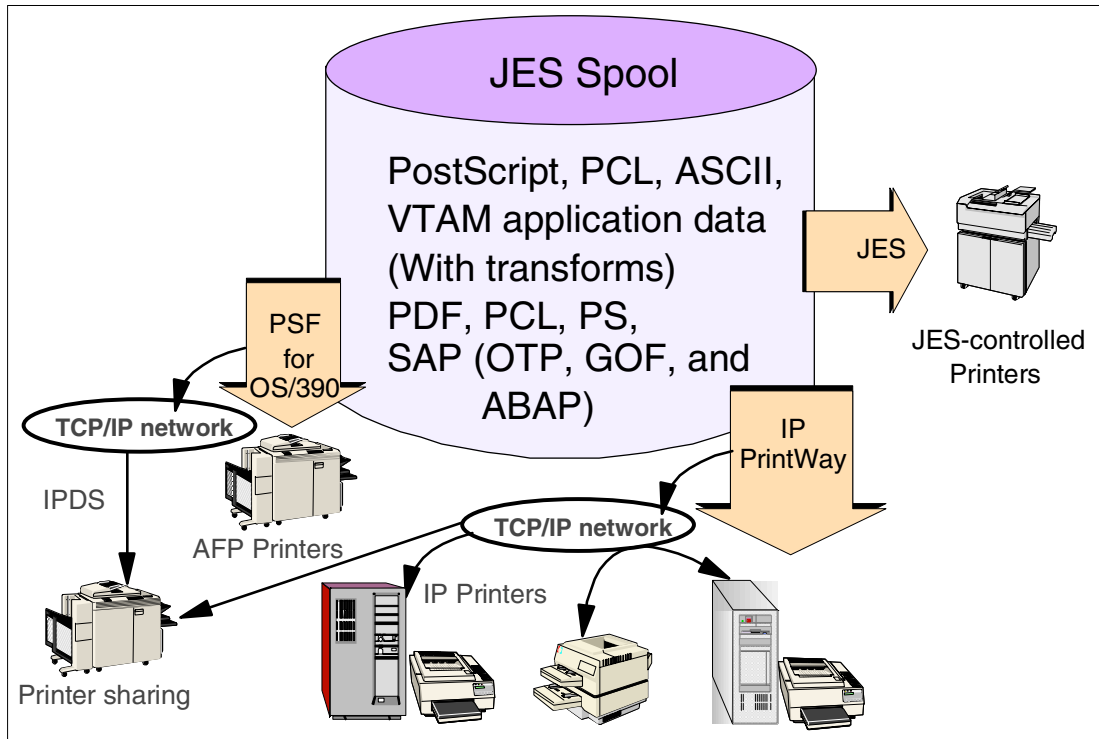


Figure 1-4 Data streams on the JES pool

Infoprint Server enhances the print data formats supported through the JES pool by allowing users to submit the following data streams:

- ▶ PostScript (PS)
- ▶ Printer Control Language (PCL)
- ▶ American National Standard Code for Information Interchange (ASCII)
- ▶ Data from VTAM applications (such as CICS and IMS)
- ▶ Portable Document Format (PDF)
- ▶ SAP Output Text Format (OTF)
- ▶ SAP Advanced Business Application Programming (ABAP)

1.3.1 Printing data from the JES pool

Data stored on the JES pool for printing is processed by the following JES and Infoprint Server components:

- ▶ PSF-controlled AFP printers

These printers operate under the control of Print Services Facility (PSF). They provide all-points-addressable capabilities, and although defined by JES, are driven by PSF. PSF thereby assumes the processing overhead that JES typically performs to support printer operation.

- SNA protocol printers

PSF for OS/390 is an IBM-licensed printer-driver program that manages and controls data transmitted to Advanced Function Presentation (AFP) printers that are channel-attached using SNA.

- TCP/IP protocol printers

PSF views TCP/IP-attached printers in the same way it views SNA-attached printers. The routing of data is determined on the basis of the IP address, and is performed by IP gateways. After the network is configured correctly, the MVS host appears to be communicating directly with the TCP/IP-attached printer.

Note: PSF 3.2.0 now supports AFP printers that can attach to a Fibre Connectivity (FICON) channel. A FICON channel has a higher channel data rate than a parallel or ESCON channel and can be used to quickly send large amounts of data to the printer. This support exploits the high performance capabilities of a FICON channel, takes advantage of the multi-host feature available on FICON channel-attached printers, and lets you define AFP FICON channel-attached printers to your I/O configuration.

- ▶ JES-controlled printers

As the central point of control over the job output function, JES controls output devices: local and remote printers, punches, and card readers. You can use JES initialization statements to define each device.

- ▶ IP PrintWay printers

IP PrintWay, a component of Infoprint Server, transmits output data sets from JES spool to remote printers or to host systems in your TCP/IP network. A print server can be running on the host system. The remote printer or host system must support either the LPR/LPD protocol, the IPP protocol, the VTAM protocol, or direct socket printing.

- LPR/LPD protocol printers

IP PrintWay transmits data to the printer or print server at the IP address (or host name) that you specify in the printer definition. You might want to select the LPR protocol to take advantage of the formatting options that IP PrintWay can specify in the LPD control file, for example, printing a banner page, and printing copies instead of direct socket protocol.

Note: See “Defining IP PrintWay printers with LPR protocol” on page 66

- IPP protocol printers

Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed printing using Internet tools and technologies. IPP is a client-server type protocol that should allow the server side to be either a separate print server or a printer. IP PrintWay can use the Internet Printing Protocol (IPP) to transmit data sets to the target printer. To use the IPP protocol, an IPP server must be running in the remote printer or host system.

Note: See “Defining IP PrintWay printers with IPP protocol” on page 76

- VTAM protocol printers

IP PrintWay also transmits output data sets from JES to VTAM-controlled printers that are defined to VTAM as LU0, LU1, or LU3 printers. The supported output data streams are Data Stream Compatibility/Data Stream Extended (DSC/DSE) and SNA Character String (SCS). This support in IP PrintWay requires that you install the Coax Printer Support feature of Infoprint Server Transforms.

Infoprint Server Transforms for OS/390 provides a new priced feature, the Coax Printer Support feature. This feature lets you use IP PrintWay to print S/390 line-data documents on VTAM-controlled printers defined as VTAM LU type 0 (DSC), LU type 1 (SCS), or LU type 3 (DSC) printers.

Note: See “IP PrintWay printers with VTAM protocol” on page 77

- Direct socket printers

IP PrintWay can use the direct socket printing protocol to transmit data sets to the target printer. The direct socket printing protocol is a TCP/IP protocol in which IP PrintWay transmits data directly to a designated port on the target system. To use this protocol, the remote printer or print server must support direct socket printing.

For large data sets, the direct sockets printing protocol can provide better performance.

Note: See “IP PrintWay printers with direct socket protocol” on page 75

- Printer sharing

As shown in Figure 1-4 on page 5, the printer can be shared because it can print IPDS streams (AFP) and be defined as an IP PrintWay printer and print PostScript and PCL data.

New features and PTFs

Beginning with OS/390 Release 10, several enhancements became available through PTFs, as follows:

- VTAM support in IP PrintWay with PTFs UW73909 and UW73913. Also, part of the support is included in the Infoprint Server Transforms Feature and requires the Coax Printer Support feature with PTF UW74922. See “IP PrintWay printers with VTAM protocol” on page 77.
- New ISPF Protocol panels for IP PrintWay are included in UW73909 and UW73913. See “Protocol options” on page 74 for the protocol panels for LPR, direct socket, IPP, and VTAM.
- IPP enhancements in IP PrintWay to remove the JAVA code in the IPP client and replace it with C/C++ code are included in UW74452 for APAR OW46515.

1.4 Benefits of the Infoprint Server

Infoprint Server delivers improved efficiency and lower overall printing cost with the flexibility for high-volume, high-speed printing from anywhere in the network. With Infoprint Server, you can reduce the overall cost of printing while improving manageability, data retrievability, and usability.

The strength of Infoprint Server is that all applications and users place their print data on the resilient JES spool. This provides a level of security for the data because of the checkpoint and restart and reprint capabilities of JES.

1.5 Infoprint Server components

Figure 1-5 shows the components of Infoprint Server and how they fit into your operating system. They are:

- ▶ Printer Inventory
- ▶ Print Interface
- ▶ z/OS UNIX System Services printing commands
- ▶ Infoprint Server Transforms
- ▶ NetSpool
- ▶ IP PrintWay
- ▶ Simple Network Management Protocol (SNMP) Subagent
- ▶ Windows Workstation support

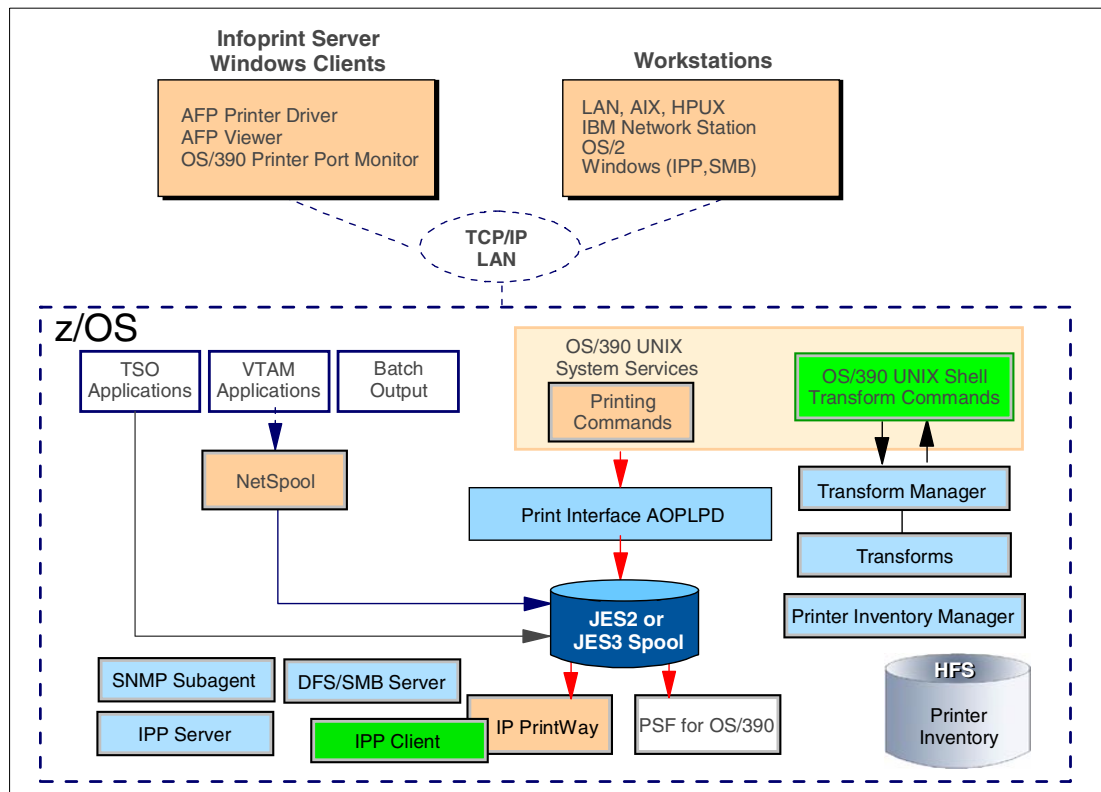


Figure 1-5 Infoprint Server components

1.5.1 Printer Inventory

The Printer Inventory contains information about both local and remote printers and is maintained by the system administrator using an ISPF application or a Print Inventory Definition Utility (PIDU), as shown in Figure 1-6 on page 9. When a user sends data to be printed, the printer definition in the Printer Inventory is used to perform the following functions:

- ▶ Validate the print request to ensure the data stream can be printed on the specified printer
- ▶ Assign processing attributes for the data set to be printed before placing the data set on the JES spool

Printer definitions in the Printer Inventory include objects for Print Interface, IP PrintWay, and NetSpool components of Infoprint Server. This means that the system administrator has to define a printer only once for all components of the Infoprint Server.

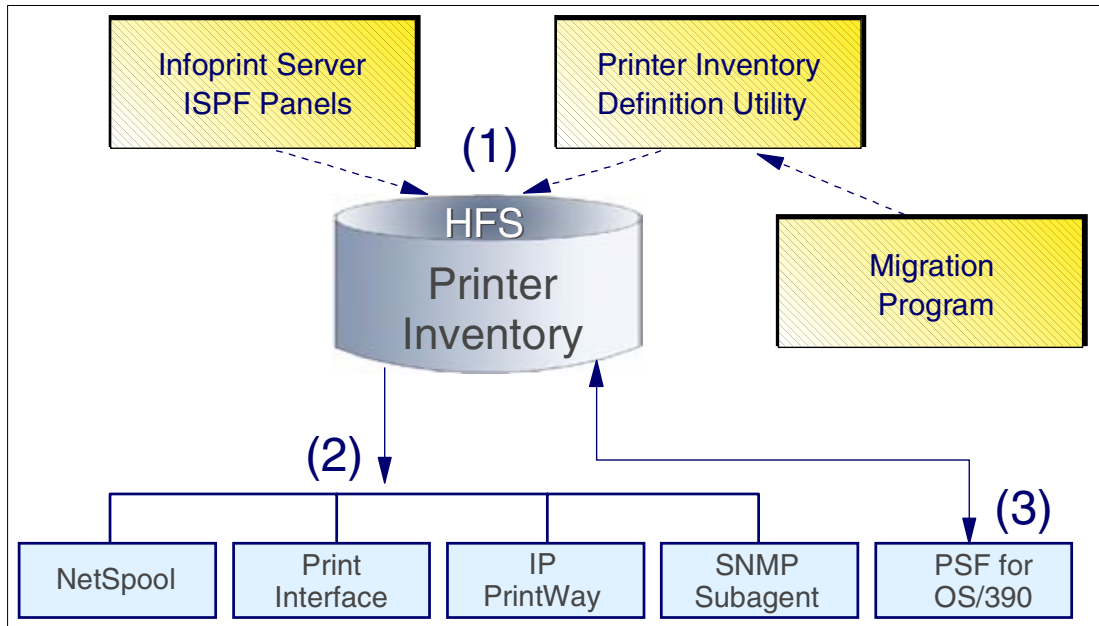


Figure 1-6 Printer Inventory Manager manages the Printer Inventory

1.5.2 Print Interface

Print Interface is the component of the Infoprint Server that accepts input from remote workstations that have TCP/IP access, and from z/OS UNIX System Services printing commands, and creates output data sets on the JES spool.

When you start Infoprint Server, a set of address spaces are created for the Print Interface. These UNIX address spaces perform the following functions:

- ▶ Provide z/OS UNIX System Services printing commands that let you print from the local z/OS system using the **1p** command.
- ▶ Provide an LPD function that lets you submit print requests from remote workstations, using the **1pr** command, that have TCP/IP access.
- ▶ Supports an Internet Printing Protocol (IPP) standard protocol for printing over the Internet. An IPP client must run in the remote system.
- ▶ Receives print requests from remote workstations that use the IPP or the SMB printing protocol, which is standard on Windows systems.
- ▶ Provides an SAP R/3 Output Management System that can receive SAP R/3 print requests.
- ▶ The AOPPRINT Job Control Language (JCL) procedure, provided in SYS1.PROCLIB, lets you submit print data to Print Interface when the batch job executes. The output is placed on the JES spool by Print Interface. This procedure allows you to take advantage of all the features of Infoprint Server and of the optional Infoprint Server Transforms (5697-F51).

Note: See “AOPPRINT JCL procedure customization” on page 51.

Print Interface requests

When a print request is received, Print Interface performs the following functions:

- ▶ It creates an output data set on the JES spool for each print request to be printed.

Print Interface maps the printing options specified on **lp** commands from z/OS UNIX users and some of the printing options specified on **lpr** commands from workstation users to JES output parameters, as shown in Figure 1-7. These parameters are the same that users can specify on JCL statements, on TSO OUTDES and ALLOCATE commands, and on the OUTADD and DYNALLOC macro instructions.

- It responds to query requests with the status of the Infoprint-managed output data set on the JES spool or a list of the printers defined in the Printer Inventory.
- It responds to cancel requests by removing data sets that are on the JES spool. The cancel requests are processed only for data sets that have not yet been selected for printing. For security reasons, Print Interface does not let unauthenticated users on remote systems cancel print requests.

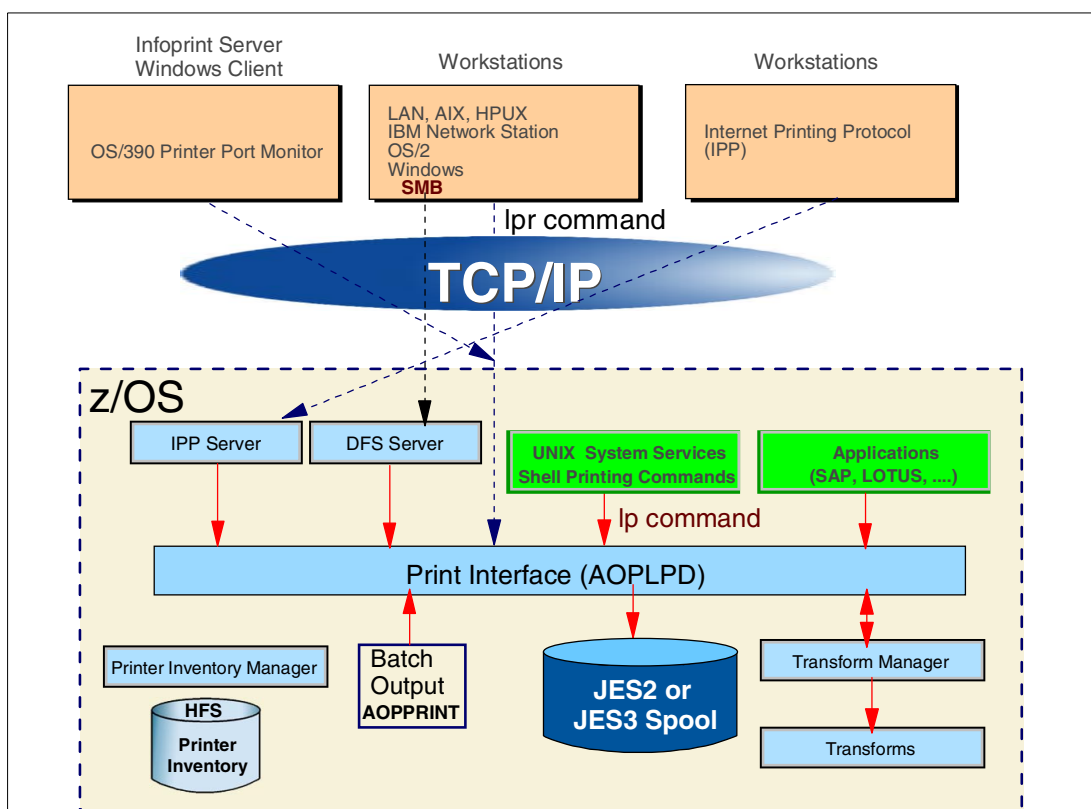


Figure 1-7 Workstations and users passing data sets to the JES spool via Print Interface

SAP R/3 support

Print Interface lets SAP R/3 users print to any printers defined in the Printer Inventory. If the target printer is an IBM AFP printer, Print Interface can automatically invoke the SAP to AFP transform provided by Infoprint Server Transforms to convert SAP OTF and SAP ABAP data to the AFP format, as shown in Figure 1-8 on page 11.

This support allows you to print SAP data directly to IBM AFP printers on z/OS instead of converting SAP data to AFP format on an AIX system (using either PSF for AIX or Infoprint Manager for AIX) and uploading the AFP data to z/OS for printing.

To print from SAP R/3, the SAP R/3 application server can run either on the z/OS system or on another system (such as AIX or Windows NT).

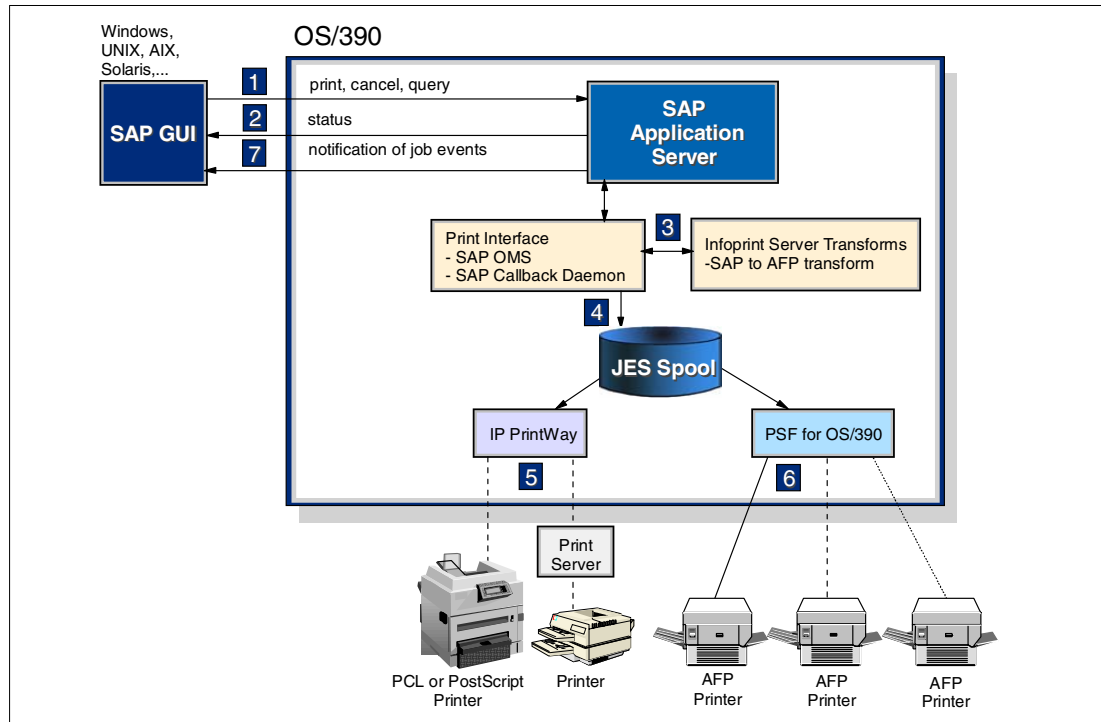


Figure 1-8 SAP R/3 application server running on a z/OS system

1.5.3 z/OS UNIX print commands

A z/OS UNIX Services user can print files using the Print Interface services. Print Interface provides enhanced versions of the z/OS UNIX System Services shell printing commands. These commands have more functions than the standard UNIX shell printing commands.

Note: These enhanced commands adhere to the XPG4.2 standard, which makes it possible to port UNIX applications to z/OS UNIX. See “Customization of UNIX shell print commands” on page 37 for the details.

The UNIX Services shell commands available for processing of files to be printed are:

- lp** The **lp** command sends files for printing to Print Interface running on the local z/OS system. The files that a UNIX user can print are:
- HFS files that include PDF, PCL, SAP, and PS
 - MVS data sets
 - Sequential data sets
 - Partitioned data set members
- lpstat** The **lpstat** command queries the status of print jobs. It also queries the names, locations, and descriptions of printers that the administrator has defined in the Printer Inventory.
- cancel** The **cancel** command cancels print requests, provided that the data set allocated on the JES spool has not yet been selected.

1.5.4 Infoprint Server Transforms

Transforms are programs that convert a data stream from one format to another, for example, from PCL to AFP, PDF to AFP, and so on. The IBM-provided transforms are implemented as DLL filters.

Data stream transforms give you the flexibility to print a variety of output on a wide range of printers. The transforms can run automatically when associated as a filter for a Print Interface supported data format on a printer definition.

Note: See “Infoprint Server Transforms” on page 176.

This product provides filters that transform files in any of the following formats to AFP files for printing on AFP printers:

- ▶ PCL 5e
- ▶ PDF 1.2
- ▶ PostScript Language Level 3
- ▶ SAP ABAP versions 1 and 2
- ▶ SAP OTF versions 1 and 2

Additionally, there are AFP to PCL, AFP to PDF, and AFP to PostScript transforms that are separately priced features of Infoprint Server Transforms. These features can transform files in AFP or S/390 line data format to any of the following formats for printing on PCL or PostScript printers, or posting on the Web:

- ▶ PCL 5, 5e, or 5c (color)
- ▶ PostScript 1.2 (monochrome or color)
- ▶ PDF Language Level 2 (monochrome or color)

z/OS UNIX Services users can run the transforms with the following z/OS UNIX System Services shell commands:

- | | |
|----------------|--|
| pcl2afp | Transforms a PCL file to an AFP file. |
| pdf2afp | Transforms a PDF file to an AFP file. |
| ps2afp | Transforms a PostScript file to an AFP file. |
| sap2afp | Transforms a SAP ABAP or SAP OTF Version 1 or Version 2 file to an AFP file. |
| afp2pdf | Transforms an AFP or line data file to a PDF file. |
| afp2pcl | Transforms an AFP or line data file to a PCL file. |
| afp2ps | Transforms an AFP or line data file to a PostScript file. |

Figure 1-9 on page 13 shows a client data set passed from the workstation to AOPLPD. The Printer Inventory entry for the specified printer causes the data stream to be converted using a specified transform to convert the data to AFP format for printing on an AFP printer.

Note: For customization of the transforms, see “Infoprint Server Transforms” on page 176.

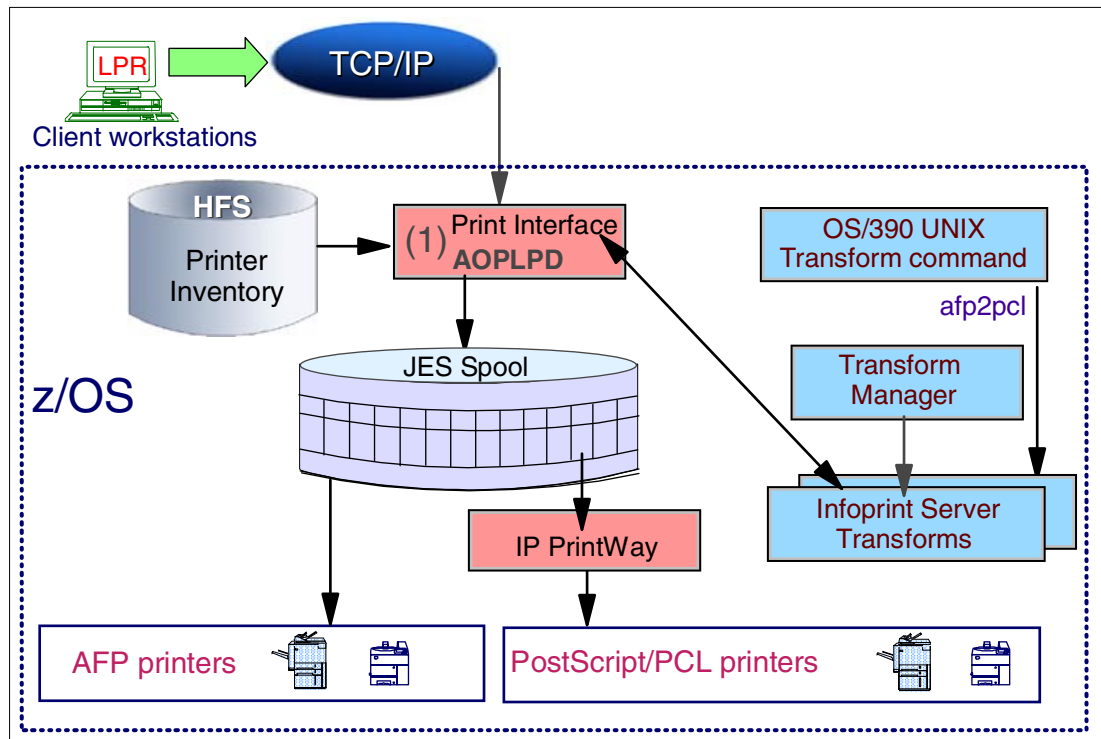


Figure 1-9 A client user data set is transformed before being placed on the spool

1.5.5 NetSpool

NetSpool is the component of Infoprint Server that allows an installation to automatically direct VTAM application data targeted for a network printer to the JES spool without changing the VTAM applications. From the JES spool, the data set can be printed on a JES or PSF for OS/390 printer or sent to another location for printing.

NetSpool intercepts print data from VTAM applications, such as CICS and IMS. It converts the data into S/390 line data and creates output data sets on the JES spool, as shown in Figure 1-10 on page 14. JES, PSF for OS/390, or IP PrintWay can print the output data sets. The data sets can also be transmitted to another location for printing by JES networking support.

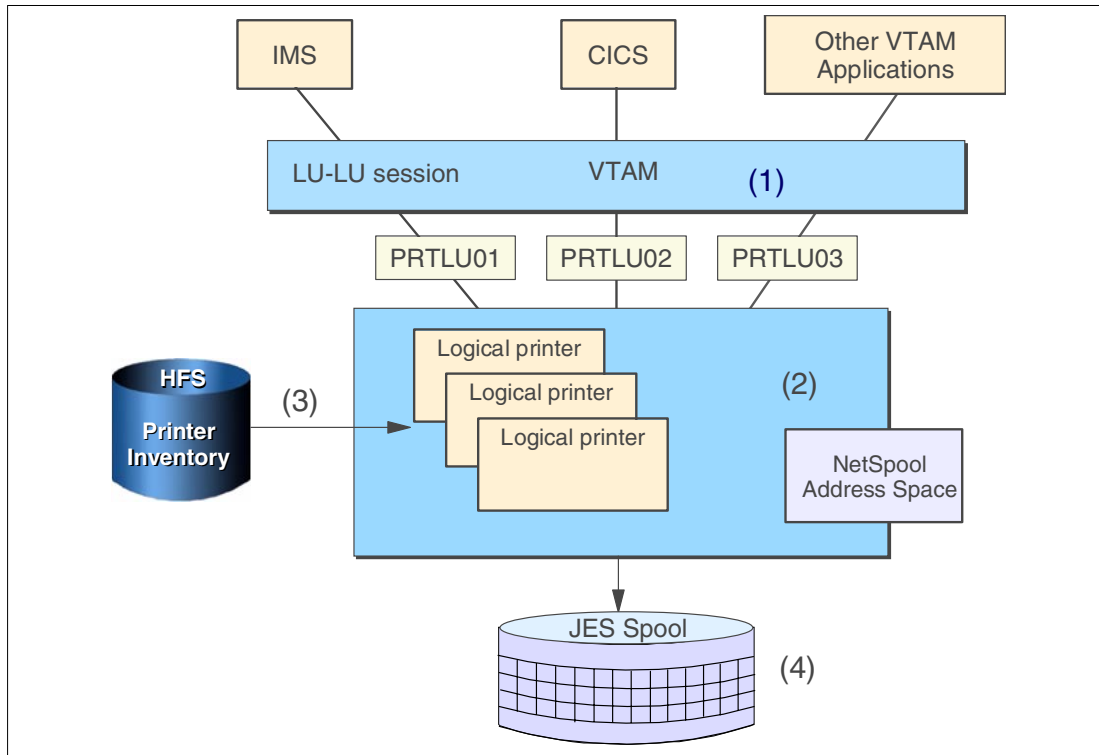


Figure 1-10 VTAM applications passing print data sets to the JES spool via NetSpool

NetSpool data streams

NetSpool supports the following data streams:

- ▶ SNA character string (SCS) data over a logical unit (LU) type 1 session
- ▶ 3270 data over an LU type 0 or type 3 session
- ▶ Binary data over an LU type 0, type 1, or type 3 session

1.6 IP PrintWay

IP PrintWay is the component of Infoprint Server that transmits output data sets from the JES spool to printers in a TCP/IP network.

IP PrintWay selects output data sets from the JES spool and transmits them to remote printers or to host systems in your TCP/IP network. A print server can be running on the host system. The remote printer or host system must support either the LPR/LPD protocol, the IPP protocol, or direct socket printing.

IP PrintWay starts when an operator enters the PrintWay FSA printer start command. You do not need to enter a command to start the FSS because JES starts the FSS automatically when the first FSA under control of that FSS is started. Figure 1-11 on page 15 shows an overview of IP PrintWay processing.

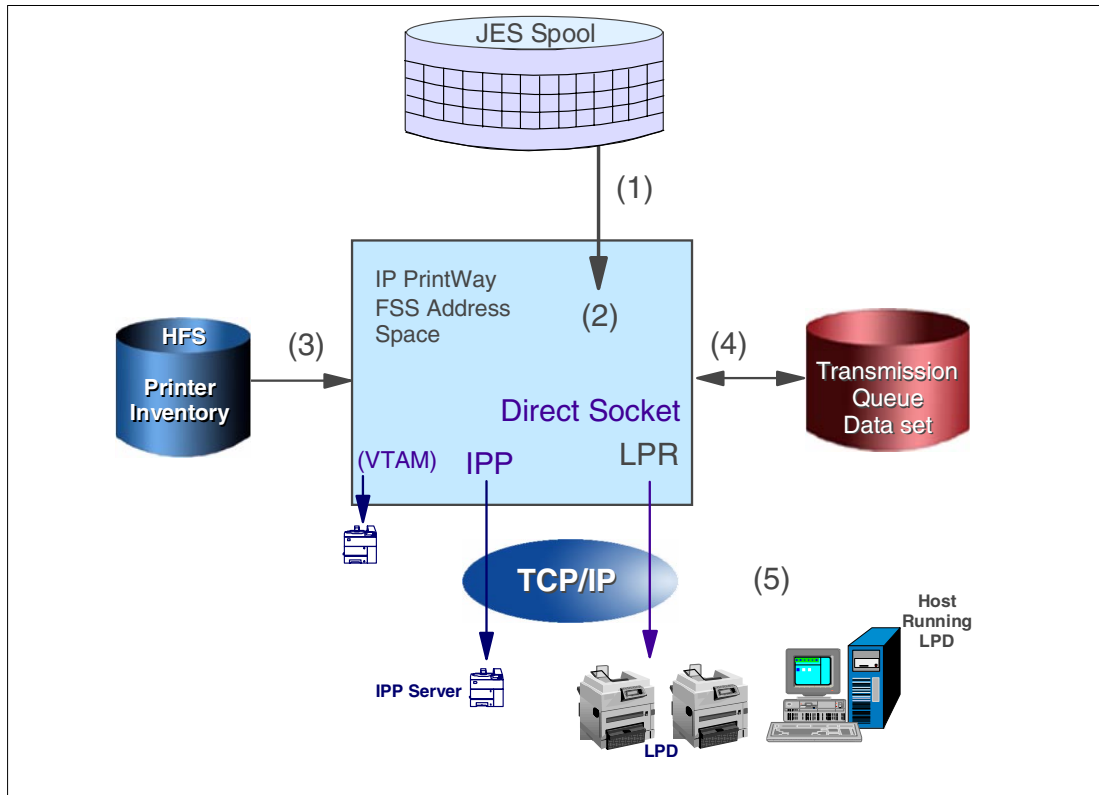


Figure 1-11 IP PrintWay FSA selects a data set from the spool and sends it to the target printer

IP PrintWay can also transmit output data sets from the JES spool to VTAM-controlled printers that are defined to VTAM as LU0, LU1, or LU3 printers. The supported output data streams are:

- ▶ Data Stream Compatibility/Data Stream Extended (DSC/DSE)
- ▶ SNA Character String (SCS).

Note: This support in IP PrintWay requires that you install the Coax Printer Support feature of Infoprint Server Transforms.

1.7 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) enables a management station to configure, monitor, and receive trap messages from network devices. The administrator can enable SNMP reporting for a PSF-controlled printer in the FSA definition for the PSF printers in the Printer Inventory.

The SNMP subagent reports the status of printers that do not have their own SNMP subagent to a printer management program, for example, IBM Network Printer Manager for the Web (NPM). NPM allows an operator to monitor printers throughout the network from a Web browser running on any workstation.

The SNMP subagent allows an operator to use the NPM GUI to monitor the status of printers throughout the system from a Web browser running on a single workstation. When something goes wrong, the interface permits the operator to check for details about any printer. Users can also check the status of printers before they submit jobs.

The Network Printer Manager server, shown in Figure 1-12, requires Windows NT. The GUI can run on any workstation with a Web browser that supports Java 1.1, for example, Netscape Navigator (in Netscape Communicator 4.51) or Microsoft Internet Explorer 5.0.

NPM supports up to 25 printers per z/OS connection.

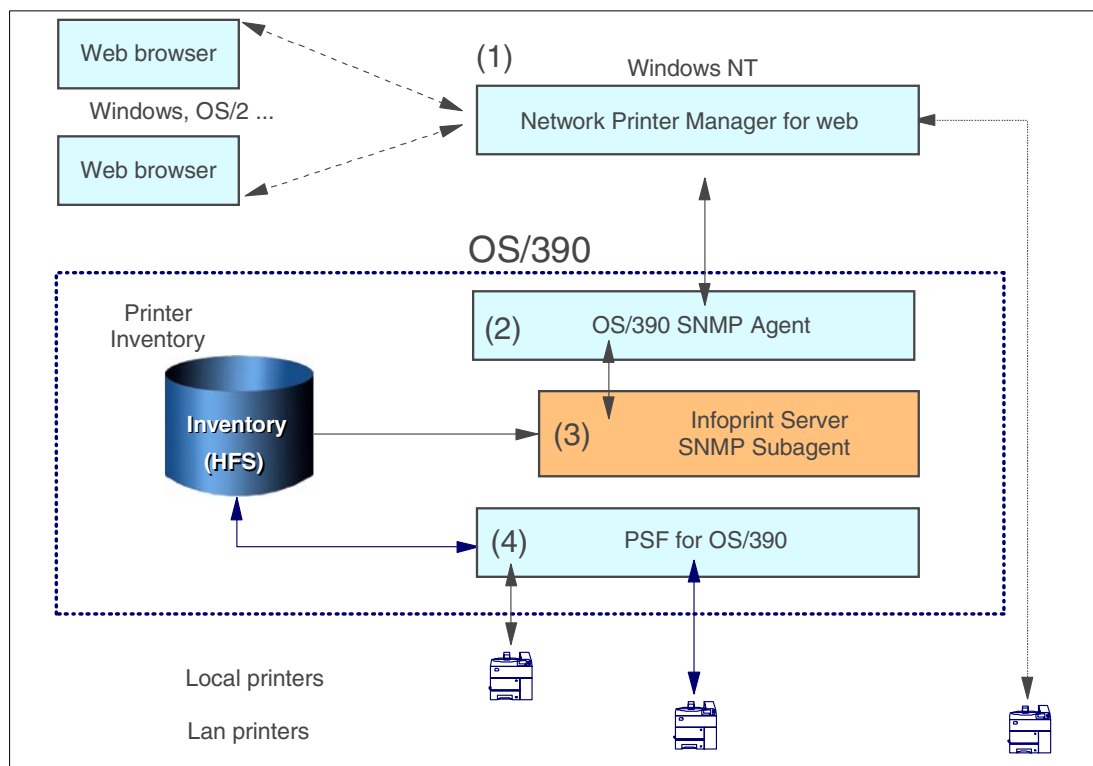


Figure 1-12 Simple Network Management Protocol environment

1.8 Windows workstation support

Infoprint Server supports three client programs for Windows systems that allows Windows users to print documents on AFP printers and other z/OS printers. The three client programs for Windows workstations, shown in Figure 1-13 on page 17, are:

1. OS/390 Printer Port Monitor

The OS/390 Printer Port Monitor for Windows 95/98 and Windows NT sends files submitted for printing from Windows applications to the Print Interface component of Infoprint Server. The OS/390 Printer Port Monitor lets users print from Windows applications that support printing to any printer that the administrator has defined in the Printer Inventory.

2. AFP Printer Driver

The AFP Printer Driver for Windows 95/98, Windows NT, and Windows 2000 creates output files in AFP format, so that Windows users can print on AFP printers. The AFP Printer Driver can create output files that contain documents, overlays, or page segments. It can also create inline form definitions for printing documents with special options, such as printing on both sides of the paper.

3. AFP Viewer plug-in

The AFP Viewer plug-in for Windows 95/98, Windows NT, and Windows 2000 allows Windows users to view files in AFP format using an Internet browser. Users can also print from the AFP Viewer plug-in.

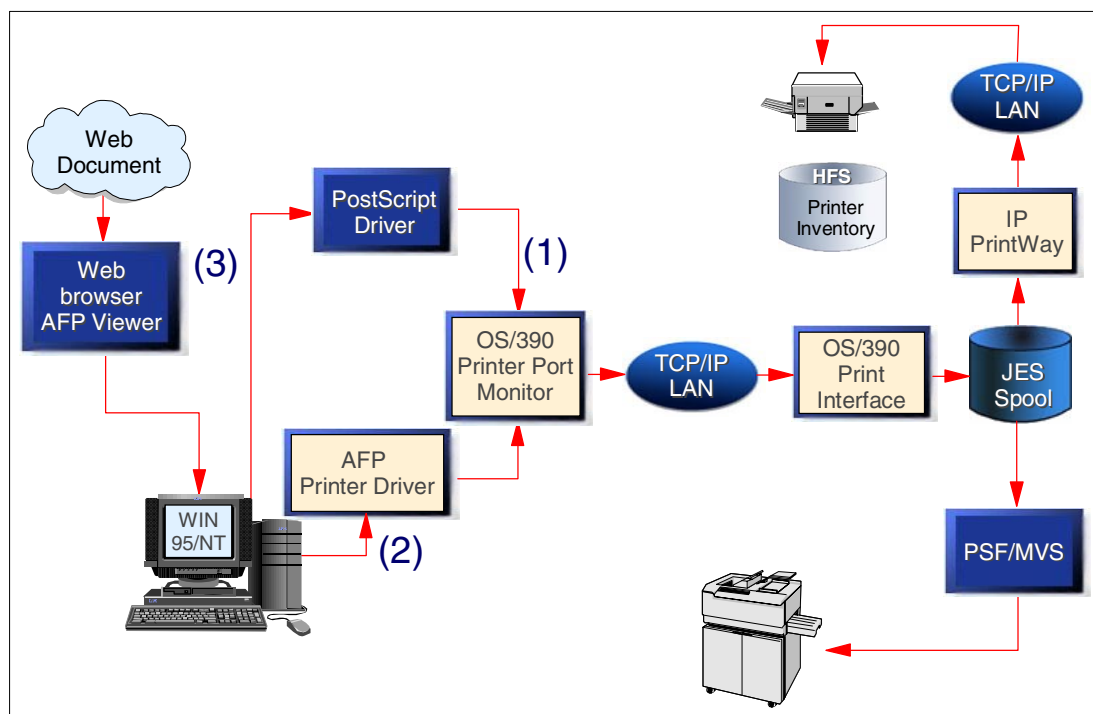


Figure 1-13 Windows workstation user has access to three client programs

1.8.1 Printing from Windows to z/OS

For Windows users who do not download the IBM client programs, Print Interface also includes support for the following Windows printing protocols:

- SMB, used by Windows 95, Windows 98, Windows NT, and Windows 2000
- Internet Printing Protocol (IPP), used by Windows 2000

Note: Windows SMB support is available via APAR OWxxxxx for OS/390 Release 8 and is supported in the product beginning with OS/390 Release 9.

Therefore, Windows users can use standard print submission methods to print to z/OS printers without installing the OS/390 Printer Port Monitor. However, installing the OS/390 Printer Port Monitor may improve performance. Also, the OS/390 Printer Port Monitor lets users specify Infoprint Server job attributes.

Note: For customization of SMB support, see “DFS and SMB support” on page 159.

1.9 Internet Printing Protocol (IPP)

IPP is an application-level protocol that can be used for distributed printing using Internet tools and technologies.

The Internet Printing Protocol provides the capability for a workstation user to install an IPP client and use the Internet to print a document using HTTP protocol from the workstation. The user can print the document in the following ways, as shown in Figure 1-14 on page 19:

- ▶ Directly to a printer that has an IPP server installed, shown as Step (1).
- ▶ To a host that has an IPP server active with Infoprint server, Step (2), and then have the document printed by either:
 - An AFP printer using PSF for OS/390
 - A network printer via TCP/IP using the IP PrintWay using an IPP client to print to a printer with an IPP server installed
 - A network printer via TCP/IP using the IP PrintWay to a printer in the IP network
 - A local printer

In Figure 1-14 on page 19, a user from a Windows 2000 workstation submits a document for printing using a wizard printer with the IPP printing protocol, and using an IPP client that is included in Windows 2000. The file is transmitted over the Internet across the TCP/IP LAN to the Print Interface component, the IPP Server, shown as Step(2).

The Print Interface component creates an output data set on the JES spool and assigns the attributes to the print data set that is defined in the Printer Inventory for the printer that the user had selected. The JES output attributes determine to which printer the output is to be printed for either PSF for OS/390 (Step 2a), IP PrintWay (Step 2b), or JES.

If the selected printer is a remote PostScript printer, the IP PrintWay component transmits the output data set to the PostScript printer on the TCP/IP LAN, Step (2c).

If the printer selected, such as the Infoprint 21, supports the IPP protocol, the IP PrintWay component can communicate with the printer's IPP server (Step 2d) using an IPP client.

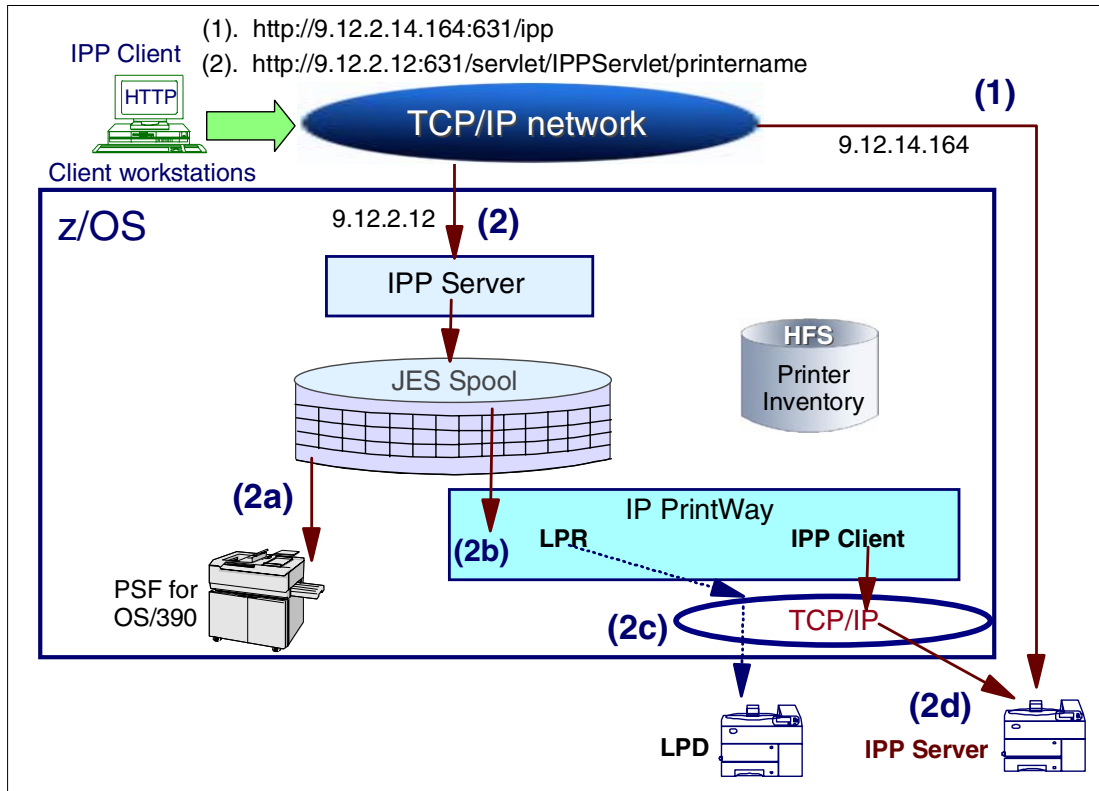


Figure 1-14 Internet Printing Protocol from a user workstation



Infoprint Server customization

This chapter describes the system programmer customization needed for Infoprint Server. Since Infoprint Server is a UNIX application, knowledge of z/OS UNIX System Services is necessary to complete the customization.

This chapter contains information on how to customize and describes the following components and customization steps:

- ▶ Printer Inventory Manager
- ▶ Defining security for Infoprint Server
 - Run the **aopsetup** shell script
 - Set the security environment for operators starting Infoprint Server
- ▶ UNIX System Services file structure
 - Customizing the HFS directories
- ▶ Customization of Infoprint Server configuration files
- ▶ Infoprint Server environment variables
- ▶ Infoprint Server ISPF application customization
- ▶ Starting the Printer Inventory Manager
- ▶ TCP/IP customization
- ▶ Starting and stopping Infoprint Server
- ▶ Print Interface customization
- ▶ AOPPRINTJCL procedure customization
 - AOPSTART JCL procedure
- ▶ Print Interface customization
- ▶ AOPBATCH utility program
- ▶ Starting Infoprint Server

2.1 Printer Inventory Manager

All Infoprint Server components use a common Printer Inventory managed by the Printer Inventory Manager. The Printer Inventory consists of HFS files that contain information about the z/OS print environment.

The Printer Inventory Manager is a key component because in order to use the functions provided by Infoprint Server, you must define the printer environment in the Printer Inventory. Infoprint Server provides the following tools to manage the Printer Inventory, as shown in Figure 2-1:

- ▶ ISPF panels to create the printer definitions
- ▶ The Printer Inventory Definition Utility (PIDU) to back up the Printer Inventory on a regular basis
- ▶ A migration program to help an administrator migrate from previous releases of IP PrintWay, NetSpool, and the OS/390 Print Server

These tools can be used to create printer definitions required by all the following components:

- ▶ IP PrintWay
- ▶ NetSpool
- ▶ SNMP subagent
- ▶ PSF for OS/390

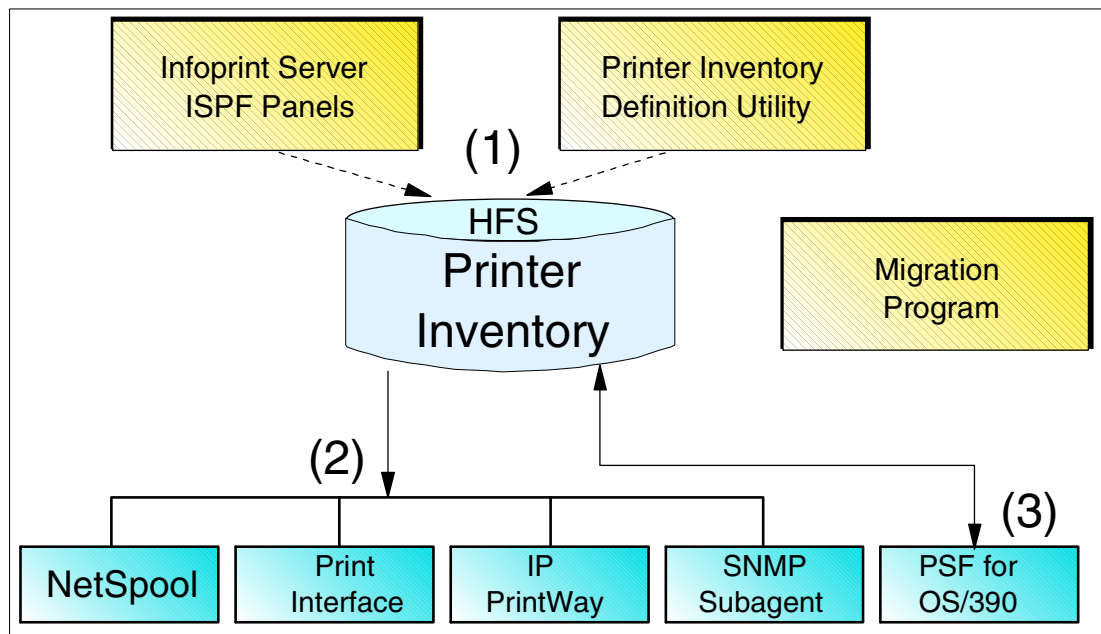


Figure 2-1 Infoprint Server tools used to create component definitions in the Printer Inventory

Note: The Printer Inventory cannot be shared by any Infoprint Server components running at the same or different levels on other systems or be shared between systems in the same sysplex.

You can create the following types of definitions in the Printer Inventory:

Printer A printer definition is required for each printer to which NetSpool, Print Interface, or IP PrintWay can print. The same definition contains all printer information required by NetSpool, Print Interface, and IP PrintWay.

Component To manage a large number of printer definitions, you can create components. Components contain printer information common to more than one printer definition; one component can be included in multiple printer definitions.

Printer pool Printer pool definitions list the printer definitions to which NetSpool can broadcast data.

FSS and FSA These definitions define IP PrintWay or PSF configuration information.

You must customize and start the Printer Inventory Manager before you can create entries in the Printer Inventory.

Once the Printer Inventory Manager is customized, you can start Inforprint Server by using the *aopstart* command from the OMVS command line or an AOPSTART procedure in proclib. Either option starts the Printer Inventory Manager daemon (**AOPD**) which creates an address space necessary to support this environment.

When users send print files to the Print Interface and then to the JES spool for print processing, the printer they specify must be defined in the Printer Inventory.

2.1.1 Printer Inventory

The Printer Inventory Manager component of Inforprint Server maintains the Printer Inventory file, *master.db*, in */var/Printsrv*. This file includes object definitions for:

- ▶ All printer definitions to be placed in the Printer Inventory for the following components:
 - Print Interface
 - NetSpool
 - IP PrintWay
 - PSF for OS/390
- ▶ Printer pool definitions, which contain information about printer groups for NetSpool
- ▶ FSA definitions, which contain configuration information for IP PrintWay and PSF for OS/390 functional subsystem applications (FSAs)
- ▶ FSS definitions, which contain configuration information for IP
- ▶ PrintWay and PSF for OS/390 functional subsystems (FSSs)

The Printer Inventory Manager tools used to create and maintain the Printer Inventory are as follows:

- ▶ An Inforprint Server ISPF application to be used to define the printers. Each print request specifies the name of a printer definition for the target printer in the Printer Inventory. Print Interface uses information in the printer definition to determine how to process the data, whether or not to transform the data, and so on.
- ▶ A Printer Inventory Definition Utility program (PIDU) for the administrator to create and manage entries in the Printer Inventory. See “PIDU export command” on page 225 for details on creating a backup of the Printer Inventory.

2.1.2 ISPF application customization

To be able to use the ISPF panels to create printer definitions, you need to do the following:

- ▶ Modify the TSO/E logon procedure to include the REXX exec, panel, and message libraries or alternatively create a REXX exec or CLIST that dynamically creates the Inforprint Server ISPF application environment and invokes the AOPINIT REXX exec.

- Modify the AOPINIT exec for your environment. This exec initializes default data set names for the Print Interface Path and Trace data sets. These default names can be set the same for all users, or they can be set to different values for different user IDs.

Note: The customization for the logon procedure and the ISPF environment can be done by referring to the z/OS Program Directory or by using *z/OS Infoprint Server Customization*, G544-5694.

2.2 Infoprint Server security

The Infoprint Server uses the System Authorization Facility (SAF) to determine which users can read and update the Printer Inventory and which users can use the **aopstart** and **aopstop** operator commands that start and stop the Infoprint Server.

The System Authorization Facility (SAF) provides a system interface that conditionally directs control to the Resource Access Control Facility (RACF), if RACF is present, or a user-supplied processing routine when receiving a request from a resource manager. Another external security product may be substituted for RACF. SAF, however, does not require any other program product as a prerequisite, but overall system security functions are greatly enhanced and complemented by the concurrent use of an external security product.

You must implement the steps described in this section to enable security checking in Infoprint Server; otherwise, users cannot read and update the Printer Inventory or start Infoprint Server daemons.

2.2.1 Define Infoprint Server administrators and operators

Define Infoprint Server administrators and operators to the security product as z/OS UNIX users. An Infoprint Server administrator must be able to read and update the Printer Inventory. The examples that follow use the AOPADMIN RACF group name for administrators; however, you can use any name. Connect Infoprint Server administrators to the collection (hereafter called group) of Infoprint Server administrators.

The user profiles of the Infoprint Server administrators and operators must have an OMVS segment and a home directory. Establish an OMVS segment and define the home directory to RACF using the RACF **ADDUSER** command to add a new user or the **ALTUSER** command to change an existing user, or what ever methods are currently in use to do security definitions.

Note: Instead of RACF commands you could use the ISPF RACF application to create the OMVS segment.

Administrators

The administrators should be the people who install and maintain via SMP/E the product. They should customize and define the printer environment.

The group profile must have an OMVS segment and a z/OS UNIX group identifier (GID). The GID should be unique in your installation. The RACF command to define group AOPADMIN is as follows:

```
ADDGROUP (AOPADMIN) OMVS (GID(group-identifier))
```

You may add as many users to this group as needed. You can use existing users or create new users, for example, AOPADM1 and AOPADM2.

```
ADDUSER AOPADM1 PASSWORD(xxxxxx) +
```

```

NAME('xxxxxxxx xxxxx') +
OWNER(xxxxxx) UACC(ALTER) DFLTGRP(AOPADMIN) +
AUTHORITY(JOIN) GRPACC SPECIAL OPERATIONS +
TSO(ACCTNUM(xxxxxxxx) PROC(xxxxxxxx) SIZE(xxxxx) MAXSIZE(0) +
UNIT(xxxxxxxx)) +
OMVS(UID(user-identifier) HOME('/u/userid') PROGRAM('/bin/sh'))

```

Note: In the examples, specify the *user-identifier* as an integer that is different from any other UIDs in your installation.

Operators

Infoprint Server operators can start and stop Infoprint Server using the **aopstart** and **aopstop** commands or the **AOPSTART** and **AOPSTOP** procedures.

The suggested group name for Infoprint Server operators is AOPOPER; however, you can use any name. The group profile must have an OMVS segment and a z/OS UNIX group identifier (GID). The GID should be unique.

RACF group command example:

```

ADDDGROUP (AOPOPER) OMVS(GID(group-identifier))

```

Define as many users as you need.

```

ADDUSER AOPOPR1 PASSWORD(xxxxxx) +
NAME('xxxxxxxx xxxxx') +
OWNER(xxxxxx) UACC(ALTER) DFLTGRP(AOPOPER) +
AUTHORITY(JOIN) GRPACC SPECIAL OPERATIONS +
TSO(ACCTNUM(xxxxxxxx) PROC(xxxxxxxx) SIZE(xxxxx) MAXSIZE(0) +
UNIT(xxxxxxxx)) +
OMVS(UID(user-identifier) HOME('/u/userid') PROGRAM('/bin/sh'))

```

Note: If your installation's security policy does not require a distinction between Infoprint Server administrators and operators, you can skip this step and use the group name for Infoprint Server administrators in the **aopsetup** step.

2.2.2 Define resource profile AOPADMIN in the FACILITY class

An Infoprint Server administrator can read and update the Printer Inventory. To prevent any other users from accessing the Printer Inventory, who are not part of the AOPADMIN group, use the RACF FACILITY class to protect the Inventory.

Define a resource profile to the security product named AOPADMIN in the FACILITY class (the name *must be* AOPADMIN).

```

RDEFINE FACILITY AOPADMIN UACC(NONE)

```

Users must have read (or higher) access to the AOPADMIN resource profile in order to view or update the Printer Inventory. Give Infoprint Server administrators read access to the AOPADMIN resource profile. This is done by using the RACF PERMIT command to authorize users to the resource profile, for example:

```

PERMIT AOPADMIN CLASS(FACILITY) ACCESS(READ) ID(AOPADM1)

```

When you define these profiles, or any time you add new profiles, they must be refreshed in storage by issuing the following RACF command:

```

SETROPTS RACLIST(FACILITY) REFRESH

```

2.2.3 Create user environment

For each user you define for administrators or operators, you should do the following:

- ▶ Create a home directory
- ▶ Create a user HFS data set
- ▶ Mount the user HFS

HOME directory

The `$HOME/.profile` file (where `$HOME` is a variable for the home directory for your individual user ID) is an individual user profile. Any values in the `.profile` file in your home directory that differ with those in the `/etc/profile` file override them during your shell session. z/OS provides a sample individual user profile.

For each user, create the user's home directory as `(/u/userid)`. This allows for the capability to set up for the user a `$HOME/.profile` for defining login environment parameters for that user. This allows all the Infoprint Server environment variables to be set in the `/etc/profile` file and to apply them for all users on the Z/OS system. The administrator has specific environment variables in a `.profile` file.

To create the user's home directory `(/u/userid)`, and set the directory permissions to give the user read, write, and execute access to the directory, use the z/OS UNIX `mkdir` command or the z/OS automount facility. For example:

```
cd /u
mkdir -m 0700 userid
```

Allocate a user HFS data set

You can use a batch job, as shown in Figure 2-2, or you can use the ISPF shell or the TSO/E `ALLOCATE` command to create an HFS data set.

```
//STEP01 EXEC PGM=IEFBR14
//HFS      DD DSN=OMVS.AOPADM1.HFS,SPACE=(CYL,(20,1,1)),
//          DSNTYPE=HFS,DCB=(DSORG=PO),
//          DISP=(NEW,CATLG,DELETE),
//          STORCLAS=OPENMVS
```

Figure 2-2 Allocate a user HFS data set for user AOPADM1

Mount the HFS data set

If you are not using the automount facility, issue a `mount` command to mount the user HFS.

```
MOUNT FILESYSTEM('OMVS.AOPADM1.HFS') TYPE(HFS) MOUNTPOINT('/u/aopadm1')
```

After mounting the new file system for the first time, change the owner and group owner. These values are saved in the new file system and are reused when the file system is remounted later. Use the `chown` command to set these values. For the `/u/aopadm1` directory, to set the user name and group name as the owner, use:

```
chown -R user /u/aopadm1
```

Note: You can use the OMVS ISPF Shell TSO application panels instead of the z/OS UNIX `mkdir` and `chown` commands to create the user's directory and HFS data set.

Figure 2-3 on page 27 shows the mounted HFS data sets for several administrators and operators.

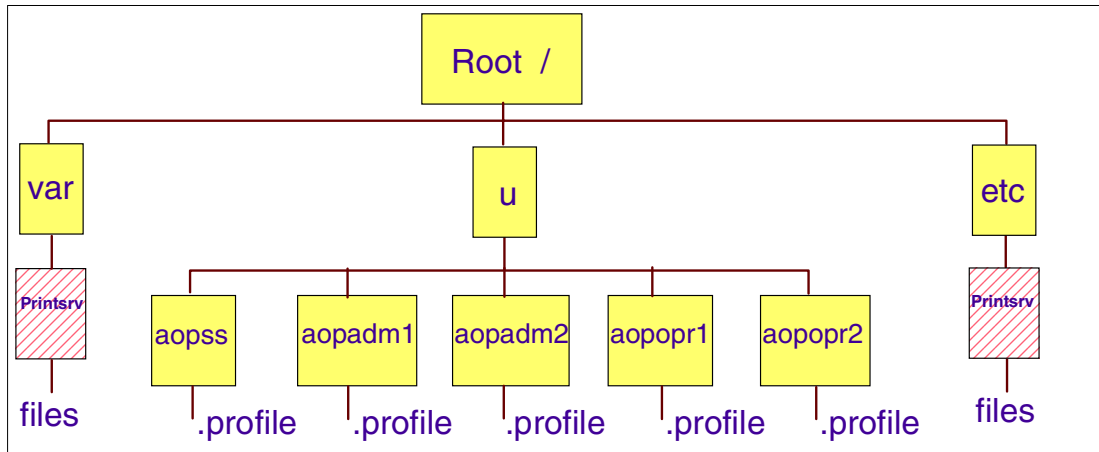


Figure 2-3 Administrators and operators defined to Infoprint Server

Automount facility

You can allocate the user's home directory to a user HFS data set and use the automount facility to manage the user HFS data set. The automount facility lets you designate directories as containing only mount points. As each mount point is accessed, an appropriate file system is mounted. The mount point directories are internally created as they are required. When the file system is no longer being used, the mount point directories are deleted.

Note: See *z/OS UNIX System Services Planning*, GA22-7800 for more information on setting up the automount facility

2.3 Run the aopsetup shell script

Run the **aopsetup** shell script to define permissions. You can run **aopsetup** from an rlogin shell, from an OMVS session, or with the BPXBATCH command. Specify the names of the groups you defined for Infoprint Server operators and administrators as arguments to **aopsetup**. For example, if you defined one group to RACF named AOPOPER for operators and another group named AOPADMIN for administrators, enter:

```
/usr/lpp/Printsrv/bin/aopsetup AOPOPER AOPADMIN
```

If you use the same group name for both Infoprint administrators and operators, specify the name twice on the **aopsetup** command.

Note: Run **aopsetup** from a user ID that is a superuser (UID of 0) or has read access to the (RACF) BPX.SUPERUSER resource in the FACILITY class.

2.4 Operator control of starting Infoprint Server

We recommend that you define a security product user ID for the AOPSTART and AOPSTOP started tasks that can be used to start the Infoprint Server if you plan to use the Internet Printing Protocol daemon (IPPD). This user ID should be dedicated for the starting and stopping of the Infoprint Server. It allows for all of the Java (required for IPPD) and environment variables to be set at startup without having to define them in the */etc/profile*, which is common to all users.

2.4.1 Define a user ID for the AOPSTART and AOPSTOP JCL procedures

The AOPSTART and AOPSTOP JCL procedures use the **aopstart** and **aopstop** commands to start and stop the Inforprint Server. Use a security product such as RACF to define a user ID, **aopss** in the example, as follows:

1. Define the user ID into the already defined AOPOPER group and assign an unused UNIX user identifier (UID):

```
ADDUSER aopss OMVS(UID(8826) HOME('/u/aopss') PROGRAM('/bin/sh')) DFLTGRP(AOPOPER)
NOPASSWORD
```

The preferred approach is used to create the user's home directory (*/u/userid*), which makes it possible to set up a user *\$HOME/.profile*.

The user IDs that you assign to started procedures should have the PROTECTED attribute. Protected user IDs have both the NOPASSWORD and NOOIDCARD attributes.

You may need to connect the *aopss* user to the RACF group that protects the */usr/lpp/Printsrv/bin/aopstart* and */usr/lpp/Printsrv/bin/aopstop* files. Connecting the user to the RACF group is preferred to changing the permissions from 750 to 755.

```
CONNECT (aopss) GROUP(AOPOPER)
```

When you connect the *aopss* user to the **aopstart** and **aopstop** command group owner, you do not have to remember to run the **aopsetup** command every time you apply maintenance to the Inforprint Server because the user who applies the maintenance may change the owner of the **aopstart** and **aopstop** commands in the directory.

2. Define the start and stop procedures to the RACF STARTED class

Using the STARTED class instead of ICHRIN03 avoids updating ICHRIN03 and a re-IPL of the system in order to add or modify RACF identities for started procedures. Installations that want to use the started procedure table (ICHRIN03) instead, should refer to *z/OS SecureWay Security Server RACF System Programmer's Guide*, SA22-7681. Use the following RACF commands:

```
SETRPTS GENERIC(STARTED) - (if not already set)
SETRPTS CLASSACT(STARTED) RACLIST(STARTED) - (if not already set)
RDEFINE STARTED AOPSTART.** STDATA(USER(aopss) GROUP(AOPOPER) TRUSTED(NO))
RDEFINE STARTED AOPSTOP.** STDATA(USER(aopss) GROUP(AOPOPER) TRUSTED(NO))
SETRPTS RACLIST(STARTED) REFRESH
```

Allocate the HFS data set for user AOPSS

Figure 2-4 on page 28 shows an example of the JCL to allocate a HFS data set for the */u/aopss* directory. The HFS data set name, *OMVS.AOPSS.HFS*, shown in Figure 2-4, should comply with your automount policy if it is being used.

```
//STEP01 EXEC PGM=IEFBR14
//HFS DD DSN=OMVS.AOPSS.HFS,SPACE=(CYL,(5,1,1)),
// DSNTYPE=HFS,DCB=(DSORG=PO),
// DISP=(NEW,CATLG,DELETE),
// STORCLAS=OPENMVS
```

Figure 2-4 JCL to allocate a user HFS data set

You can use the TSO MKDIR and MOUNT commands to create and mount the */u/aopss* directory:

```
MKDIR '/u/aopss' MODE(7 5 5)
MOUNT FILESYSTEM('OMVS.AOPSS.HFS') TYPE(HFS) MOUNTPoint('/u/aopss') MODE(RDWR)
```


Create a .profile file for user AOPSS

The AOPSTART JCL procedure executes the TSO terminal monitor program IKJEFT01 in the background. The TMP invokes the **OSHELL** command which invokes BPXBATCH from TSO/E and executes the **aopstart** command. During the login to the UNIX shell the *AOPSS* user's **.profile** file is executed to define the environment variables for this individual user. This allows a consistent start up of the Infoprint Server and also allows multiple releases of the same product, Java, to coexist without requiring all the environment variables to be defined in the */etc/profile*. More information may be found in the *z/OS UNIX System Services User's Guide*, SA22-7801 in the chapter on customizing the shell.

You may use the TSO OEDIT command to edit the *aopss* user's **.profile** file. Paths to the following files should be placed into the **.profile** file if you start the **ippd** daemon. The files are in the */usr/lpp/Printsrv/classes* directory.

```
ipp.jar
ippclient.jar
ippreal.jar
ippserver.jar
```

To use the IPP server, you must mark the following Java run-time libraries APF-authorized:

```
libjava.a
libjipc.so
libzip.so
libnet.so
```

If you installed Java 1.1.8, also mark these Java run-time libraries APF-authorized:

```
libjvmei.so
libJava.WMEH.so
```

Later versions of Java might require that you APF-authorize different run-time libraries. A sample job is provided in SYS1.SAMPLIB(AOPJAUTH) to APF-authorize Java libraries. IBM recommends that you use this job because it APF-authorizes the correct Java libraries for the level of Java that is installed on your z/OS system. Instead of using the AOPJAUTH sample job, you can use the z/OS UNIX **extattr** command to APF-authorize libraries.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
SC657      /u/aopss/.profile
Command ==>
***** ***** Top of Data ****
000001 # This is a aopss' .profile defining login environment parameters
000002
000003 # These lines add the CLASSPATH variables for IPP Java
000004 CLASSPATH=/usr/lpp/Printsrv/classes/ipp.jar
000005 CLASSPATH=$CLASSPATH:/usr/lpp/Printsrv/classes/ippserver.jar
000006 CLASSPATH=$CLASSPATH:/usr/lpp/Printsrv/classes/ippclient.jar
000007 CLASSPATH=$CLASSPATH:/usr/lpp/Printsrv/classes/ippreal.jar
000008 CLASSPATH=$CLASSPATH:/usr/lpp/java18/J1.1/lib/classes.zip
000009
000010 # This line sets the LIBPATH environment variable for Java
000011 LIBPATH=$LIBPATH:/usr/lpp/java18/J1.1/lib/mvs/native_threads
000012
000013 export CLASSPATH
000014 export LIBPATH
000015 echo PATH $PATH
000016 echo LIBPATH $LIBPATH
000017 echo CLASSPATH $CLASSPATH
000018 echo JAVA_HOME $JAVA_HOME
000019
000020 # Set the prompt to display your login name, and current directory.
000021 PS1='$LOGNAME': '$PWD': ' >'
000022
000023 # Sets emacs as command line recall
000024 set -o emacs
000025
000026 # Export the variable settings so that they are known to the system.
000027
000028 export _BPX_SHAREAS="YES"
000029 export PS1

```

Figure 2-5 Create a .profile for starting the AOPIPPD daemon for user aopss

Once the *aopss* user's *.profile* is properly defined, other authorized users can execute it to define Infoprint environment when they need to issue **aopstart** command from the UNIX shell. Assuming that you have set up automount facility, you can issue the commands as shown in Figure 2-6 on page 31 in a UNIX shell to start Infoprint Servers.

If you are planning to use the IPP daemon, we recommend that you add the **REGION=0M** parameter on the **EXEC** statement in **AOPSTART** when starting the Infoprint Server from started task JCL.

```

VAINI @ SC65:/>cd /u/aopss
VAINI @ SC65:/u/aopss>. ./profile
PATH /usr/lpp/Printsrv/bin:/bin:/pp/java18/J1.1/bin:...
LIBPATH /usr/lpp/Printsrv/lib:/lib:/usr/lib:/pp/java18/J1.1/lib/mvs/native_threa...
CLASSPATH /usr/lpp/Printsrv/classes/ipp.jar:/usr/lpp/Printsrv/classes/ippserver...
JAVA_HOME /pp/java18/J1.1
VAINI:/u/aopss: >aopstart
aopd: AOP075I Daemon aoplpd was started successfully.
aopd: AOP075I Daemon aopxpd was started successfully.
aopd: AOP075I Daemon aoppipd was started successfully.
AOP075I Daemon aopd was started successfully.

```

Figure 2-6 Using the aopstart command to start Infoprint Server daemons

2.5 UNIX System Services file structure

Beginning with OS/390 Release 9, the non-sysplex sharing HFS structure is changed as shown in Figure 2-7. The directories */dev*, */tmp*, */var*, and */etc* are now symbolic links.

To access the directories you need for customization if your system is Release 9 or higher, you must enter the following from either the ISHELL or the OMVS shell:

```

/etc/
/var/

```

The / following var and etc is required with the new directory structure. Of course, if you are accessing more than one directory, you just enter:

```

/var/Printsrv

```

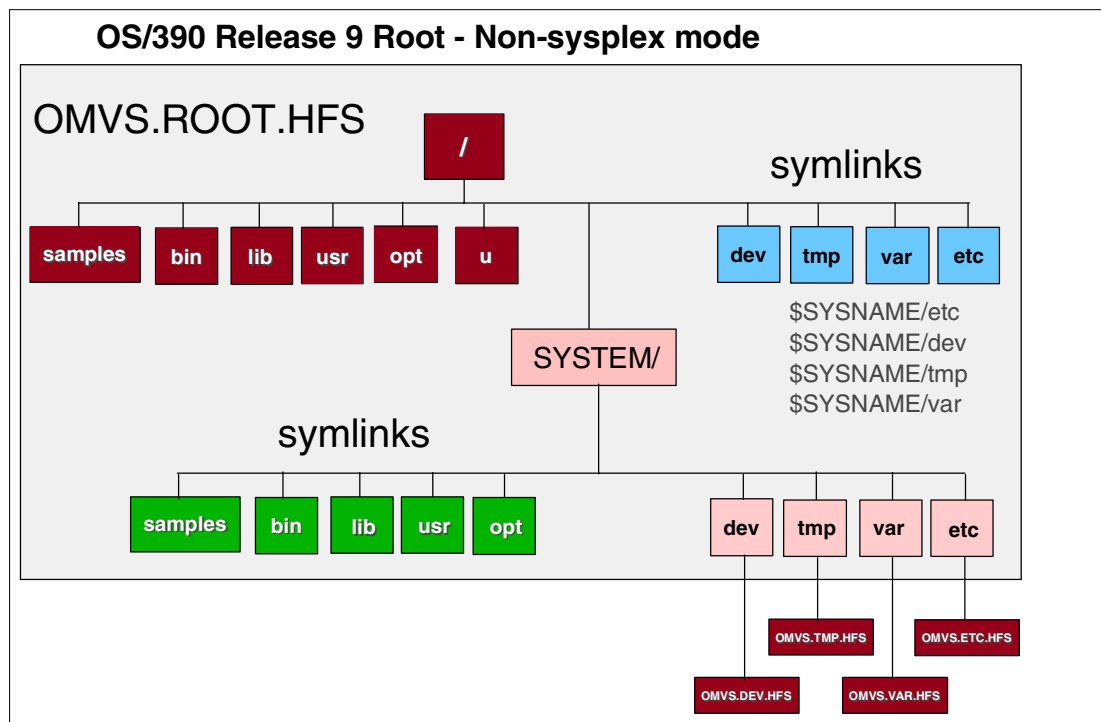


Figure 2-7 UNIX System Services file structure

2.5.1 Customize the HFS directories

As shown in Figure 2-8, the following customization steps should be done to configure the HFS and the configuration files you require for your installation:

- Step (1)** The first step in the customization is to create the `/etc/Printsrv` directory. You can use the UNIX `mkdir` command under the `/etc` directory or the ISHELL to create the `/etc/Printsrv` directory. This directory is the default location for the Infoprint Server configuration files, `aopd.conf`, `aopxfd.conf`, and `aopsapd.conf`.
- Step (2)** The second step is to create the `/var/Printsrv` directory. Issue the UNIX `mkdir` command or use the ISHELL to create the directory. If you do not create this directory, the `aopsetup` shell script will create it. See “Run the `aopsetup` shell script” on page 27.
- Step (3)** The third step sets up the Infoprint Server configuration files for use. Copy the sample configuration files from `/usr/lpp/Printsrv/samples` to the default location `/etc/Printsrv/` with the z/OS UNIX `cp` command or use the ISPF ISHELL. You can choose to copy the configuration file into another location; however, if you do, specify the full path name of the configuration file in the AOPCONF environment variable in the `/etc/profile` file.

Note: IBM requires that you mount a separate file systems at the `/var` and `/etc` mount points, as shown in Figure 2-7 on page 31. Create the `/var/Printsrv` directory in its own file system. Starting with OS/390 Release 9 this is required for both non-sysplex configurations and sysplex sharing configurations.

The `/var/Printsrv` directory contains the Printer Inventory files. The transforms can also create temporary files in subdirectories of this directory; therefore, this directory must have enough space to contain all data sets that are transformed at the same time.

If you want to use a different directory name, specify the directory name in the *base-directory* attribute in the `aopd.conf` configuration file.

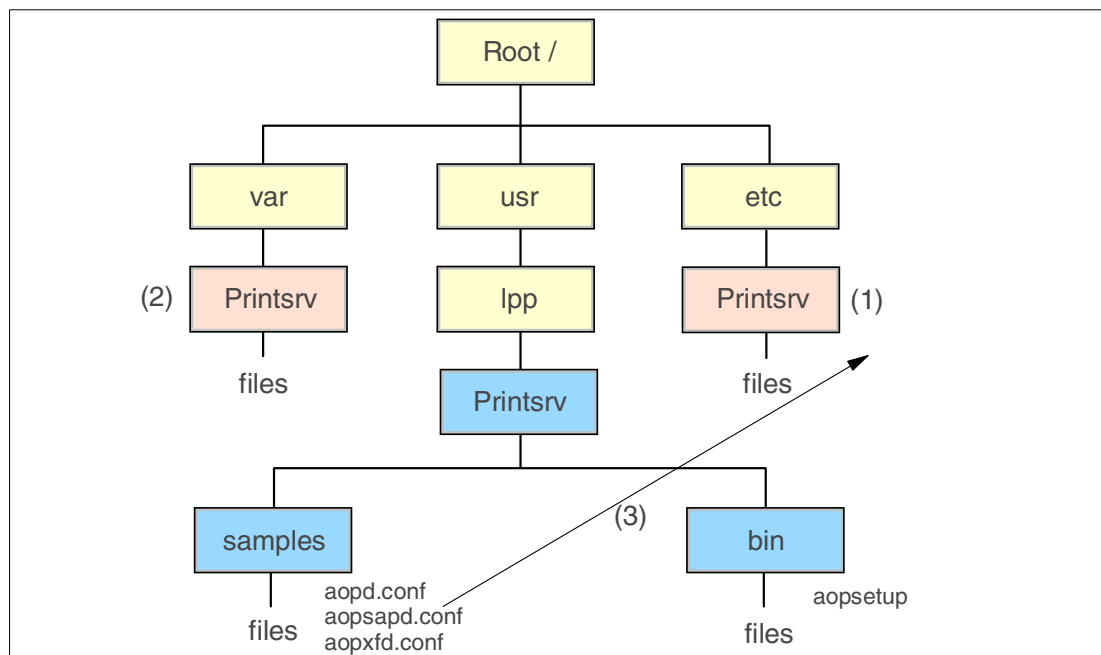


Figure 2-8 HFS directory structure for Infoprint Server directories and files

2.6 Infoprint Server configuration files

z/OS UNIX applications, like the Infoprint Server, require the use of certain configuration files. Customers are responsible for providing these files for the application to use. IBM provides default configuration files for Infoprint Server in the `/usr/lpp/Printsrv/samples` directory.

The configuration files contain information that is needed to start and activate specific functions and are required if the functions are used. There are three configuration files for use with Infoprint Server:

- | | |
|---------------------|---|
| aopd.conf | This file contains attributes that customize Infoprint Server. Figure 2-9 shows the sample configuration file for Infoprint Server. The Infoprint Server configuration file includes customization attributes for the Printer Inventory Manager and other components of Infoprint Server. This file is <i>optional</i> ; if the configuration file does not exist or if an attribute in the configuration file is omitted, default values are used. |
| aopsapd.conf | The SAP Callback daemon configuration file contains information that the SAP Callback daemon needs to log on to the SAP R/3 application servers. <i>Use this file if you require printing from SAP R/3.</i> |
| aopxfd.conf | The transform configuration file contains information that the Transform Manager uses to manage the PCL to AFP transform and also other transforms. <i>Customize this file if you plan to use the data stream transforms.</i> |

Before the first use of Infoprint Server, you must copy these IBM-provided samples to the `/etc/Printsrv` directory. Further customization of these files to include installation-dependent information should be added by you if needed.

2.6.1 Customization of configuration file **aopd.conf**

Before you can create entries in the Printer Inventory and before you can use any other Infoprint Server component, you must complete the customization of the **aopd.conf** configuration file.

If you start Infoprint Server with no *start-daemons* defined in the configuration file, only the Printer Inventory Manager will be started.

Figure 2-9 shows the sample configuration file. The **aopd.conf** configuration file includes several attributes that may need some customization for the Print Interface (**AOPLPD**) and Printer Inventory Manager (**AOPD**) daemons.

```
#-----  
# aopd.conf - Default Infoprint Server configuration  
#-----  
lpd-port-number = 515  
ipp-port-number = 631  
base-directory  = /var/Printsrv  
ascii-codepage  = ISO8859-1  
ebcdic-codepage = IBM-1047  
job-prefix      = PS  
inventory       = AOP1  
start-daemons  = { lpd }  
snmp-community  = public
```

Figure 2-9 Default Infoprint Server configuration file

ascii-codepage The name of an ASCII code page supported by z/OS. For code page names, refer to *z/OS C/C++ Programming Guide*, SC09-4765. The name specified here is displayed in the *Printer code page* field in the ISPF panel, shown in Figure 2-10 on page 34, when defining a new IP PrintWay printer definition. It is used as the source document code page when the Print Interface translates documents received from remote systems, unless a document code page is specified in the printer definition or by the job submitter, and as the target printer code page when it translates PostScript documents created with a non-ASCII code page.

Default: ascii-codepage = **ISO8859-1**

Note: When defining printers, as shown in Figure 2-10 on page 34, the code page field is not required or you can overtype a different code page. However, if the field is blank, Print Interface does not convert data, so documents submitted from remote systems or with the **1p** command might not print correctly.

Examples: ASCII: ISO8859-1, IBM-850 EBCDIC: IBM-1047, IBM-500

```

Add                                     Processing
Command ==> _____

Printer definition name . poke2 _____

Document code page . . _____
Printer code page. . . ISO8859-1 _____

Print Interface supported data formats and associated filters:
Data format:  Filter:
/ Line data   _____ (extend)
/ M0:DCA-P    _____ (extend)
/ PostScript  _____ (extend)
/ Text        aopfiltr.so (extend)
/ PCL         _____ (extend)
/ PDF         _____ (extend)
/ SAP         _____ (extend)
/ Other       _____ (extend)

_ Resubmit for filtering

Conversion between SCS and Line:
Margins: Top . . . . . Bottom . . . . . Left . . . . . Right . . . . .
Line length . . . . . Page length . . . . .
Tabs: Vertical . . . . . (extend)
      Horizontal . . . . . (extend)

IP PrintWay Line-to-Text Conversion:
/ Pagination
  Margins: Top . . . . . Bottom . . . . .
  
```

Figure 2-10 Processing options when defining IP PrintWay printers to the Printer Inventory

ebcdic-codepage The name of the code page used by the printers you define that are PSF for OS/390, General printers, and IP PrintWay printers using VTAM protocol. Print Interface print converts print data to EBCDIC as required by the target printer when data is received and allocates output data sets on the JES spool. Print Interface uses this code page as the source (document) code page, when Print Interface cannot determine the user locale for the **1p** command.

inventory While you can change this name, it is not recommended. If you do change the name assigned to the Printer Inventory, you must change the start procedures for IP PrintWay, NetSpool, and PSF for OS/390 to specify this name on the EXEC statement.

	Default: inventory = AOP1
base-directory	Specifies the directory path in which the Printer Inventory Manager creates Printer Inventory files. Also, Print Interface creates files in this directory. Default: base-directory = /var/Printsrv
start-daemons	<p>The Infoprint Server daemons that are started when you issue the aopstart command. You can specify one or more daemon names, enclosed in braces. The Printer Inventory Manager (AOPD) daemon always starts, regardless of which daemons you specify in this attribute. To start only the Printer Inventory Manager daemon, type braces with no daemon names.</p> <p>ippd - The aopipdd daemon. This IPP server daemon processes print jobs submitted by an Internet Printing Protocol (IPP) client.</p> <p>lpd - The aoplpd daemon. This line printer daemon processes print jobs submitted with commands, such as lpr, and by the OS/390 Printer Port Monitor for Windows.</p> <p>snmpd - The aopsnmpd daemon. This SNMP subagent daemon provides status information to the OS/390 SNMP agent about printers controlled by PSF for OS/390.</p> <p>xfd - The aopxfd daemon. This Transform Manager daemon manages the PCL and PostScript (and PDF) to AFP transforms provided by Infoprint Server Transforms for OS/390.</p> <p>In UNIX, a daemon is a long-lived process that runs unattended to perform continuous or periodic system functions. Some daemons are started automatically to perform their task while others operate periodically. The MVS equivalent is a started systems task.</p>
lpd-port-number	<p>The number of the port at which the Infoprint Server LPD waits for print requests. Port 515 is the well-known port for communication between LPRs and LPDs. If you specify a port other than 515, ensure that the port is not used by any other service on the z/OS system and that you customize the LPRs to send print requests to the new port. Windows users must specify this port when configuring the OS/390 printer port monitor on a Windows system. This attribute is optional.</p> <p>You must assign port 515 to the Print Interface LPD if you want IP PrintWay to transform data from one format to another. This port number must not be reserved in the PROFILE.TCPIP data set.</p> <div data-bbox="602 1493 1459 1738" data-label="Text"> <p>Note: TCP/IP ships in hlq.SEZAINST(SAMPPROF) a sample configuration file for the TCP/IP. The sample profile includes a definition for port 515: 515 TCP LPSEVER ;LPD Server. If you use the sample profile for TCPIP, you must change the port definition to: 515 TCP OMVS, comment it out, or remove it to enable Infoprint Server LPD use of the port 515. See <i>z/OS Communications Server IP Configuration Guide</i>, SC31-8775 for additional considerations.</p> </div>
	Default: lpd-port-number = 515
ipp-port-number	The number of the port at which the IPP server waits for print requests. Port 631 is the well-known port for communication between IPP clients and IPP servers. If you specify a port other than 631, ensure that the port

is not used by any other service on the z/OS system. This attribute is optional.

This port number must not be reserved in the PROFILE.TCPIP data set.

Default: ipp-port-number = **631**

job-prefix

A prefix added to the job identifier that Print Interface generates for data sets it allocates on the JES spool. This prefix can help the operator identify data sets on the JES spool allocated by Print Interface. Specify two letters, numbers, or national (@ \$ #) characters; the first character cannot be numeric. The Print Interface job identifier is not the same as the JES job identifier. The JES job identifier is assigned by JES and is part of the fully-qualified data set name.

Default: job-prefix = **PS**

snmp-community

The name assigned to the SNMP community for making SNMP requests to the z/OS system. If you specified a community name to the z/OS SNMP agent, specify the same community name in this statement. If you did not provide a community name to the z/OS SNMP agent, the SNMP agent accepts requests from any SNMP manager with the community name of public. Specify 1-32 uppercase or lowercase letters, numbers, or special characters; however, note that the CS for z/OS SNMP agent and the SNMP manager might restrict the length of the community name or the allowed characters. The community name is case sensitive.

Default: snmp-community = **public**

2.7 Setting Infoprint Server environment variables

Infoprint Server environment variables may be defined in the */etc/profile* file so that they apply for all users on the z/OS system, or you can set up command text that sets and exports the variables. If you change the values while any Infoprint Server daemons are running, the settings do not change for the current login shell, daemons, or other processes already started.

You can edit the */etc/profile* file with the TSO/E OEDIT command and the z/OS UNIX **oedit** command. To set the environment variables required by the Printer Inventory Manager, assuming that you installed Infoprint Server libraries in the default locations, add the following commands to the */etc/profile* file:

```
export AOPCONF=/etc/Printsrv/aopd.conf
export LIBPATH=/usr/lpp/Printsrv/lib:$LIBPATH
export MANPATH=/usr/lpp/Printsrv/man/%L:$MANPATH
export NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
export PATH=/usr/lpp/Printsrv/bin:$PATH
```

To display the current value of an environment variable, use the z/OS UNIX **echo** command, for example:

```
echo $PATH
```

The Printer Inventory Manager environment variable descriptions:

AOPCONF

The full path name of the Infoprint Server configuration file. This environment variable is *optional*; if you created the configuration file in */etc/Printsrv/aopd.conf*, you do not need to set this environment variable.

AOPPATH	The directory that contains attribute files used by the lp command. This environment variable is <i>optional</i> ; if you have not created attribute files for the lp command, you do not need to set this environment variable.
LIBPATH	The path used to find dynamic link library (DLL) files. This environment variable is <i>required</i> . If you installed Infoprint Server files in the default directory, add /usr/lpp/Printsrv/lib to any existing values. If you installed Infoprint Server files in a different directory, add the directory to any existing values.
MANPATH	<p>The path of directories that contain the man pages. This environment variable is <i>required</i>. If the LANG environment variable identifies the language in which you want to view Infoprint Server man pages, add /usr/lpp/Printsrv/man/%L to the values in this variable; otherwise, add one of the following values to any existing values:</p> <p>English /usr/lpp/Printsrv/man/En_US Japanese /usr/lpp/Printsrv/man/Ja_JP Spanish /usr/lpp/Printsrv/man/Es_ES</p> <p>Note: Add the new directory path <i>before</i> /usr/man/%L in the MANPATH environment variable, so that the Infoprint Server versions of the lp, lpstat, and cancel man pages are displayed.</p>
NLSPATH	<p>The path of directories that contain message catalogs. This environment variable is <i>required</i>. If the LANG environment variable identifies the language in which you want to receive Infoprint Server messages, add /usr/lpp/Printsrv/%L/%N to the values in this variable; otherwise, add one of the following values:</p> <p>English /usr/lpp/Printsrv/En_US/%N Japanese /usr/lpp/Printsrv/Ja_JP/%N Spanish /usr/lpp/Printsrv/Es_ES/%N</p> <p>%L represents the value of the LANG environment variable. %N is the catalog filename.</p>
PATH	The path used to locate executables. This environment variable is <i>required</i> . If you installed Infoprint Server executables in the default directory, add /usr/lpp/Printsrv/bin to the existing values.

2.7.1 Customization of UNIX shell print commands

Print Interface provides enhanced z/OS UNIX shell printing commands **lp**, **lpstat**, and **cancel**.

Important: Make sure that **/usr/lpp/Printsrv/bin** is added in the directory *before* **/bin** in the PATH environment variable.

Set command path in /etc/profile

For all z/OS UNIX users to be able to use the modified UNIX print commands to communicate directly with Print Interface, **/etc/profile** should have the following statement:

```
# This sets a default command path, including your current working
# directory (CWD).
PATH=/usr/lpp/Printsrv/bin:/bin:.
```

If other command paths are needed, just make sure that **/usr/lpp/Printsrv/bin** appears before **/bin**.

Environment variables in /etc/profile

The Printer Inventory Manager daemon has to be active to run these commands. You can optionally set the following **lp** command environment variables in the */etc/profile* file:

AOOPTIONS	Specifies a string of job attributes and values that are to be in effect for each lp command. The lp command includes the attributes specified in this environment variable before the attributes specified, if any, on the -o option of the lp command. Because the lp command reads the value of the AOPTIONS environment variable before the options you specify on the command line, a user can override the values of this variable. This variable is <i>optional</i> and can be used to specify job attributes that are constant for all print jobs. Refer to <i>z/OS Infoprint Server User's Guide</i> , S544-5746 for the format of the job attributes and values.
PRINTER	The default printer for the lp command. The value in LPDEST overrides the value in PRINTER.
LPDEST	This variable is <i>optional</i> . You can also use Infoprint Server ISPF panels to define a default printer. See <i>z/OS Infoprint Server Operation and Administration</i> , S544-5745. The printer named in either LPDEST or PRINTER overrides the default printer named on the ISPF configuration panel.

2.8 Infoprint Server ISPF application customization

The Infoprint Server administrator can use Infoprint Server ISPF panels to add, browse, copy, edit, and delete printer definitions and other objects in the Printer Inventory.

Infoprint Server ISPF panels are available in English, Japanese, and Spanish. The Japanese version of the panels has been translated to code page IBM-939; the Spanish version of the panels has been translated to code page IBM-284.

2.8.1 Define Infoprint Server Printer Inventory Manager to ISPF

The Printer Inventory Manager ISPF application requires the following libraries (application data elements) either through the dynamic ISPF LIBDEF service or as part of the ISPxLIB DD statement concatenation:

- ▶ ISPLLIB, depending on the language in which you want to view the panels:

English	AOP.SAOPPENU
Japanese	AOP.SAOPPJPN
Spanish	AOP.SAOPPESP
- ▶ ISPMLIB, depending on the language in which you want to receive messages.

English	AOP.SAOPMENU
Japanese	AOP.SAOPMJPN
Spanish	AOP.SAOPMESP

2.8.2 The AOPINIT exec

Before using the ISPF panels, review and customize the AOPINIT exec provided in AOP.SAOPEXEC. The Printer Inventory Manager ISPF application REXX EXEC library, AOP.SAOPEXEC, is required to be included into the SYSPROC or SYSEXEC DD statement concatenation, or be dynamically available through the TSO ALTLIB service.

Note: If you have not added the Language Environment run-time library (CEE.SCEERUN) or the C++ run-time library (CBC.SCLBDLL) to the system LNKLIST, concatenate these data sets to the ISPLLIB DD statement, concatenate them to STEPLIB, or use theTSOLIB service before you invoke ISPF to make the libraries available for your TSO session.

To customize your ISPF environment for Infoprint Server, you can choose several options:

- The ISPF *z/OS System Programmer Primary Option Menu* (member ISR@390S in the ISP.SISPPENU library) provides a sample of how to invoke the Printer Inventory Manager ISPF application. You can update your ISPF primary option menu to include a selection option for the Printer Inventory Manager, as shown in Figure 2-11. Assuming that you have the required libraries concatenated to the appropriate DD statements, add SELECT into the panel.

```
)AREA SAREA39
::::::::::::
.8 .Infoprint Srv.Infoprint Server for OS/390      .
::::::::::::
)INIT
.ZVARS = '(ZCMD)'
.HELP = ISR00003
)PROC
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
::::::::::::
8, 'CMD(AOPINIT &ZTRAIL) NEWAPPL(AOP) SCRNAME(INFOPRT) NOCHECK'
::::::::::::
' ' ' '
  ,
  *,'?' )
```

Figure 2-11 ISPF Primary Option Menu SELECT for Printer Inventory Manager

- We recommend that you use the example in Figure 2-12 on page 40, which shows a sample REXX exec that dynamically activates the Printer Inventory Manager ISPF application environment and invokes the application. You can use the supplied AOPINIT without any changes.

```

/* AOPIN -- REXX -- Init AOPINIT Environment */
Trace "0"
/* Setup AOPINIT libs */
Address ISPEXEC
  "LIBDEF ISPMLIB DATASET ID('AOP.SAOPMENU')"
  "LIBDEF ISPPLIB DATASET ID('AOP.SAOPPENU')"
Address "TSO"
  "ALTLIB ACT APPL(EXEC) DS('AOP.SAOPEXEC') "
/* Suppress IGD104I message for Selection 6 */
x = OUTTRAP("OF."); "PROF LIST"; x = OUTTRAP("OFF");
do i = 1 to of.0; if wordpos("WTPMSG",of.i) <> 0 then do;
  "PROF NOWTPMSG"; wtp = 1; leave; end; else wtp = 0;
end
/* Invoke AOPINIT rexx */
Address ISPEXEC
"CONTROL ERRORS RETURN"
"SELECT CMD(AOPINIT) NEWAPPL(AOP) PASSLIB "
/* Undo AOPINIT environment */
"LIBDEF ISPMLIB "
"LIBDEF ISPPLIB "
Address "TSO"
  if wtp then "PROF WTPMSG"
  "ALTLIB DEACT APPL(EXEC)"
Exit 0

```

Figure 2-12 Sample REXX exec to invoke the Printer Inventory Manager ISPF application

2.8.3 Primary ISPF panel for Infoprint Server

The sample REXX exec, or any option you choose, invokes the primary ISPF application panel, shown in Figure 2-13. This panel manages the inventory of printer information that is used by NetSpool, the Print Interface, IP PrintWay, and PSF for OS/390. This panel lets you add, display, update, or select printer definitions; manage functional subsystem application (FSA), functional subsystem (FSS), and printer definition pool parameters; view the IP PrintWay transmission queue and message log; and change panel configuration parameters.

```

AOPIPP1                      Infoprint Server: Printer Inventory Manager
Option ==> _____

Printer Definitions
  1 Add                      Add a printer definition
  2 List                     List printer definitions
  3 Select                   Select printer definitions to list

Other Functions
  4 FSS/FSA/Pool            Manage other inventory definitions
  5 PrintWay Queue          View IP PrintWay transmission queue
  6 PrintWay Message        View IP PrintWay message log
  7 Configure               Change panel configuration

```

Figure 2-13 Infoprint Server ISPF application primary option menu

Change panel configuration customization

The AOPINIT exec sets default values for fields that are displayed on the Infoprint Server Configuration panel, Option 7 in Figure 2-13. You can change any of the following values, shown in Figure 2-14 on page 41, in the AOPINIT exec. To modify AOPINIT, follow the instructions in the comments in AOPINIT to make your changes.

AOIPCF Configuration	
Command ==>	
<hr/>	
/	Confirm delete requests
_	Trace Printer Inventory ISPF internals
ISPF trace directory.	/var/Printsrv/trace
<hr/>	
Printer Inventory:	
Configuration file	/etc/Printsrv/aopd.conf
NLS path	/usr/lpp/Printsrv/%L/%N
Language	En_US
Default printer	FIJVBIN
<hr/>	
IP PrintWay:	
Message log	INFOPRNT.SC43.MSGFILE
Trace data set	ROGERS.ANFISPF.TRACE

Figure 2-14 Configuration panel displayed from Option 7

The parameters are as follows:

- ▶ Printer Inventory Manager ISPF trace file: The name of the file to contain the trace of ISPF processing. The default file name is */var/Printsrv/trace*.
- ▶ Configuration file: The path name of the Infoprint Server configuration file. The default file name is */etc/Printsrv/aopd.conf*.
- ▶ NLS path: The directory path that contains the Infoprint Server message catalog. If the LANG environment variable identifies the language in which you want to receive Infoprint Server messages, specify */usr/lpp/Printsrv/%L/%N*; otherwise, add one of the following values:
 - English */usr/lpp/Printsrv/En_US/%N*
 - Japanese */usr/lpp/Printsrv/Ja_JP/%N*
 - Spanish */usr/lpp/Printsrv/Es_ES/%N*

The default is */usr/lpp/Printsrv/%L/%N*.

- ▶ Language: The value of the LANG environment variable. The default is *En_US*.
- ▶ Default printer: The name of the printer definition that the lpstat command uses as the default printer. Also, the name of the printer definition that lp command uses if the PRINTER and LPDEST environment variables are not set. The default is *lp1*.
- ▶ IP PrintWay message log data set: The name of the data set where IP PrintWay writes messages. The default is *ANF.MESSAGE*.
- ▶ IP PrintWay ISPF trace data set: The name of the data set to contains the trace of IP PrintWay ISPF processing. The default is *userid.ANFISPF.TRACE*.

Note: Individual ISPF users can also change the values for their ISPF sessions on the Configuration panel itself.

2.9 Starting the Printer Inventory Manager

You do not have to start TCP/IP to start the Printer Inventory Manager daemon (aopd) or the Transform Manager daemon (aopxfd).

Attention: At this stage of the Infoprint Server customization you can issue the **aopstart** command to start the Printer Inventory Manager with the Infoprint Server configuration file (*aopd.conf*) start-daemons set to **start-daemons = { }** and let the Infoprint Server administrators add printer definitions.

2.10 TCP/IP customization

You must perform some z/OS IBM Communications Server (TCP/IP) customization before you can submit print requests to Print Interface from remote systems or to use IP PrintWay to transmit print jobs to remote printers that use any of the following TCP/IP printing protocols:

- ▶ Internet Printing Protocol (IPP)
- ▶ Server Message Block (SMB) protocol

To print to TCP/IP-attached printers using IP PrintWay, you must customize TCP/IP. When you customize TCP/IP, you customize the following data sets:

- ▶ TCPIP.DATA
- ▶ PROFILE.TCPIP

2.10.1 TCP/IP LPD and Print Interface LPD

The Remote Print LPD server supports the Line Print Daemon and allows you to print on JES-controlled printers from any host in your TCP/IP network that implements the Line Print client functions. These client functions are invoked with the LPR command. LPR is available as a TSO command, and the LPD server is implemented as a started z/OS task.

Note: This was the support before the introduction of the Print Server and Infoprint Server.

Therefore, there are two LPD servers that could potentially use the same port 515:

- ▶ Standard TCP/IP LPD
- ▶ Print Interface LPD

If you were using the standard TCP/IP LPD support before installing the Infoprint Server or Print Server, and still require it in addition to the Infoprint Server, see “One TCP/IP stack sharing the same port” on page 43 or “Multiple TCP/IP stacks” on page 43.

TCP/IP LPD and Print Interface LPD on the same z/OS system

You can run the Print Interface LPD and the TCP/IP LPD on the same z/OS system. There are three TCP/IP configuration options:

1. *One TCP/IP stack*

One TCP/IP stack serving both the Print Interface LPD and the TCP/IP LPD.

In this case, you must assign different port numbers to the Print Interface and TCP/IP LPDs to distinguish between them. Because you must assign port 515 to the *TCP/IP* LPD, you must assign a *different* port to the Print Interface LPD in the **lpd-port-number** attribute in the *aopd.conf* file.

Different port numbers may not be desirable because:

- Most LPR clients cannot be configured to send print requests to a port other than the well-known port 515. However, Windows users can configure the Port Monitor to send requests to another port. Therefore, if you assign a *different* port to the Print Interface

LPD, remote users will *not* be able to use most LPR commands -- they must use the OS/390 Printer Port Monitor for Windows.

- You must assign port 515 to the Print Interface LPD if you want */P* PrintWay to use the Infoprint Server Transforms to transform data from one format to another.

2. Multiple TCP/IP stacks

Use multiple TCP/IP stacks, each with its own IP address, or a fully qualified domain name. When the Print Interface LPD and the TCP/IP LPD run on *different* stacks, you can assign port 515 to both the Print Interface LPD and the TCP/IP LPD.

For example, if you configure two stacks, one named TCPIP and another named TCPIPOE, and you want Infoprint Server to use the TCPIPOE stack, set `_BPX_SETIBMOPT_TRANSPORT=TCPIPOE` in */etc/profile* for use by all z/OS UNIX applications.

3. One TCP/IP stack sharing the same port

One TCP/IP stack with two HOME IP addresses serving both the Print Interface LPD and the TCP/IP LPD, as shown in Figure 2-15.

A BIND keyword on a PORT statement can be used to force a generic server (one that binds to INADDR_ANY) to bind to the specific IP address that is specified following the BIND keyword. The IP address that follows BIND must be in IPv4 dotted decimal format and can be any valid address for the host including VIPA and dynamic VIPA address. For multiple servers to bind to the same port with this function, the IP address for each server must be unique. See *z/OS Communications Server IP Configuration Guide*, SC31-8775 and *IBM Communications Server IP Configuration Reference*, SC31-8776 for more details.

Attention: APAR PQ37421 adds support to TCP/IP Version 2 for OS/390 Release 8 and 9. This APAR provides new function that will allow multiple dissimilar servers to use the same TCP port. For instance, both TN3270 Telnet and Unix Telnet can use port 23, as shown in Figure 2-15.

Figure 2-15 shows a portion of the PROFILE data set. One TCP/IP stack is shown in the PROFILE data set with two home addresses. The BIND keyword on the port 515 statement reserves the port for the Infoprint Server Line Print daemon LPD. The Print Interface LPD must run on the OMVS stack.

```
HOME
  9.12.6.113    OSAL2100 ; WTSC65.ITS0.IBM.COM
  9.12.6.125    OSAL2160 ; WTSC650E.ITS0.IBM.COM
PORT
  23 TCP INTCLIEN          ; Telnet 3270 Server
  23 TCP OMVS BIND 9.12.6.125 ; OE telnet server
  515 TCP LPSEVER          ; LPD Server
  515 TCP OMVS BIND 9.12.6.125 ; Infoprint LPD stack for UNIX
```

Figure 2-15 Statements from the TCP/IP PROFILE data set for sharing port 515

2.10.2 TCP/IP PROFILE considerations

The TCP/IP PROFILE data set contains configuration statements used during TCP/IP address space initialization. You can use the **VARY TCPIP, ,OBEYFILE** command to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCP/IP address space.

The following TCPCONFIG and PORT statements in the TCP/IP PROFILE configuration data set are important for Print Interface and IP PrintWay:

► TCPCONFIG:

- INTERVAL minutes SENDGARBAGE TRUE

IP PrintWay relies on TCP to detect when a connection with the receiving system is no longer usable by sending keep-alive packets; therefore, if you plan to use IP PrintWay, consider specifying a shorter interval than the default, which is approximately two hours.

For example, consider specifying an interval of 10 minutes. Note that the interval you specify applies to all TCP applications that direct TCP to send keep-alive packets.

Specify SENDGARBAGE TRUE if any target host requires that the keep-alive packet contain data.

- UNRESTRICTLOWPORTS

Ports 1 through 1023 are not reserved. Both Print Interface and IP PrintWay must be able to obtain a port in the 0 through 1023 range without having that port explicitly reserved for their use; therefore, specify this parameter to ensure that the low ports are not reserved.

- TCPRCVBFRSIZE buffersize

The TCP receive buffer size. IP PrintWay requires a buffer size of at least 32768.

- TCPSENDBUFRSIZE buffersize

The TCP send buffer size. IP PrintWay requires a buffer size of at least 32768.

► PORT:

The port numbers that are used by the Print Interface LPD, the IPP server, and the OS/390 SNMP agent must not be reserved in the TCP/IP PROFILE data. The default port numbers are:

- Print Interface LPD: port 515
- Print Interface IPP server: port 631
- OS/390 SNMP agent: ports 161 and 162

2.11 Starting and stopping Infoprint Server

The **aopstart** command starts the Printer Inventory Manager daemon, **aopd**. It also starts any other Infoprint Server daemons specified in the **start-daemons** attribute in the **aopd.conf** configuration file.

The **aopstop** command stops either the Printer Inventory daemon and any other active Infoprint Server daemons, or it stops only selected Infoprint Server daemons, depending on the command options.

The **aopstart** command does not have any options. The full format of the **aopstop** command is:

aopstop [-d daemon]... [now]

-d daemon Specifies the daemon to stop. If you omit this option, all daemons that are active are stopped, including the Inventory Manager daemon. You can specify one of these values:

ippd Stops the Internet Printing Protocol daemon, **aopippd**.

lpd Stops the line printer daemon, LPD.
snmpd Stops the SNMP subagent daemon, aopsnmpd.
xfd Stops the Transform Manager daemon, aopxfd.

now Stops the daemons immediately. If work is in progress, incorrect output or data loss may result. If this option is not specified, the daemons stop after current activity completes.

If you have set up all the Printer Inventory daemon and Infoprint Server daemons environment variables in */etc/profile*, the **aopstart** and **aopstop** commands can be entered as follows:

- ▶ From the z/OS UNIX shell, where you can enter **aopstart** and **aopstop**.
- ▶ As a recommended alternative, beginning with OS/390 Release 8, you can use JCL procedures to invoke the **aopstart** and **aopstop** commands. The JCL procedures in SYS1.IBM.PROCLIB that Infoprint Server ships are named AOPSTART and AOPSTOP.
- ▶ Add the **aopstart** command to the */etc/rc* shell script to start the Printer Inventory Manager (and other daemons) automatically during the IPL.

2.11.1 Infoprint Server start considerations

Before starting the Infoprint Server daemons with a **start-daemons={ lpd ippd xfd }**, make sure that the following conditions have been established:

- ▶ The Infoprint environment variables have been set in */etc/profile* or set through
- ▶
- ▶
- ▶
- ▶ **\$HOME/.profile:**

```
export AOPCONF=/etc/Printsrv/aopd.conf
export LIBPATH=/usr/lpp/Printsrv/lib:$LIBPATH
export MANPATH=/usr/lpp/Printsrv/man/%L:$MANPATH
export NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
export PATH=/usr/lpp/Printsrv/bin:$PATH
```

- ▶ Ensure that TCP/IP has finished initialization before starting the line printer daemon (AOPLPD), the IPP server daemon (AOIPPD), or the z/OS SNMP agent.

Note: You do not have to start TCP/IP to start the Printer Inventory Manager daemon (*aopd*) or the Transform Manager daemon (*aopxfd*).

- ▶ Start the Printer Inventory daemon before starting IP PrintWay, Netspool, PSF for OS/390, or the Printer Inventory Manager ISPF applications.

Important: The region size required to execute the Infoprint Server daemons is important.

- ▶ The daemons that require the larger memory sizes are:
 - The IPP Server daemon (IPPD) - 83 MB
 - The Transform Manager Daemon (AOPXFD) - 42 MB to 64 MB

2.11.2 Region size considerations for the IPPD daemon

To ensure that sufficient memory is available when you start the IPP server, do one of the following, depending on how you plan to issue the **aopstart** command to start Infoprint Server:

z/OS UNIX command line

From the z/OS UNIX command line, the MAXASSIZE parameter in the BPXPRMxx member of SYS1.PARMLIB determines the system-wide maximum region size for an address space. For example, specify the following statement in the BPXPRMxx member:

```
MAXASSIZE(87041808)    /* 83*1024*1024 = 83MB (minimum)    */
```

You can also set the MAXASSIZE parameter dynamically with the **SETOMVS** command. For example:

```
SETOMVS MAXASSIZE=87041808
```

The ASSIZEMAX parameter of the RACF ADDUSER and ALTUSER commands lets you define the maximum region size for a particular user in the OMVS segment of the user's RACF profile. If the ASSIZEMAX parameter is not specified, the MAXASSIZE value in BPXPRMxx is used. If you specify the ASSIZEMAX parameter for the user ID that starts Infoprint Server, specify at least 87041808.

TSO/E OMVS command line

From the TSO/E OMVS command line, if you issue the **aopstart** command from TSO, the SIZE parameter on the TSO/E LOGON panel determines the maximum region size for an address space. Set the SIZE parameter to 83 MB or larger.

AOPSTART procedure

If you use the AOPSTART procedure to issue the command, the REGION parameter on the EXEC statement in the procedure determines the maximum region size. Specify REGION=83M or higher. If the REGION parameter is not specified, the default region size defined for your installation is used.

2.11.3 IPPD daemon considerations

If you start the IPPD daemon, you must set the following environment variables in */etc/profile* if you do not use the default directories */usr/lpp/Printsrv* and */usr/lpp/java*, as shown in Figure 2-16 and Figure 2-17 on page 47.

```

/usr/lpp/java/
Select one or more files with / or action codes.
  Type  Filename
_ Dir   .
_ Dir   ..
_ File  AJVRAWTA
_ File  AJVRAWTH
_ File  AJVRAWTR
_ File  AJVRAWTW
_ File  AJVSWING
_ File  AJVS390
_ File  AJVTRANS
_ Dir   J1.1
_ Dir   META-INF
_ Dir   RAWT-1.1
_ Dir   Swing-1.1ibm
_ File  Swing-1.1ibm.jar
_ File  translator.sh

```

Figure 2-16 /usr/lpp/java/ directory list

```

/usr/lpp/Printsrv/classes/
Select one or more files with / or action codes.

  Type  Filename
_ Dir   .
_ Dir   ..
_ Dir   IBM
_ File  ipp.jar
_ File  ippclient.jar
_ File  ippreal.jar
_ File  ippserver.jar

```

Figure 2-17 Java CLASSPATH files in the default Infoprint Server directory

2.11.4 AOPJAUTH SAMPLIB member

A sample job to APF-authorize the Java libraries is provided in the SYS1.SAMPLIB AOPJAUTH member. This job is used to customize the IPPD server environment. It marks the following Java libraries as APF-authorized:

- ▶ libjava.a
- ▶ libjtc.so
- ▶ libzip.so
- ▶ libnet.so
- ▶ libjvmei.so
- ▶ libJavaWMEH.so

Important: When maintenance is applied to Java, APF authorization is lost; therefore, you must rerun AOPJAUTH after maintenance is applied to Java.

To run the AOPJAUTH sample job, you must have at least read access to the BPX.FILEATTR.APF FACILITY class profile as follows:

```

RDEFI;;E FACILITY BPX.FILEATTR.APF UACC(NONE)
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) ACCESS(READ) ID(AOPADM1)

```

Also, you must have superuser authority (UID of 0) or you must have access to the BPX.SUPERUSER FACILITY class profile as follows:

```
RDEFINE FACILITY BPX.SUPERUSER UACC(NONE)
PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(AOPADM1)
```

If you edit AOPJAUTH, do not modify the PARM parameter for the AOPBATCH program on the EXEC statement, because data after the STDIN DD statement contains executable z/OS UNIX commands, as shown in Figure 2-18 on page 48. Also, make sure that these lines do not have sequence numbers.

```
//AUTHJAVA EXEC PGM=AOPBATCH,PARM='su'
//STDIN DD *
export PS1=' ' PS2=' ' PS3=' ' PS4=' '
cd /usr/lpp/java/J1.1/lib/mvs/native_threads
extattr +a libjava.a libjtc.so libzip.so libnet.so
if test -e libjvmei.so; then
    extattr +a libjvmei.so
fi
if test -e libJavaWMEH.so; then
    extattr +a libJavaWMEH.so
fi
cd /usr/lpp/java/IBM/J1.3/bin
extattr +a *.so
cd classic
extattr +a *.so
/*
```

Figure 2-18 EXEC statement and STDIN from AOPJAUTH JCL

2.11.5 AOPSTART JCL procedure

Figure 2-19 shows the AOPSTART JCL procedure, which has been modified to include the REGION=0M parameter on the EXEC statement. To execute the procedure you can enter a **S AOPSTART** operator command, or you can submit a batch job that executes the AOPSTART procedure provided your user ID is authorized for the **aopstart** command.

```
/*-----
/*  AOPSTART  This procedure starts the z/OS Infoprint Server
/*           daemons.  The user ID assigned to this proc must be a
/*           member of the AOPOPER group.
/*-----
//AOPSTART  PROC
//AOPSTART  EXEC  PGM=IKJEFT01,PARM='osHELL aopstart',REGION=0M
//SYSPROC   DD    DSN=SYS1.SBPXEXEC,DISP=SHR
//SYSTSIN   DD    DUMMY
//SYSTSPRT  DD    SYSOUT=*
```

Figure 2-19 AOPSTART JCL procedure

2.11.6 AOPSTOP JCL procedure

Figure 2-20 on page 49 shows the AOPSTOP JCL procedure, which has been modified to include OPT= parameter on the PROC statement. The OPT= parameter allows to add options to the **aopstop** command. For example, when you want to stop the Internet Printing Protocol daemon immediately, enter the operator command:

```

S AOPSTOP,OPT='-d ippd now'
or
S AOPSTOP,PARM='osHELL aopstop -d ippd now'

```

See “Starting and stopping Infoprint Server” on page 44 for **aopstop** command options.

```

/*-----
/*  AOPSTOP  This procedure stops the z/OS Infoprint Server
/*          daemons.  The user ID assigned to this proc must be a
/*          member of the AOPOPER group.
/*-----
//AOPSTOP  PROC  OPT=
//AOPSTOP  EXEC  PGM=IKJEFT01,PARM='osHELL aopstop &OPT'
//SYSPROC  DD    DSN=SYS1.SBPXEXEC,DISP=SHR
//SYSTSIN  DD    DUMMY
//SYSTSPRT DD    SYSOUT=*

```

Figure 2-20 AOPSTOP JCL procedure

When you want to restart the daemons that have been stopped, just enter the **S AOPSTART** operator command.

2.11.7 Starting Infoprint daemons from /etc/rc

To start Infoprint Server daemons automatically during system initialization, you can specify the **aopstart** command and required environment variables in the */etc/rc* shell script.

This example shows lines that you can add to */etc/rc* to specify the required environment variables and the **aopstart** command:

```

LIBPATH=/usr/lpp/Printsrv/lib:$LIBPATH \
NLSPATH=/usr/lpp/Printsrv/En_US/%N:$NLSPATH \
PATH=/usr/lpp/Printsrv/bin:$PATH \
aopstart

```

The backslashes indicate that this is a multi-line command. Therefore, the environment variables are effective only for the **aopstart** command. You must start a new line immediately following the backslash; do not type any spaces or characters between the backslash and the new line.

The environment variables shown in the example assume that Infoprint Server files are installed in the default locations. Also, you might need to specify other environment variables, which include:

BPXK_SETIBMOPT_TRANSPORT

The TCP/IP job name associated with the TCP/IP stack you want to use. Set this environment variable if your installation wants to limit Infoprint Server to only one TCP/IP stack.

The following is an example of a port 515 display from a two TCP/IP stack system when the installation **_BPXK_SETIBMOPT_TRANSPORT** variable was *not* set:

EZZ2350I	MVS	TCP/IP	NETSTAT	CS	V2R10	TCP/IP NAME: TCPIPOE	23:21:58
EZZ2585I	User Id	Conn	Local Socket			Foreign Socket	State
EZZ2586I	-----	----	-----			-----	----
EZZ2587I	AOPLPD	00002A55	0.0.0.0..515			0.0.0.0..0	Listen
EZZ2350I	MVS	TCP/IP	NETSTAT	CS	V2R10	TCP/IP NAME: TCPIPMVS	23:23:02

EZZ2585I	User Id	Conn	Local Socket	Foreign Socket	State
EZZ2586I	-----	----	-----	-----	----
EZZ2587I	AOPLPD	00002AF0	0.0.0.0..515	0.0.0.0..0	Listen

AOPCONF Names the Infoprint Server configuration file. The file named in this variable takes precedence over configuration file */etc/Printsrv/aopd.conf*.

AOPXFD_CONF Names the Infoprint Server configuration file used by the Transform Manager. The file named in this variable takes precedence over configuration file */etc/Printsrv/aopxfd.conf*.

CLASSPATH Lists the directories where the IPP server and IPP client files are located.

JAVA_HOME Lists the directory where the Java files used by the IPP server and IPP client are located.

LIBPATH Lists the directory where the Infoprint Server DLLs are located.

NLSPATH Lists the directory where the Infoprint Server message catalogs are located.

PATH Lists the directory where the Infoprint Server daemons are located.

2.12 Print Interface customization

Several customization tasks need to be performed before you submit print requests to Print Interface. Some tasks are required by all installations that want to use Print Interface, while some tasks are optional and only required if the function is needed in your installation.

The tasks that are required are:

- ▶ TCP/IP customization is required if you plan to submit print jobs to Print Interface from remote systems, or if you plan to use IP PrintWay to transmit print jobs to remote printers. See “TCP/IP customization” on page 42.
- ▶ JES blank truncation support, see “JES blank truncation considerations” on page 50.

The tasks that are optional are:

- ▶ AOPPRINT JCL procedure.
- ▶ End users need to know the host IP address and printer names defined in the Printer Inventory that they can use to print in order to use the **lpr** command.

2.12.1 JES blank truncation considerations

IBM recommends that you specify

OUTCLASS(J) **BLNKTRNC=NO** (in JES2)

SYSOUT,CLASS=J,TYPE=PRINT,**TRUNC=NO** (in JES3)

for all output classes in which Print Interface allocates output data sets. The binary data streams (such as PCL and PostScript) received and written by the Print Interface can become corrupted if blanks (EBCDIC x'40') are removed from the ends of the records.

JES3 APAR

Make sure that APAR OW48548 is installed; it affects blank truncation.

Data stream transforms

Important: You must specify *no* blank truncation also when you use the AFP to PCL, AFP to PostScript, and AFP to PDF transforms. If JES truncates trailing blanks in the PCL, PostScript, or PDF output on the JES spool, the output will be corrupted.

Refer to *z/OS JES2 Initialization and Tuning Reference*, SA22-7533 and *z/OS JES3 Initialization and Tuning Reference*, SA22-7550 for more information about blank truncation.

2.12.2 AOPPRINT JCL procedure customization

Print Interface provides the AOPPRINT JCL procedure in SYS1.IBM.PROCLIB so that local users can submit the **lp** command from a batch job. Refer to *z/OS Infoprint Server User's Guide*, S544-5746 for additional information about how to use the AOPPRINT JCL procedure. Figure 2-21 shows the Print Interface AOPPRINT JCL procedure.

```
/*-----  
/* AOPPRINT - z/OS Print Server batch print procedure  
/*-----  
//AOPPRINT PROC PRINTER='p1',OPTIONS=,OUTCLASS='*',ERRCLASS='*'  
//LP      EXEC PGM=AOPBATCH,  
//  PARM='/lp -d &PRINTER -o "&OPTIONS" //DD:SYSIN'  
//STDOUT  DD  SYSOUT=&OUTCLASS  
//STDERR  DD  SYSOUT=&ERRCLASS  
/* STDENV may point to a data set containing environment variables.  
/*      Builtin values will work for the default installation.  
/*STDENV  DD  DSN=USERID.JCL(ENVVARS),DISP=SHR
```

Figure 2-21 Default AOPPRINT JCL procedure in SYS1.IBM.PROCLIB

The AOPPRINT JCL procedure customization includes:

- ▶ To change the name of the printer definition, set the printer definition name in the PRINTER option on the PROC statement to your installation's default printer. The printer name should exist in the printer inventory.
- ▶ To define environment variables used by the Printer Inventory Manager, define environment variables in the data set allocated to the STDENV DD statement. If you have Infoprint Server files installed in the */usr/lpp/Printsrv/* directory, you do not need to specify the LIBPATH, NLSPATH, and PATH environment variables.
- ▶ If you do not have the Language Environment run-time libraries (CEE.SCEERUN) and the C++ run-time library (CBC.SCLBDLL) in the system LNKLIST, add a STEPLIB DD statement for these libraries.

AOPP procedure JCL example

Figure 2-22 shows a modified JCL procedure that can be invoked from a batch job or as a started task. The JCL assumes that Infoprint Server files are installed in the */usr/lpp/Printsrv/* directory. On the PROC statement the options are:

- ▶ PRT= assigns the default printer name
- ▶ OPT= sets the **lp** command options
- ▶ PD= (PATH/DATA set disposition) set to string DISP=SHR

```

/*-----
/* AOPP - z/OS Print Server batch print procedure
/*-----
//AOPP      PROC PRT='IAZFSS',OPT=,OCL='*',ECL='*',PD='DISP=SHR'
//IEFPROC   EXEC PGM=AOPBATCH,
//  PARM='/lp -d &PRT -o "&OPT" //DD:IEFRDER'
//STDOUT    DD  SYSOUT=&OCL
//STDERR    DD  SYSOUT=&ECL
//IEFRDER   DD   &PD

```

Figure 2-22 Modified AOPPRINT JCL procedure in SYS1.PROCLIB

AOPP procedure usage examples

To print an MVS data set (for example the AOPP JCL procedure) to the default printer, enter the operator command:

```
S AOPP,DSN=SYS1.PROCLIB(AOPP)
```

If you want to print an HFS file (for example /etc/profile) to the default printer, enter the operator command:

```
S AOPP,PATH='/etc/profile',PD=
```

To issue the same print request from JCL:

```

//VAINIPR JOB (999,POK),EXPERT,MSGLEVEL=1,MSGCLASS=T,NOTIFY=&SYSUID
//          EXEC AOPP,PD=
//IEFRDER DD PATH='/etc/profile'

```

Figure 2-23 AOPPRINT JCL procedure batch job

To print an MVS data set to a PostScript printer, enter the following operator command:

```
s aopp,prt='pokeps',dsn=sys1.proclib(aopp),opt='document-format=line'
```

The *pokeps* printer definition in the Printer Inventory includes the following filter definition for **line** data:

```

# display printer pokeps
printer-codepage = IS08859-1
filters = {
    line -> "afp2ps.dll %filter-options"
}

```

To issue the same print request from JCL:

```

//VAINIPR JOB (999,POK),EXPERT,MSGLEVEL=1,MSGCLASS=T,NOTIFY=&SYSUID
// EXEC AOPP,PRT='pokeps',OPT='document-format=line duplex=yes'
//IEFRDER DD DSN=SYS1.PROCLIB(AOPP)

```

Figure 2-24 JCL using AOPPRINT JCL procedure

2.12.3 End-user (lpr) considerations

To use the Print Interface from workstations, an administrator needs to provide information to the end user to allow printing of files to the host.

Line printer requester (LPR) is a client that allows the local host user using the **lpr** command to submit a print request via TCP/IP for printing on a remote printer server, which is the Print Interface LPD with Infoprint Server.

A Windows user can also use the OS/390 Printer Port Monitor for Windows, which uses the LPR protocol to submit print requests.

A workstation user needs to know the following information to print to an Infoprint Server defined printer:

- The IP address or host name of the z/OS system where the Print Interface LPD is running.
- The name of the printer definition for the target printer defined in the Printer Inventory.

Attention: This printer name in the printer definition is case sensitive. The end user must be sensitive to this.

In addition to the IP address and the printer name, a user using the OS/390 Printer Port Monitor also needs to know the following:

- The printer driver to use for the target printer

For printing to IBM AFP printers, Windows users can download the AFP Printer Driver for Windows. If you have specified either the PCL to AFP transform or the PostScript to AFP transform in the printer definition, users can use a PostScript or PCL printer driver because the transform converts the data to AFP format on the z/OS system.

- The port number 515 on which the Print Interface LPD is listening, or if the well-known port 515 is not used, the port number being used

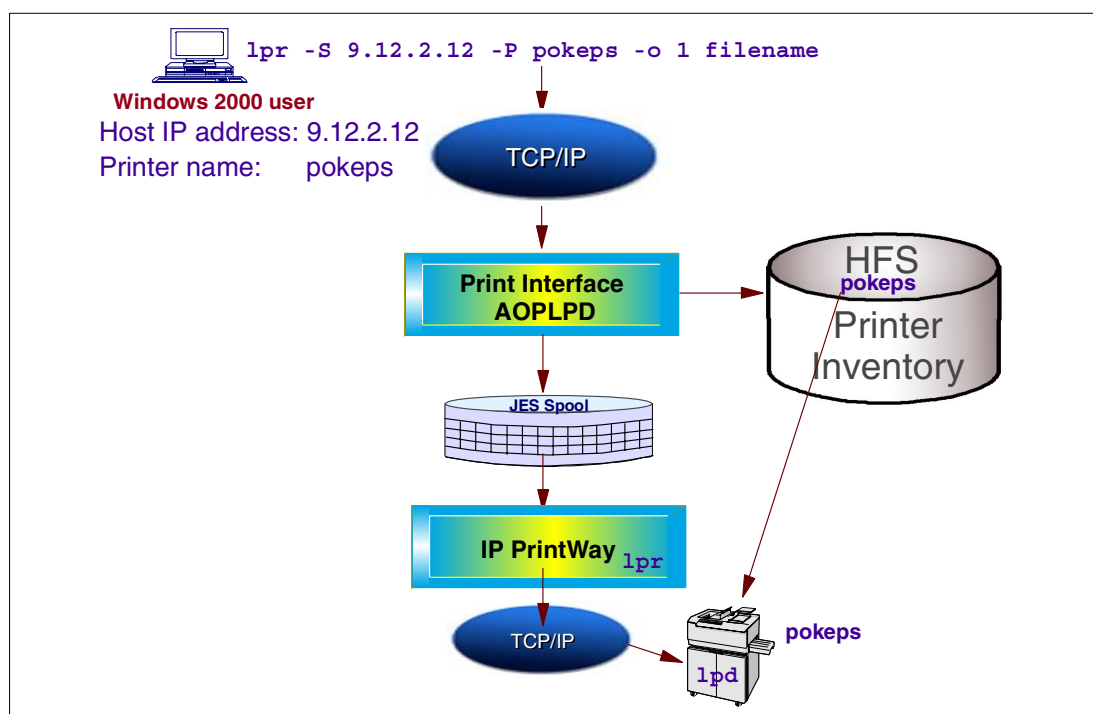


Figure 2-25 A Windows user using the lpr command

Note: For customization of the LPR to LPD from IP PrintWay, see “LPR to LPD considerations” on page 114.

For additional information about how to submit jobs from remote systems and how to configure the OS/390 Printer Port Monitor for Windows, see “Infoprint Port Monitor” on page 138. Also, refer to *z/OS Infoprint Server User's Guide*, S544-5746.

2.12.4 Print Interface LPR/LPD considerations

Some LPRs let you specify whether the LPR should send the LPD control file first, before the data, or last after the data file. The Print Interface LPD can receive the LPD control file either first or last. However, IBM recommends that the LPR send the control file first because the Print Interface LPD uses less storage and can process the data file more quickly.

Sending large files

When sending large files to the Print Interface LPD, especially when the printer definition specifies a data transform, users on an AIX system might need to change the time-out value used by the AIX system. The time-out value is the amount of time the AIX system waits for an acknowledgement from the LPD that transmission was successful. Because the Print Interface LPD returns an acknowledgement only after data is transformed to a format accepted by the printer and written successfully to the JES spool, the LPD might send an acknowledgement only after several minutes. If the document contains several gigabytes of data that need to be transformed, the LPD might send an acknowledgement only after an hour or more.

LPD compatibility filter

The LPD compatibility filter, **aopfiltr.so**, provides support for some LPR command codes and parameters that the Print Interface LPD does not otherwise support. LPD command codes and parameters are specified in the LPD control file sent by the LPR with each document to be printed.

The LPD compatibility filter provides support that is similar to that provided by the z/OS Communications Server (TCP/IP) LPD. For example, with the LPD compatibility filter, the Print Interface LPD supports the **p** command code, which lets you print a header with a page number on each page. The LPD compatibility filter also lets job submitters and the administrator specify the **-f**, **-l**, and **-w** filter options, which correspond to the **FILTER**, **LINECOUNT**, and **WIDTH** parameters of the TCP/IP LPR command.

Refer to *z/OS Infoprint Server Operation and Administration*, S544-5745 and *z/OS Infoprint Server User's Guide*, S544-5746 for more information.

2.13 AOPBATCH utility program

The AOPBATCH utility program is provided by Infoprint Server. AOPBATCH lets you use MVS JCL to run a program that resides in an HFS file. IBM recommends that you use AOPBATCH instead of BPXBATCH to run programs provided by Infoprint Server because AOPBATCH sets default values for the **PATH**, **LIBPATH**, and **NLSPATH** environment variables that are suitable for installations that installed Infoprint Server files in default locations. Also, AOPBATCH lets *stdin* be read from a DD statement and lets *stdout* and *stderr* be written to a DD statement.

The program AOPBATCH uses the **spawn()** function to create a process that runs in the same address space as AOPBATCH under these conditions:

- ▶ The set-group-ID bits are not set for the program.
- ▶ Enough storage exists to spawn a process in the same address space.
- ▶ The share address space extended attribute is set for the HFS program object. This attribute is set by default.

With AOPBATCH you can specify the optional DD statements to set up the z/OS UNIX environment and to allocate the **STDERR**, **STDIN**, and **STDOUT** files.

Note: The STDIN, STDOUT, and STDERR statements are suitable only for text data. Do not use STDIN, STDOUT, and STDERR to specify an MVS data set that contains non-text data. For example, do not specify an MVS data set in STDOUT when you run a transform command because the output (AFP, PostScript, PDF, or PCL data) is not text data.

2.13.1 Using AOPBATCH

Figure 2-26 shows how to use the AOPBATCH program. The example JCL invokes the Infoprint Server pidu program.

```
//AOPBATCH JOB .....  
//PIDU      EXEC PGM=AOPBATCH,PARM='pidu'  
//STDIN     DD *  
list printer;  
/*  
//STDOUT    DD  SYSOUT=*  
//STDERR    DD  SYSOUT=*  
//STDENV    DD  *  
PATH=/usr/lpp/Printsrv/bin:/bin:/usr/bin  
LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib  
NLSPATH=/usr/lpp/Printsrv/%L/%N:/usr/lpp/nls/msg/%L/%N  
/*
```

Figure 2-26 AOPBATCH program JCL example

Where:

EXEC PGM=AOPBATCH,PARM='program_name options'

PARM= Specifies a program name and options

program_name - Names the executable program file that resides in an HFS. The program name is case sensitive. The file must reside in one of the directories specified in the PATH environment variable or you must specify the pathname. Use the STDENV DD statement to set the PATH environment variable if the default value set by AOPBATCH is not suitable.

Programs provided with Infoprint Server, which you can run using AOPBATCH are:

pidu - The Printer Inventory Definition Utility program. See “Running the pidu command as a batch job” on page 223.

aopmig - The Printer Inventory migration program.

If the program name contains a slash, precede program_name with a slash. For example, if the program_name is /mylib/myprogram, specify:

PARM='//mylib/myprogram'.

Note: You cannot use AOPBATCH to run a shell command or a shell script directly. To run a shell command or script, specify sh as the program name, for example:

PARM='sh -c myscript' or PARM='/sh -c /mylib/myscript.

AOPBATCH returns the exit code of the spawned process. If AOPBATCH cannot execute the program, the return code is 4.

options - Options accepted by program *program_name*. For a description of the allowed options, refer to the documentation for the program. For Infoprint Server programs, refer to one of the following publications: *z/OS Infoprint Server User's Guide*, S544-5746 or *z/OS Infoprint Server Operation and Administration*, S544-5745.

STDENV Specifies environment variables for use by program *program_name*. You can specify the environment variables in-stream in the JCL, in an HFS file, or in an MVS data set. Specify the environment variables in the format *variable = value*, with only one environment variable on each line or record.

If you omit the STDENV DD statement or do not specify one of the following environment variables, AOPBATCH sets the following default values, which are suitable for running Infoprint Server programs if your installation installed Infoprint Server files in the default directories:

PATH=/usr/lpp/Printsrv/bin:/bin:/usr/bin

LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib

NLSPATH=/usr/lpp/Printsrv/%L/%N:/usr/lib/nls/msg/%L/%N

AOPBATCH also sets the HOME environment variable to the user's home directory and sets the LOGIN variable to the user ID.

STDERR Specifies a SYSOUT data set, an HFS file, or an MVS data set.

STDIN Specifies input to the program. You can specify input in-stream in the JCL, in an HFS file, or in an MVS data set.

STDOUT Specifies a SYSOUT data set, an HFS file, or an MVS data set.

Note: AOPBATCH is installed in SYS1.LINKLIB; therefore, you do not need to include a STEPLIB DD statement in your JCL.

2.14 Starting Infoprint Server

Now that all the system options are configured and ready for users to submit print requests, let's review all the possible start options. The **aopstart** command has been issued in one of the following ways:

- ▶ **S AOPSTART** from an operator console
- ▶ A batch job submitted using AOPBATCH
- ▶ **aopstart** from the OMVS command line

In the aopd.conf configuration file, to start the daemons, specify:

```
start-daemons = { lpd xfd ippd }
```

Note: The AOPD daemon is always started.

You have also customized the following configuration files:

```
aopsapd.conf - aopxfd.conf
```

In the following examples, all functions of Infoprint Server are activated. Issuing a **D A,L** command shows the following:

AOPD	STEP1	AOPSS	NSW	AO	AOPLPD	STEP1	AOPSS	OWT	AO
AOPXFD	STEP1	AOPSS	OWT	AO	AFP2PCLD	STEP1	AOPSS	OWT	AO
AFP2PDFD	STEP1	AOPSS	OWT	AO	AFP2PSD	STEP1	AOPSS	OWT	AO
PCL2AFPD	STEP1	AOPSS	OWT	AO	PS2AFPD	STEP1	AOPSS	OWT	AO
AOIPPD	STEP1	AOPSS	OWT	AO	AOPSAPD	STEP1	REDADM	OWT	AO
PRINTW43	PRINTWAY	IEFPROC	NSW	S	WTRIAZF	WTRIAZF	IAZSFSS	NSW	S

Figure 2-27 Infoprint Server address spaces

AOPD	Printer Inventory Manager daemon
AOPLPD	Printer Interface LPD daemon
AOPXFD	Infoprint Server Transform Manager daemon
AFP2PCLD	AFP to PCL transform daemon
AFP2PDFD	AFP to PDF transform daemon
AFP2PSD	AFP to PostScript transform daemon
PCL2AFPD	PCL to AFP transform daemon
AOIPPD	Print Interface Internet Printing Protocol (IPP) daemon
AOPSAPD	SAP R/3 daemon
PS2AFPD	PostScript and PDF to AFP transform daemon
PRINTW43	IP PrintWay LPR to LPD address space
WTRIAZF	Sample FSS



Defining printers in the Printer Inventory

This chapter describes how to define printers in the Printer Inventory. The Printer Inventory is a set of files maintained by the system administrator. It contains information about the z/OS printers, including both local and remote printers. The Printer Inventory is common to the Print Interface, IP PrintWay, and NetSpool, which are components of Infoprint Server. This means that the system administrator can define a printer once for all components of Infoprint Server.

This chapter describes the following:

- ▶ Printer Inventory overview
- ▶ Printer Inventory ISPF application
- ▶ Adding printer definitions for:
 - IP PrintWay printers
 - PSF printers
 - General printers
- ▶ Displaying defined printers
- ▶ Sending print data sets to the Print Interface
 - Viewing data sets on the JES spool

3.1 Printer Inventory

For users to send data to the JES spool, they must be aware of the printer destinations that are defined in the Printer Inventory in the HFS using one of the following:

- ▶ Infoprint Server ISPF panels
- ▶ Printer Inventory Definition Utility (PIDU)
- ▶ Infoprint Server Migration program

There are many ways for a user to submit a data set for printing on a printer defined in the Print Inventory. In Figure 3-1, a Windows workstation user uses the `lpr` command to send a file to be printed on a printer that is defined in the Printer Inventory.

The user is required to know the following:

- ▶ The IP address of the host z/OS system
- ▶ The names of the defined printers in the Print Inventory that the user is allowed to access, that process the type of file format the user wants to print.

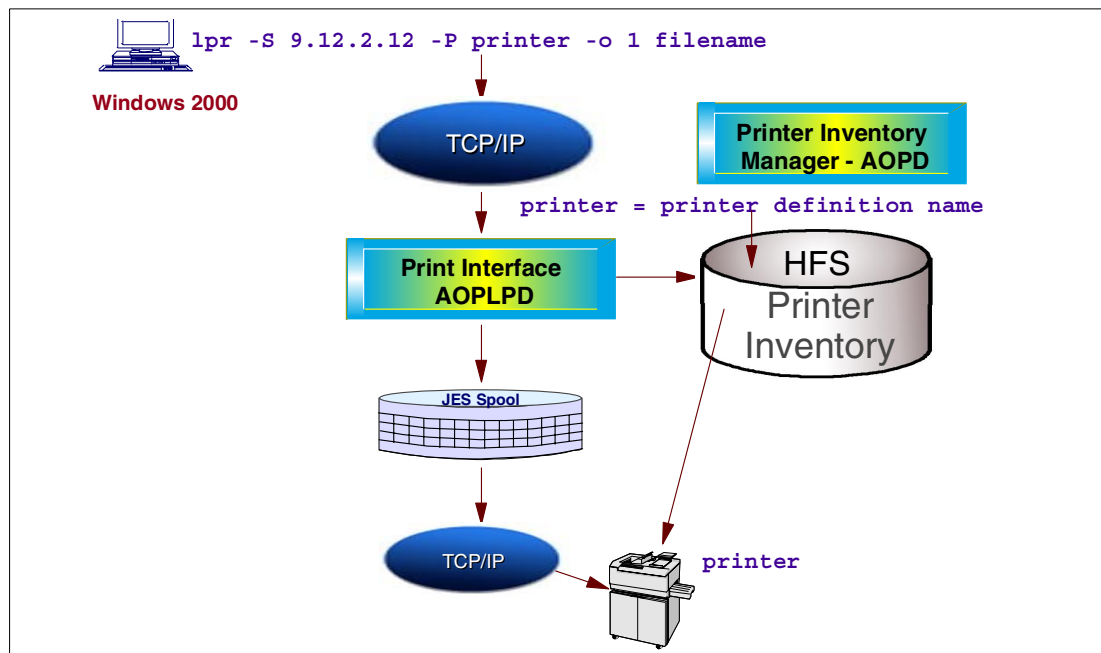


Figure 3-1 User sends a file for printing using Infoprint Server

Infoprint Server provides an ISPF application with ISPF panels to manage entries in the Printer Inventory. You can add, list, browse, copy, edit, and delete entries in the Print Inventory. The type of entries that you can manage are:

- ▶ Printer definitions
- ▶ Components
- ▶ Printer pool definitions
- ▶ FSA definitions
- ▶ FSS definitions

The Printer Inventory Manager daemon, AOPD, must be active to work with entries in the Print Inventory.

You can also use the Printer Inventory Definition Utility (PIDU) to manage entries in the Printer Inventory. The PIDU program can be useful for creating a large number of entries in the Printer Inventory. Also, you can use the PIDU program to create a backup copy of the Printer Inventory; see “Printer Inventory Definition Utility” on page 213.

3.2 Printer Inventory ISPF application

To define a printer to process output to be sent to the host from a workstation for printing on a printer in the IP network via IP PrintWay, you need to use the ISPF application using TSO/E. The Primary Menu is shown in Figure 3-2.

Important: Once you enter any of the panels, to determine what to specify on each line, place the cursor on the line and hit PF1 and you will receive the help screen for that entry.

```

A0PIPP1                      Infoprint Server: Printer Inventory Manager
Option ===> _____

Printer Definitions
  1 Add                      Add a printer definition
  2 List                     List printer definitions
  3 Select                   Select printer definitions to list

Other Functions
  4 FSS/FSA/Pool            Manage other inventory definitions
  5 PrintWay Queue          View IP PrintWay transmission queue
  6 PrintWay Message        View IP PrintWay message log
  7 Configure               Change panel configuration

```

Figure 3-2 ISPF application Primary Menu

3.2.1 Printer definitions

From the Primary Menu, the three options for Printer Definitions are:

- Add** Select **Option 1** when you want to add a new printer definition. The panel that is then displayed, shown in Figure 3-4 on page 64, lets you select the type of printer definition. If the type is IP PrintWay, you must also select the protocol that you want IP PrintWay to use to transmit data to the remote printer. To select a type and protocol, type a selection number in the Option field and press Enter.
- List** Select **Option 2** when you want to list the printers that have been previously defined. The panel that is displayed lists all printer definitions. The printer definitions are listed by name, type, description, location, DEST, CLASS, FORMS, and LU name. Depending on how many printer definitions are listed, you might need to scroll through the list.
- Select** Select **Option 3** when you want to set the criteria for the printers you want to display. The more criteria you enter, the more selective the list is because printer definitions are only listed if they match all the criteria you specify. If you do not enter any criteria, all printer definitions are displayed.

3.2.2 Other functions

The Other functions, shown in Figure 3-2 on page 61, are briefly described in this section and in other sections of this redbook. Option 7 should be configured first since it affects some of the other options (see “Configuration options - Option 7” on page 62).

FSS/FSA/Pool - Option 4

When you select Option 4, you receive a panel that lets you add, list, or select various types of definitions within the Printer Inventory such as:

- ▶ Functional subsystem application (FSA)
FSA definitions provide information for working with a printer or group of printers that are not related to a specific job. These definitions include trace, default form definition, and PSF separator page information.
- ▶ Functional subsystem (FSS)
FSS definitions manage the address space in which FSAs run.
- ▶ Printer definition pool
Printer definition pool definitions let a user select a group of printers to print to. The job is then sent to all the printers included in the pool.

Note: See “FSS definitions” on page 78.

IP PrintWay Transmission queue - Option 5

This is a data set for IP PrintWay. When a data set is selected to be printed, IP PrintWay creates an entry in the transmission-queue data set.

The IP PrintWay transmission-queue data set is a VSAM key-sequenced data set. You can define it with any name. A sample job is provided in SYS1.SAMPLIB(ANFDEAL), with a default data set name of ANF.QUEUE.

After allocating space for the transmission-queue data set, use the sample JCL in SYS1.SAMPLIB(ANFQINIT) to initialize the data set with binary zeroes. If you did not use the default name, ANF.QUEUE, for the data set, change the name of the data set in the sample JCL.

Note: See “Allocate and initialize the transmission queue data set” on page 103.

IP PrintWay Message log - Option 6

IP PrintWay writes messages that track data-set transmissions in an IP PrintWay message-log data set. This is a required data set.

You can create one message-log data set to be shared by all IP PrintWay functional subsystems (FSSs); or you can create separate data sets for each FSS. To allocate space for the data set and initialize it, use the sample JCL in SYS1.SAMPLIB(ANFMIJCL) to run the ANFMFILE program.

See “Printing to IP PrintWay printers with JCL” on page 115.

Configuration options - Option 7

When you select this Option, the displayed panel, shown in Figure 3-3 on page 63, lets you select your configuration options. This panel contains the IBM default values when it is first displayed for the following panel options:

- ▶ You can select to confirm deletion requests for printer definitions
- ▶ ISPF trace directory name of an existing HFS directory to which trace entries are written when the Trace ISPF Internals field is selected.

- ▶ Printer Inventory configuration file directory and file name location of an existing Infoprint Server for the configuration file.
- ▶ NLS, and language path directory and file name location of an existing National Language Support (NLS) file, and on the next line a value that represents the language used for the Printer Inventory.
- ▶ A default printer name that you choose from your list of defined printers when the PRINTER and LPDEST environment variables are not set.
- ▶ An IP PrintWay message log name of the sequential data set, which is initialized during the IPL
- ▶ The PrintWay configuration that IP PrintWay uses to log messages during an ISPF session.
- ▶ The trace data set is the name of the sequential data set to which IP PrintWay writes trace records when tracing an ISPF session. If the data set does not exist, IP PrintWay allocates and catalogs the data set. If the data set already exists, IP PrintWay appends new trace statements to it.

AOPIPCF Configuration	
Command ==> _____	
/ Confirm delete requests	
_ Trace Printer Inventory ISPF internals	
ISPF trace directory. . /var/Printsrv/trace _____	
Printer Inventory:	
Configuration file .	/etc/Printsrv/aopd.conf _____
NLS path	/usr/lpp/Printsrv/%L/%N _____
Language	En US _____
Default printer. . .	FIJVBIN _____
IP PrintWay:	
Message log.	INFOPRNT.SC43.MSGFILE _____
Trace data set . . .	ROGERS.ANFISPF.TRACE _____

Figure 3-3 Configuration panel to customize Infoprint Server options

If you do not use the default in your definitions, you must customize this panel for your environment.

3.3 Add Printer definitions

When you select **Option 1** on the Primary Menu panel to add a printer to the Inventory, the panel shown in Figure 3-4 on page 64 is displayed.

There are three categories of printers that can be defined:

- | | |
|-----------------------|---|
| IP PrintWay | IP PrintWay supports four types of protocol to the devices it supports. |
| PSF for OS/390 | This printer type is for IBM AFP printers controlled by PSF for OS/390 or a comparable product. AFP printers can be locally-attached, SNA-attached, or TCP/IP-attached. |
| General | This is a printer type that is not like the other two categories, such as a line printer controlled by JES. |

```

A0PIPP1          Infoprint Server: Printer Inventory Manager
0                Choose a Definition Type and Protocol
P
  A0PIPDAP =>
  1 IP PrintWay, LPR protocol
  2 IP PrintWay, direct sockets protocol
  3 IP PrintWay, IPP protocol
  4 IP PrintWay, VTAM protocol
  5 PSF for OS/390
  6 General definition type
0

```

Figure 3-4 Add a printer definition panel

Note: This panel is new beginning with OS/390 Release 9 and z/OS Release 1. In previous versions, there were only three options (1, 5, and 6). The old Option 1 allowed specification of LPR, IPP, and direct socket support in the protocol custom values selection panel. The VTAM protocol support is new with PTFs UW73909 and UW73913.

3.3.1 Printer definition considerations

When defining printers in the Printer Inventory for users to print to, you must create at least one printer definition to represent each actual printer.

For some printers, to satisfy some printing requirements, you might need to set up different configurations of printer definitions for the same printer. In this example, one definition or destination for the end user allows different data streams to be passed to the same destination because the actual device can print any of the supported data streams, or transforms could convert the data stream to one that is supported by the device.

Another example is that you want to create one printer definition to represent several actual printers. In this case, users print to a destination defined in the Printer Inventory and then there is a pool of printers that all select the characteristics assigned to the output.

Print attributes

The characteristics of a printer and print jobs that you specify in a printer definition are called attributes. Using the ISPF panels, you specify attributes as values in the panel fields. Using the Printer Inventory Definition Utility (PIDU), you specify attributes as keyword and value pairs.

Not all attributes in a printer definition need to be specified because Infoprint Server and JES supply default values for many of the attributes. Some attributes are used by all components of Infoprint Server; however, other attributes are used by only one component. Therefore, depending on which components of Infoprint Server you plan to use in your installation, you might not need to specify all of the attributes. For example, if you do not plan to use NetSpool to print VTAM application data, then you do not need to specify attributes that only NetSpool uses.

The online help for each panel field describes which components use the field.

Most of the attributes that you can specify in a printer definition are divided into logical groups called sections. The six sections of a printer definition are:

- Allocation

- ▶ Processing
- ▶ NetSpool Options
- ▶ NetSpool End-of-File
- ▶ IP PrintWay Options
- ▶ Protocol

3.4 Defining IP PrintWay printers

IP PrintWay supports four types of protocols to print data to IP PrintWay defined printers in the Printer Inventory. These protocols are shown in Figure 3-5 on page 66.

- ▶ LPR protocol

IP PrintWay can use the LPR protocol to transmit data sets to the target printer. The LPR protocol is a TCP/IP protocol defined by RFC 1179 and is described briefly in “LPR to LPD protocol” on page 2. To use this protocol, an LPD that adheres to RFC 1179 must be running in the remote printer or print server.

- ▶ Direct sockets protocol

IP PrintWay can use the direct sockets printing protocol to transmit data sets to the target printer. Direct sockets printing protocol is a TCP/IP protocol in which IP PrintWay transmits data directly to any designated port, port 8234 as shown in Figure 3-5 on page 66, on the target system. To use this protocol, the remote printer or print server must support direct sockets printing.

- ▶ IPP protocol

IP PrintWay can use the Internet Printing Protocol (IPP) to transmit data sets to the target printer. To use the IPP protocol, an IPP server must be running in the remote printer or host system.

- ▶ VTAM protocol

IP PrintWay can use the VTAM protocol to transmit data sets to a VTAM-controlled printer. The printer must accept either the SNA Character String (SCS) or the Data Stream Compatible/Data Stream Extended (DSC/DSE) data stream and must be defined to VTAM as LU type 0, 1, or 3.

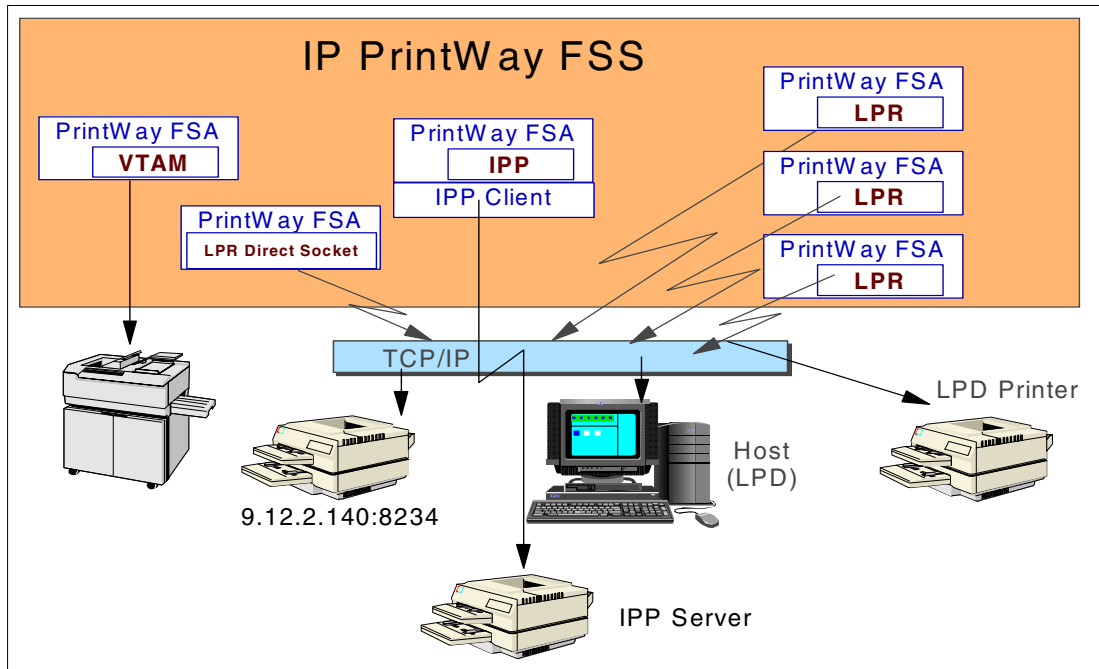


Figure 3-5 IP PrintWay supports use of 4 protocols to printers

Attention: Although there are four different protocols supported for IP PrintWay printers, defining the Printer Inventory entries is done using the same ISPF panels. There is one additional step necessary for the VTAM protocol printers; see “IP PrintWay printers with VTAM protocol” on page 77.

3.4.1 Defining IP PrintWay printers with LPR protocol

This is Option 1 from Figure 3-4 on page 64. The panel that is displayed, shown in Figure 3-6 on page 67, is the same for Options 1 to 4 for IP PrintWay printers.

There are several ways to define printers, depending on the installation and the number of printers involved. The following are common to all types of printers that are definable:

- ▶ Printer definition name, description, and location
- ▶ Section
- ▶ Component name
- ▶ Custom values

AOPIPDC1 IP PrintWay Printer Definition		
Command ==> _____		
Printer definition name . <u>pokeps</u>		
Description . <u>3130 Postscript printer</u>		(extend)
Location. . . <u>Room 2c-16 Bldg 8</u>		(extend)
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> *
Processing	=> _____	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> *
Protocol	=> _____	=> *
/ Use DEST, CLASS, and FORMS for IP PrintWay printer selection		
NetSpool LU name . _____ LU classes . . _____		(extend)

Figure 3-6 Defining IP PrintWay printers

In this section we describe the above options for all the printer types.

Printer definition name, description, and location

Printer name The name assigned to the printer definition. It must be a unique printer definition name in the Printer Inventory. A job submitter uses this name to select the printer.

Attention: The printer name assigned in the printer definition is case sensitive.

Description A description of the printer definition. This field is not required; however, the description can help users select a printer.

Location A description of where the printer described by this printer definition is located. This field is not required; however, the location can help users select a printer.

Section

The printer definition is divided into sections (such as Allocation and Processing). In each section, you can specify different types of printer attributes by choosing to select a component name for fixed attributes and then using custom values as follows:

Component name For each section, you can create objects in the Printer Inventory called components (such as Allocation components and Processing components). Consider creating components when several printer definitions share the same printer attributes. Specify the shared attributes in the component and specify the name of the component in each printer definition. When you change values in a component, all printer definitions that name the component use the new values. To specify printer attributes in a component:

Move the cursor to the Component name field for a section; press Enter.

On the next panel, select an existing component or create a new component and select it. Return to the Printer Definition panel. See “Defining components” on page 87.

Custom values Instead of using components, or in addition to components, you can specify printer attributes as custom values. Custom values apply only to this printer definition. Custom values override the same values in a

component. To enter the definitions for the Custom values, shown in Figure 3-6 on page 67, place the cursor on each line to display the appropriate panel for each section and press Enter.

DEST, CLASS, and FORMS

This is important in terms of users submitting via JCL to IP PrintWay printers. So, place a slash (/) to indicate that a job submitter can use the DEST, CLASS, and FORMS parameters to select this printer definition. Obviously, a blank indicates that DEST, CLASS, and FORMS cannot be used.

Note: Whether or not this indicator is set, a job submitter can specify a printer name on an OUTPUT JCL statement to select this printer definition (FSSDATA='printer=name' parameter).

If you set the indicator, specify at least one DEST, CLASS, or FORMS value in the Allocation section; see “Allocation section” on page 68. If one of these values is blank, the corresponding JCL parameter is not used for printer selection. The values in the DEST, CLASS, and FORMS fields must, together, be unique in all printer definitions with this indicator set.

Allocation section

By placing the cursor on the Allocation line under the Custom values and pressing Enter, Figure 3-7 on page 69 is displayed.

Note: When you see “More: +” on any panel screen, it means the panel is larger than the display screen. Use PF8 to scan down further for the additional fields.

The most important fields on this panel are:

- CLASS** Specify an output data set class that is used by an IP PrintWay FSS in the writer selection parameters. If the field is left blank, the JES default class is used.
- DEST** Specify the destination to be the same as the printer name.
- FORMS** Specify a default FORMS value for your installation.

Important: Many fields on this panel become part of the data set output characteristics for the processing of the data set when it finally resides on the JES spool.

You must specify at least one of these options, a DEST, CLASS, or FORMS value.

Important: The combination of DEST, CLASS, and FORMS must be *unique* in all printer definitions, which means that no other printer can have the same exact combination of these three fields.

If a value in the DEST, CLASS, or FORMS field is blank and a user submits a job using JCL to print the output via IP PrintWay, the value in the JCL parameter is not used for the printer selection.

The fields that create the data set characteristics for the data sets that are to be printed are in the following sections:

- ▶ Spool allocation values
- ▶ Values for Separator Pages
- ▶ Resource-related attributes
- ▶ Error-reporting attributes

These data set characteristics are passed to an arriving data set by the Print Interface. See “Print Interface receives data sets” on page 134.

AOIPDAL		Allocation	
Command ==> _____			
Printer definition name . <u>pokeps</u> _____			
			More: +
Spool allocation values:			
CLASS	<u>J</u>	LINECT.	_____
DEST.	<u>POKEPS</u>	PRMODE.	_____
JES node.	_____	PRTY.	_____
FCB	_____	SEGMENT	_____
FLASH count	_____	THRESHLD.	_____
FLASH name.	_____	UCS	_____
FORMS	<u>STD</u>	WRITER.	_____
GROUPID	_____		
USERDATA	_____		
			(extend)
BURST	<u>2</u>	1. Yes 2. No	
HOLD.	<u>2</u>	1. Yes 2. No	
OUTDISP	<u>1</u>	1. Purge 2. Leave 3. Keep 4. Hold 5. Write	
Values for Separator Pages:			
Address _____ (extend)			
Building	_____		
Department	_____		
Name	_____		
Room	_____		
Title	_____		
Resource Related Values:			
Form definition	_____		
Character sets	_____		
Overlay front	_____	Back	_____

Figure 3-7 Allocation options panel - (Part 1)

AOIPDAL		Allocation	
Command ==> _____			
Printer definition name . <u>pokeps</u> _____			
			More: - +
Overlay front	_____	Back	_____
Input tray	_____		
Output bin	_____		
Page definition	_____		
Resource library.	_____ (extend)		
Image shift x-direction front	_____	Back	_____
y-direction front	_____	Back	_____
Error Reporting Values:			
Print error reporting.	<u>1</u>	1. None 2. All 3. Character 4. Position	
Error disposition.	<u>1</u>	1. Default 2. Hold 3. Quit	
_ Print error messages			
Maximum messages.	_____		
Other Values:			
Notify	_____	at node	_____
	_____	at node	_____
	_____	at node	_____
	_____	at node	_____
Checkpoint pages	_____		
Checkpoint seconds	_____		
Copies	_____		
Copy group	_____	_____	_____
Color map.	_____		
Com setup member	_____		
JES form length.	_____		
Resolution	_____		
Duplex	<u>2</u>	1. Simplex 2. Duplex 3. Tumble	

Figure 3-8 Allocation options panel - (Part 2)

AOPIPDAL		Allocation	
Command ==> _____			
Printer definition name . <u>pokeps</u> _____			
Resource library. _____			More: - (extend)
Image shift x-direction front . . _____		Back . . _____	
y-direction front . . _____		Back . . _____	
Error Reporting Values:			
Print error reporting. . . _		1. None 2. All 3. Character 4. Position	
Error disposition. . . . _		1. Default 2. Hold 3. Quit	
_ Print error messages			
Maximum messages. . . _			
Other Values:			
Notify	_____	at node . .	_____
	_____	at node . .	_____
	_____	at node . .	_____
	_____	at node . .	_____
Checkpoint pages . .	_____		
Checkpoint seconds .	_____		
Copies	_____		
Copy group	_____		
Color map.	_____		
Com setup member . .	_____		
JES form length. . .	_____		
Resolution	_____		
Duplex	2	1. Simplex 2. Duplex 3. Tumble	
Label data pages . .	1	1. Yes 2. No	
Restrict printable area . .	1	1. Yes 2. No	
_ Table reference characters			

Figure 3-9 Allocation options panel - (Part 3)

Processing section

The Processing section panel, shown in Figure 3-10 on page 71, lets you view and customize processing and validation attributes for a printer definition. You see this panel when you press Enter from the Custom Values field on the Printer Definition panel, shown in Figure 3-6 on page 67.

The values specified on the panel are used by Print Interface, NetSpool, and IP PrintWay to process data and validate that a print job can print on this printer.

When print data arrives to the Print Interface AOPLPD daemon, this panel definition, for the specified printer, is used to validate the print data stream. This is done to make sure the print stream can print on the printer specified by the user.

Data formats

When defining the valid data streams that are supported for this printer, input a “/” next to the data format. The Print Interface supported data streams that are allowed to be received on a defined printer are shown in Figure 3-10 on page 71.

Note: Make sure that you select data formats for a printer that is being defined that the printer itself can print.

The data formats allowed for Infoprint Server-supported printers are as follows:

Line data	Character data; can have carriage controls and TRCs
MO:DCA-P	IBM Mixed Object Document Content Architecture Presentation
PostScript	Adobe PostScript
Text	Character data; can have LF, CR, HT, VT, FF controls
PCL	Hewlett Packard Printer Control Language

PDF	Adobe Portable Document Format
SAP	SAP output text format (OTF) or ABAP format
Other	All data formats that are not one of the above types

Note: The print data stream is rejected if the data format received is not selected by a non-blank in the Print Interface-supported data formats in Figure 3-10.

AOPIPDRR

Processing

Command ==>

Printer definition name .

pokeps

Document code page . .

Printer code page. . .

ISO8859-1

More: +

Print Interface supported data formats and associated filters:

Data format: Filter:

/ Line data

_____ (extend)

- M0:DCA-P

_____ (extend)

/ PostScript

_____ (extend)

/ Text

aopfiltr.so _____ (extend)

/ PCL

_____ (extend)

- PDF

_____ (extend)

- SAP

_____ (extend)

- Other

_____ (extend)

- Resubmit for filtering

Conversion between SCS and Line:

Margins: Top . . .

Bottom . . .

Left . . .

Right . . .

Line length

Page length . . .

Tabs: Vertical . . .

(extend)

Horizontal . . .

(extend)

IP PrintWay Line-to-Text Conversion:

/ Pagination

Margins: Top . . .

Bottom . . .

Page height . . .

58

Figure 3-10 Processing section panel (Part 1)

The Processing section panel is also used to define the use of Infoprint Server Transforms. See “Transform classes” on page 181.

Document code page

The code page used to create documents submitted to this printer definition is used by Print Interface and IP PrintWay as the source when converting documents from ASCII to EBCDIC or from EBCDIC to ASCII. This field is not required because a default code page is used.

Note: For code page defaults, see Figure 2-9 on page 33 and the description of the code page defaults.

In most cases, the default is suitable. Print Interface uses either the default ASCII code page or the code page for the locale. However, to print ASCII documents submitted with the **1p** command, specify an ASCII code page either in this field or on the **1p** command.

Chapter 3. Defining printers in the Printer Inventory

71

Attention: Leave either or both code page fields blank if you want IP PrintWay to use the standard TCP/IP translate table.

Printer code page

Specify an ASCII code page if the printer is a PCL printer or a PostScript printer. Specify an EBCDIC code page if the printer is an IBM AFP printer or a VTAM-controlled printer.

If this field is blank, IP PrintWay uses code page IBM-850 as the default; for the VTAM protocol, IP PrintWay does not convert data.

Data format: Filter:

A filter is a program that can inspect and modify data before Print Interface writes the data to an output data set on the JES spool. For each type of data format that Print Interface supports, you can specify the name of an associated filter.

When you specify the name of a filter for a supported data format in a printer definition, Print Interface automatically calls that filter before writing the data set to the JES spool.

Infoprint Server provides a filter, `aopfiltr.so`, which prepares text data for printing on ASCII printers. This filter converts ASCII line-feed controls that are not preceded by carriage-return controls to carriage-return and line-feed controls (X'0D0A'). The X'0D0A' control is suitable for most ASCII printers.

Note: IBM recommends that you associate this filter with the text data format in the printer definitions for all ASCII printers controlled by IP PrintWay.

This is where you specify a filter name or data stream transform. A data stream transform can transform data to a new data format before Print Interface writes the data to the JES spool. If this field is blank, no filter is used.

Infoprint Server Transforms provides the following filters for printing line data formatted with an AFP page definition on an ASCII printer:

afp2pcl.dll Transforms line data to PCL 5, 5e, or 5c (color)

afp2ps.dll Transforms line data to PostScript level 2

afp2pdf.dll Transforms line data to PDF 1.2

The filter format is a filter name followed by options. Specify the absolute pathname of the filter unless the filter is in a directory named in one the following:

- ▶ LIBPATH (for filter DLLs)
- ▶ PATH (for UNIX filters) environment variable

For more information on Infoprint Server Transforms, see “Infoprint Server Transforms” on page 176.

Resubmit for filtering

When this field is selected, by entering a `/`, IP PrintWay resubmits batch job data sets to Print Interface. Print Interface calls the filter (if any) for the data format and then writes the data to a new output data set on the JES spool for processing by IP PrintWay. For more information, see “Resubmit for filtering” on page 167.

AOPIPDRR		Processing	
Command ==> _____			
Printer definition name . <u>pokeps</u>			
			More: -
IP PrintWay Line-to-Text Conversion:			
/ Pagination			
Margins: Top . . . _____		Bottom . . . _____	
Page height . . . <u>58</u>			
/ Print page header			
Maximum document size . _____			
Maximum copies. _____			
Forms supported _____ (more)			
Duplex supported. . . . / Simplex / Duplex / Tumble			
Print-error reporting supported . / Character / Position			
Input tray name:	Number: (more)	Output bin name:	Number: (more)
_____	_____	_____	_____
IP PrintWay attributes:			
SOSI mode _ 1. None 2. ASCII 3. Space 4. EBCDIC			
Translation dataset qualifier . _____			
Double-byte translate table . . . _____			
1. BIG5	2. EUCKANJI	3. HANGEUL	
4. IBMKANJI	5. JIS78KJ-ASCII	6. JIS78KJ-JISROMAN	
7. JIS83KJ-ASCII	8. JIS83KJ-JISROMAN	9. KSC5601	
10. SCHINESE	11. SJISKANJI	12. TCHINESE	

Figure 3-11 Processing options panel (Part 2)

IP PrintWay options section

The panel, shown in Figure 3-12 on page 74, is displayed when you put the cursor in the Custom values column for IP PrintWay option as shown in Figure 3-6 on page 67. Use the new panel displayed to define the IP PrintWay options required to communicate with the printer LPD:

- Process data prior to transmission to the remote system
- How long to wait for a response from the remote system
- How long to retain data sets on the JES spool after transmission

Note: These attributes apply to all of the transmission protocols that IP PrintWay supports, which includes lpr, IPP, and direct sockets.

AOIPDIO		IP PrintWay Options	
Command ==> _____			
Printer definition name . <u>pokeps</u> _____			
Retention period:			
Successful. . . .	<u>0000:00:00</u>	Failure . .	<u>0000:01:00</u>
Retry time	<u>0000:00:10</u>		
Retry limit. . . .	<u>1</u>		
Response timeout . . _____			
Exits:			
Begin data set. .	_____	End data set. .	_____ Record. . _____
Document header . .	_____	(extend)	
/ Translate document header			
Document trailer . .	_____	(extend)	
/ Translate document trailer			
Dataset grouping . .	<u>2</u>	1. None 2. Job 3. Concatenate job	
Formatting:			
Line termination. . . .	<u>25</u>		
Transparent data char .	<u>35</u>		
Carriage control type .	<u>-</u>	1. None 2. Machine 3. ANSI	
Delete form feed. . . .	<u>1</u>	1. None 2. Leading 3. Trailing 4. Both	
Formatting.	<u>1</u>	1. None 2. Standard	
		3. Translate only 4. Use FCB	
PostScript header . . .	<u>-</u>	1. Add 2. Ignore	
		3. Landscape 4. Always landscape	
<u>-</u> Omit line termination at EOF			

Figure 3-12 Defining IP PrintWay options

Protocol options

New ISPF Protocol panels for IP PrintWay are included in UW73909 and UW73913. There is a Protocol options panel for each of the supported protocols as follows:

- ▶ LPR protocol as shown in Figure 3-13 on page 75
- ▶ Direct socket protocol as shown in Figure 3-14 on page 75
- ▶ IPP protocol as shown in Figure 3-15 on page 76
- ▶ VTAM protocol as shown in Figure 3-16 on page 77

The panel, shown in Figure 3-13 on page 75, is displayed when you put the cursor in the Custom values column for Protocol as shown in Figure 3-6 on page 67. Here you determine how to transmit data sets from the JES spool to the remote printer lpd or print server. The fields specify the target printer and are usually unique for each printer definition. They include:

- ▶ Printer IP address of the remote printer or print server
- ▶ Print queue name, which is the name in the remote printer or print server

A0PIPDIL		LPR Protocol	
Command ==>			
Printer definition name .	pokeps		
Printer IP address .	9.12.2.4		(extend)
Print queue name .	afccu2		(extend)
LPR Processing Options:			
Mode	2	1. Control file first 2. Control file last 3. Stream 4. Remote PSF	
/ Restrict ports			
- Print banner page			
Banner class .			
Banner job name			
Filename			
Indent			
Owner			
Print function . . .	f		
Title			
Width			
User options			

Figure 3-13 IP PrintWay LPR protocol definition panel

After completing all of the needed options for defining a printer, use PF3 as many times as required to exit the definition of the new printer and then the printer is defined.

3.4.2 IP PrintWay printers with direct socket protocol

From the Add a printer definition panel, shown in Figure 3-4 on page 64, Select **Option 2** to create direct socket printers. IP PrintWay uses a direct sockets printing protocol to transmit data sets to the target printer. This printing protocol is taken from the Direct Sockets Protocol panel, as shown in Figure 3-14.

When defining this type of printer, the Allocation, Processing, and IP PrintWay Options panels are the same as the other IP PrintWay printer definitions.

A0PIDID		Direct Sockets Protocol	
Command ==>			
Printer definition name .	pokep		
Printer IP address .	9.12.2.140		(extend)
Port number	1088		

Figure 3-14 Direct sockets protocol panel

Where:

- IP address field** Specify the IP address or hostname of the target printer or print server.
- Port number field** Specify the port number at which the printer or print server supports printing.

Direct sockets printing protocol is a TCP/IP protocol in which IP PrintWay transmits data directly to a designated port on the target system. To use this protocol, the remote printer or print server must support direct sockets printing.

IP PrintWay transmits data to the printer or print server at the IP address (or host name) and port number that you specify in the printer definition. Some printers support the direct sockets printing protocol as well as other protocols, such as the LPR protocol.

For large data sets, the direct sockets printing protocol can provide better performance. However, you might want to select the LPR protocol to take advantage of the formatting options that IP PrintWay can specify in the LPD control file, for example, printing a banner page, and printing copies.

3.4.3 Defining IP PrintWay printers with IPP protocol

IP PrintWay uses an Internet Printing Protocol (IPP) to transmit data sets to the target printer. To use the IPP protocol, an IPP server must be running in the remote printer or host system.

To start the IPP server on the host system, you must specify the following in the *aopd.conf* configuration file:

```
start-daemons = { ippd }
```

An IPP client running in a user workstation can use the IPP protocol to pass a print data stream to the IPP server on the host. The IPP server on the host then puts the data stream on the JES spool.

Do not confuse the IPP client that sends print requests to the Print Interface IPP server with the IPP client that is part of the IP PrintWay component. The IPP client in IP PrintWay transmits output data sets from the JES spool to IPP servers running in a remote printer or host system. To do this, you need to define an IPP printer or host system in the Printer Inventory by selecting **Option 3** from the Add a printer definition panel, shown in Figure 3-4 on page 64.

When defining this type of printer, the Allocation, Processing, and IP PrintWay Options panels are the same as the other IP PrintWay printer definitions.

The IPP protocol panel is shown in Figure 3-15.

```

AOPIPDII                                     IPP Protocol
Command ==> _____
Printer definition name . color_____
URL. . . . . http://9.12.14.164:631/ipp_____ (extend)
  
```

Figure 3-15 IPP protocol panel for IP PrintWay

IP PrintWay transmits the print data stream to the IPP server at the URL you specify in the printer definition.

Along with the data, IP PrintWay also transmits a control file containing IPP job attributes. If the print request was originally submitted from an IPP client (such as the Windows 2000 IPP client) to the host IPP server, then IP PrintWay transmits all of the IPP job attributes originally specified with the print request. For all print requests, regardless of how the print request was submitted, IP PrintWay passes the following IPP job attributes to the IPP client:

- ▶ Copies
- ▶ Document-name
- ▶ Job-name
- ▶ Requesting-user-name
- ▶ Sides

3.4.4 IP PrintWay printers with VTAM protocol

The Coax Printer Support feature is a priced feature of Infoprint Server Transforms. This feature lets you print to VTAM-controlled printers that are defined to VTAM as LU0, LU1, or LU3 printers. The supported output data streams are Data Stream Compatibility/Data Stream Extended (DSC/DSE) and SNA Character String (SCS). The IP PrintWay component of Infoprint Server uses the Coax Printer Support feature to print to VTAM-controlled printers.

You add a VTAM printer in the Printer Inventory by selecting **Option 4** from the Add a printer definition panel, shown in Figure 3-4 on page 64, just like all the other IP PrintWay printers, and create entries for each of the Custom values. Then for the protocol Custom value, you use the VTAM Protocol panel to specify the Printer LU name as shown in Figure 3-16.

A0PIPDIV	VTAM Protocol
Command ==> _____	
Printer definition name . <u>NS</u>	
Printer LU name. . . <u>SC43PR02</u>	
VTAM Processing Options:	
Printer logmode. . .	_____
Checkpoint pages . .	<u>5</u>

Figure 3-16 VTAM protocol panel for IP PrintWay

Then do the following:

- ▶ Create a VTAM APPL definition for the VTAM printers.

If you have installed the Coax Printer Support feature of the Infoprint Server Transforms and plan to use VTAM-controlled printers, you must create a separate VTAM APPL definition statement for each IP PrintWay FSS.

- ▶ Create an FSS definition for each VTAM printer.

Each IP PrintWay FSS that drives a VTAM-controlled printer must be defined in the Printer Inventory.

Create a VTAM APPL definition

When IP PrintWay prints to VTAM-controlled printers, it runs as a VTAM application program. Therefore, you must create a VTAM APPL definition statement in the SYS1.VTAMLST data set for each IP PrintWay FSS. A separate APPL statement is required for each FSS.

An APPL statement defines a PSF application program node, called a logical unit (LU), within the network domain of a VTAM supervisor services control point. This APPL statement also defines some of the communication capabilities of the application program. You should also define an APPL statement for each PSF functional subsystem application (FSA) for each printer, and you must define at least one FSA per printer.

Create the IP PrintWay FSS APPL statement in a new or existing application major node definition. Figure 3-17 on page 78 shows one model sample APPL statement under a major node named PRINTWAY.

```

PRINTWAY VBUILD TYPE=APPL           Application major node
*
SC43PW0? APPL   AUTH=ACQ           Let IP PrintWay acquire printer
*
* VTAM compares a model application program name with the name of the
* application program trying to open its ACB. It compares each name,
* character for character, scanning from left to right. For each
* character, VTAM looks for a match in the following order:
*
* 1. Most specific match: The actual character (A-Z, 0-9, @, #, or $)
* 2. Next most specific match: A single-character wildcard character (?)
* 3. Least specific match: A multiple-character wildcard character (*)

```

Figure 3-17 Sample APPL statement for IP PrintWay FSSs

AUTH=ACQ This parameter permits IP PrintWay to initiate a session with the printer. AUTH=ACQ is required.

SC43PW0? This APPLID name is a model and indicates that you do not have to define each VTAM printer by an exact name.

Refer to *z/OS IBM Communications Server: SNA Resource Definition Reference*, SC31-8778 for complete information about the APPL statement and its parameters.

Create an FSS definition for each VTAM printer

PSF sends data through VTAM to a printer, either directly or by way of a control unit. The control unit then sends the data to a printer, either directly or by way of PSF/2 or PSF for AIX. To define an FSS definition for a VTAM printer, see “FSS definitions” on page 78.

Attention: It is required to create this definition if any of the FSAs in the FSS will transmit output to VTAM-controlled printers.

3.4.5 FSS definitions

In an IP PrintWay FSS definition, you can specify configuration information that applies to an IP PrintWay FSS and to all FSAs in that FSS. Creating an IP PrintWay FSS definition is optional, except for VTAM-controlled printers.

Also, you must create an FSS definition to change the IP PrintWay configuration defaults. If an IP PrintWay FSS definition does not exist when IP PrintWay starts, IP PrintWay uses default values for the fields you can specify in an FSS definition.

You add an FSS definition in the Printer Inventory by selecting Option 4 from the Printer Inventory Manager panel shown in Figure 3-2 on page 61 and then the FSS, FSA, and Pool Management panel is then displayed, as shown in Figure 3-18 on page 79.

FSS, FSA, and Pool Management panel

From this panel, you can specify IP PrintWay configuration information in the following objects in the Printer Inventory:

FSS definition IP PrintWay uses values specified in this definition when the IP PrintWay FSS starts. In an IP PrintWay FSS definition, you can specify configuration information that applies to an IP PrintWay FSS and to all FSAs in that FSS. Also, you must create an FSS definition to change the IP PrintWay configuration defaults. If an IP PrintWay FSS definition does not exist when IP

PrintWay starts, IP PrintWay uses default values for the fields you can specify in an FSS definition.

Note: If you create an FSS definition after the FSS has started or if you change any information in the FSS definition, you must restart the IP PrintWay FSS to pick up the changes.

FSA definition IP PrintWay uses values specified in this definition when an IP PrintWay FSA starts. You can create one IP PrintWay FSA definition for each IP PrintWay FSA that your installation has defined to JES. In each IP PrintWay FSA definition you can specify the type of tracing to start for the FSA.

Creating FSA definitions is optional. If an FSA definition does not exist, IP PrintWay uses the tracing value specified in the FSS definition. The FSA name must match the name of the FSA as defined to JES.

Note: If you create an FSA definition after the FSA is started or if you change the FSA definition, you must restart the IP PrintWay FSA to use the changed values; however, you do not need to restart the FSS and any other IP PrintWay FSA.

Note: You can use either the Infoprint Server ISPF panels or the Printer Inventory Definition Utility (*pidu*) to create FSS and FSA definitions. Refer to *z/OS Infoprint Server Operation and Administration*, S544-5744 for more information on how to use the Infoprint Server ISPF panels or the *pidu* command.

A0PIPP2		FSA, FSS, and Pool Management
Option ===> _____		
FSA		
1 Add	Add an FSA	
2 List	List FSAs	
3 Select	Select FSAs to list	
FSS		
4 Add	Add an FSS	
5 List	List FSSs	
6 Select	Select FSSs to list	
Pool of Printer Definitions		
7 Add	Add a Pool	
8 List	List Pools	
9 Select	Select Pools to list	

Figure 3-18 FSS, FSA, and Pool Management panel

From this panel, select Option 4, and Figure 3-19 on page 80 is displayed.

```

A0PIPP2                                FSA, FSS, and Pool Management
0      Choose an FSS type to add -----
F      A0PIPSAP => -----
      1 IP PrintWay
      2 PSF for OS/390

F

      5 List          List FSSs
      6 Select        Select FSSs to list

Pool of Printer Definitions
      7 Add          Add a Pool
      8 List          List Pools
      9 Select        Select Pools to list

```

Figure 3-19 Choose an FSS type to add pop-up window

From the pop-up window shown in Figure 3-19, select option 1 and the IP PrintWay FSS definition panel, shown in Figure 3-20, is displayed.

```

A0PIPSA1                                IP PrintWay FSS
Command ==> -----

FSS name. . . PRINTWAY
Description . Ditto                                (extend)

      Old-style translation
      Hipspace blocks . . 8000
      TCP/IP job name . . . TCPIP0E
      Document code page. .
      Applid. . . . . SC43PW01
      National language . . 1 1. English 2. Spanish 3. Japanese
      Trace mode. . . . . 1 1. None 2. Internal 3. No printing 4. Full
      Trace prompt
      Trace table size .

```

Figure 3-20 IP PrintWay FSS panel

On this panel, you define a specific FSS name that is defined in the JES initialization stream, PRINTWAY, as shown in Figure 4-4 on page 98 and Figure 4-5 on page 100.

Note: For a VTAM-controlled printer, you are defining which FSS, by its FSS name PRINTWAY, this printer is controlled from, as an FSA.

The other fields and FSS attributes you enter are as follows:

- Old-style translation** An indicator of whether IP PrintWay uses the standard TCP/IP translate table when converting data from EBCDIC to ASCII. This field applies only for those printer definitions that do not specify the names of document and printer code pages or customized TCP/IP translate tables.
- Hipspace blocks** The maximum number of 4 KB blocks that each functional subsystem application (FSA) in this FSS can use in a hipspace, the system paging space. This field is not required; if it is blank, each IP PrintWay FSA uses 8,000 4 KB blocks, which is 32 megabytes.
- TCP/IP job name** The name of the TCP/IP startup procedure. This field is not required; if it is blank, the FSA uses the default TCP/IP startup procedure name.

Document code page	The name of a default EBCDIC code page to use as the source code page when IP PrintWay translates data from EBCDIC to ASCII. This field is not required; the default code page is IBM-1047.
Applid	The application program ID that IP PrintWay uses to establish a VTAM session with a printer. This ID must match the name of an APPL statement defined to VTAM. This field is <i>required</i> if the <i>VTAM protocol</i> type is selected in any printer definition used by this IP PrintWay FSA.
National language	The language that IP PrintWay uses for messages.
Trace mode	The type of IP PrintWay tracing for the FSS, and the default tracing mode for all functional subsystem applications (FSAs) within the FSS.

Note: Once you add the **Applid** on the IP PrintWay FSS panel, you must ensure that the VTAM major node defining the **Applid** is also active. If not, the IP PrintWay FSS startup *will fail*.

3.5 Defining a PSF for OS/390 printer

PSF for OS/390 is a separately orderable product that runs on z/OS and OS/390 and controls printing on AFP printers from a functional subsystem address space (FSS).

The panel, shown in Figure 3-21, lets you specify a printer definition name, description, and location. You can also specify other attributes that identify this definition, such as a NetSpool logical-unit name, description, and location.

Note: For a printer type of PSF for OS/390, Print Interface converts text data to S/390 line data before placing it on the JES spool.

The rest of the printer definition is divided into sections (such as Allocation and Processing). In each section, you can specify different types of printer attributes.

A0PIPDC2
PSF for OS/390 Printer Definition
Command ==> _____
Printer definition name . pokexfd
Description . AFP Printer - POKE (extend)
Location. . . 2C-16 (extend)
Section
Allocation ==> _____ Custom values ==> *
Processing ==> _____ ==> *
NetSpool options ==> _____ ==> _____
NetSpool end-of-file ==> _____ ==> _____
NetSpool LU name . _____ LU classes (extend)

Figure 3-21 Panel to define a PSF for OS/390 printer

Because this printer can print AFP data, you should use data transforms in the Processing section definitions to translate the data to AFP format if users submit output to this destination that contains other datastream formats, as shown in Figure 3-22 on page 82.

Of course, if you have printers that print PostScript and PCL, you might choose to have a specific printer print those data streams.

AOPIPDRR		Processing	
Command ==> _____			
Printer definition name . <u>pokexfd</u>			
			More: +
Document code page . . _____			
Printer code page. . . <u>IBM-1047</u>			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
/ Line data			(extend)
/ MO:DCA-P			(extend)
/ PostScript	<u>ps2afp.dll</u>		(extend)
/ Text			(extend)
/ PCL	<u>pcl2afp.dll</u>		(extend)
/ PDF	<u>ps2afp.dll</u>		(extend)
/ SAP	<u>sap2afp.dll</u>		(extend)
- Other			(extend)
_ Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top . . .	Bottom . .	Left . .	Right . .
Line length . . .		Page length . .	
Tabs: Vertical . .			(extend)
Horizontal . .			(extend)
IP PrintWay Line-to-Text Conversion:			
/ Pagination			-
Margins: Top . .	Bottom . .		
Page height . . .	<u>58</u>		

Figure 3-22 Processing section panel (Part 1) showing transforms

As shown in Figure 3-22, the way to define data stream translation using the transforms is to specify the filter names shown as:

ps2afp.dll - pcl2afp.dll - sap2afp.dll

When the data stream arrives at the Print Interface, it is converted to AFP format and then placed on the spool.

3.5.1 Using transforms in the printer definition

When it is necessary to transform a data stream that arrives from a user, the user has been told to use a specific destination (Printer definition name) in the Printer Inventory. When that data stream arrives, Print Interface uses the transform specified for that data stream as specified in the Printer Inventory in the Processing section. Transforms can be specified for all the printer types.

Converting AFP data

Figure 3-23 on page 83 shows Line data and MO:DCA-P data being converted to PostScript when it is received by the Print Interface AOPLPD.

AOPIPDRR		Processing	
Command ==> _____			
Printer definition name . <u>poke</u> _____			
			More: +
Document code page . . _____			
Printer code page . . . <u>ISO8859-1</u>			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
/ Line data	<u>afp2ps.dll %filter-options</u>		(extend)
/ MO:DCA-P	<u>afp2ps.dll %filter-options</u>		(extend)
/ PostScript	_____		(extend)
/ Text	<u>aopfiltr.so</u>		(extend)
/ PCL	_____		(extend)
- PDF	_____		(extend)
- SAP	_____		(extend)
- Other	_____		(extend)
_ Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top	Bottom	Left	Right
Line length		Page length	
Tabs: Vertical			(extend)
Horizontal			(extend)
IP PrintWay Line-to-Text Conversion:			
/ Pagination			
Margins: Top	Bottom		
Page height	<u>58</u>		

Figure 3-23 Processing section showing transform definitions

%filter-options

When you specify %filter-options, as shown in Figure 3-23, it causes those options that are specified in the filter-options job attribute (specified, for example, on the **lp** command) to be passed to the transform. You can type the %filter-options option in any position relative to the other filter options. If you specify filter options to the right of %filter-options, those options override the same options specified in the filter-options job attribute.

Data stream transforms

With transforms configured, PSF can support all data formats except Other; without transforms configured, PSF can support Line data, MO:DCA-P, and Text, as shown in Figure 3-22 on page 82.

Note: See “Infoprint Server Transforms” on page 176 for a complete description of the data stream transforms.

3.5.2 Customize PSF for OS/390 for sharing network printers

When customizing PSF for OS/390 to send AFP data to IBM network printers, be sure to configure PSF for OS/390 for printer sharing if you also want to print on these printers from IP PrintWay. When configured for printer sharing, IP PrintWay can also send non-AFP data streams to the printer.

The Printer Inventory and the PRINTDEV statement from the PSF procedure are used to specify parameters for:

- ▶ Dedicated network-attached printers
- ▶ Shared printers with other print drivers

PSF supports the sharing of printers with other print drivers by three different methods:

- ▶ JES spool-driven
- ▶ VTAM RELREQ
- ▶ Port switching

PSF releases a printer only at the end of a job, so that output is never split. This frees the printer when there is no longer output to be selected to be used by another printer driver. If you do not configure PSF for OS/390 for printer sharing, PSF for OS/390 does not release the printer to receive data from other sources.

Refer to *PSF for OS/390: Customization*, S544-5622 for information about how to configure PSF for OS/390 for printer sharing.

JES3 environment support

JES3 allows printers to be started dynamically when output becomes available by specifying DYNAMIC=YES on the printer DEVICE statement, as shown in Figure 3-24. When there is no more output for that printer, it stops and can become available for another application to use.

```
FSSDEF,TYPE=WTR,FSSNAME=IPDSWAY,PNAME=IPDSWAY,TERM=YES
DEVICE,DTYPE=PRTAFP1,
  JNAME=IPDSWAY,DGROUP=IPDS,DGRPNLY=NO,
  JUNIT=(,SC50,UR,,,SC49,UR,,,SC43,UR),
  FSSNAME=IPDSWAY,DYNAMIC=YES,BURST=NO,FORMS=(NO,STD),
  WC=(K),WS=(CL),HEADER=NO
```

Figure 3-24 Printer example using DYNAMIC=YES

PSF and IP PrintWay sharing

Figure 3-25 on page 85 and Figure 3-26 on page 86 show examples of IBM Network Printer 24 definitions for printer sharing in a JES environment. The two printer definitions are defined in the Printer Inventory for the same IP address, 9.12.2.8. One printer, IPDSWAY, is a PSF printer and the other, NP24, is an IP PrintWay printer.

<pre> #PSF FSS and printer definitions #----- # psf-fss IPDSWAY at 9.12.2.8 #----- create psf-fss IPDSWAY description = "JES3 SC43 IPDS FSS" tcpip-job-name = TCPIPOE ; #----- # printer IPDSPOK #----- create printer IPDSPOK location = "2C16 Poughkeepsie" description = "AFP printer" print-error-messages-maximum = 50 error-disposition = quit print-error-reporting = all destination = ANYLOCAL output-class = I document-formats-supported = { line modca postscript pcl pdf other } printer-type = psf-mvs filters = { postscript -> "ps2afp.dll %filter-o text -> aopfiltr.so pcl -> "pcl2afp.dll %filter-options pdf -> "ps2afp.dll %filter-options" } forms = IPDS include-processing = ASCII-pcl-EBDIC ; </pre>	<pre> # IP PrintWay printer definition #----- # printer NP24 at 9.12.2.8 #----- create printer NP24 printer-codepage = ISO8859-1 filters = { line -> "afp2pcl.dll %filter-options" text -> aopfiltr.so other -> "afp2pcl.dll %filter-options" } location = "2C16 Poughkeepsie" description = "NP24 - Has an IPDS card" printer-type = ip-printway retry-limit = 1 retry-time = 0000:00:10 failure-retention-period = 0000:01:00 successful-retention-period = 0000:01:00 print-queue-name = PASS printer-ip-address = 9.12.2.8 normal-output-disposition = purge hold = no forms = STD output-class = J burster-trimmer-stacker = no destination = NP24 document-formats-supported = { line text pcl other } lpr-restrict-ports = yes input-tray-map = { "TRAY 3" -> 3 } resubmit-for-filtering = yes ; </pre>
--	--

Figure 3-25 Shared IBM Network Printer 24 definitions for PSF and IP Printway

PSF JCL procedure

A PSF JCL procedure, shown in Figure 3-26 on page 86, has a TIMEOUT= parameter on the PRINTDEV statement, which defines the action that PSF is to take after a time-out when no output is available from JES. When the time specified by the disconnect interval parameter, DISCINTV=, expires and no output is available from JES, PSF takes the following actions:

- STOP** PSF stops the printer FSA, which can then be restarted only by an operator command or by JES.
- REDRIVE** PSF redrives the printer FSA according to the value specified by the Management mode parameter (MGMTMODE= **OUTAVAIL** | DIALIN | IMMED). REDRIVE is the default action.

```

//IPDSWAY  PROC TCPHLQ=TCPIP
/* THIS IS A SAMPLE PROCEDURE FOR STARTING IP PRINTWAY
//IEFPROC  EXEC  PGM=APSPPIEP,REGION=1M,TIME=NOLIMIT,PARM=(,,,TCPIPOE)
//STEPLIB  DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=&TCPHLQ..SEZALINK,DISP=SHR
//JOBHDR   OUTPUT PAGEDEF=V06483,          /* JOB HEADER PAGE          */
//          FORMDEF=A10120,CHARS=60D8      /* FORMDEF: ALTERNATIVE BIN*/
//JOBTLR   OUTPUT PAGEDEF=V06483,          /* JOB TRAILER PAGE        */
//          FORMDEF=A10110,CHARS=60D8      /* FORMDEF: MAIN BIN       */
//DSHDR    OUTPUT PAGEDEF=V06483,          /* DATA SET SEPARATOR     */
//          FORMDEF=A10110,CHARS=60D8      /* FORMDEF: MAIN BIN       */
//MSGDS    OUTPUT PAGEDEF=A08682,          /* MESSAGE DATA SET       */
//          FORMDEF=A10110,CHARS=60D8      /*                          */
//FONT01   DD DSN=SYS1.FONT300,DISP=SHR    /*                          */
//          DD DSN=SYS1.FONTLIB,DISP=SHR   /*                          */
//          DD DSN=SYS1.FONTLIBB,DISP=SHR  /*                          */
//PSEG01   DD DSN=SYS1.PSEGLIB,DISP=SHR    /* INSTALLATION PAGE SEGMENTS*/
//OLAY01   DD DSN=SYS1.OVERLIB,DISP=SHR    /* INSTALLATION OVERLAYS    */
//PDEF01   DD DSN=SYS1.PDEFLIB,DISP=SHR    /* SYSTEM PAGE DEFINITIONS  */
//FDEF01   DD DSN=SYS1.FDEFLIB,DISP=SHR    /* SYSTEM FORM DEFINITIONS  */
//SYSTCPD  DD DISP=SHR,DSN=TCPIPOE.&SYSNAME..TCPPARMS(TCPDATA)
//SYSPRINT DD SYSOUT=*
//IPDSPOK  PRINTDEV TRACE=NO,
//          FONTDD=*.FONT01,              /* FONT      LIBRARY DD      */
//          OVLYDD=*.OLAY01,              /* OVERLAY   LIBRARY DD      */
//          PSEGDD=*.PSEG01,              /* SEGMENT   LIBRARY DD      */
//          PDEFDD=*.PDEF01,              /* PAGEDEF   LIBRARY DD      */
//          FDEFDD=*.FDEF01,              /* FORMDEF   LIBRARY DD      */
//          JOBHDR=*.JOBHDR,              /* JOB HEADER SEPARATOR OUTPUT */
//          JOBTLR=*.JOBTLR,              /* JOB TRAILER SEPARATOR OUTPUT */
//          MESSAGE=*.MSGDS,              /* MESSAGE DATA SET OUTPUT   */
//          PAGEDEF=A08682,              /* DEVICE PAGEDEF DEFAULT    */
//          FORMDEF=A10111,              /* DEVICE FORMDEF  A10111    */
//          CHARS=60D8,                  /* DEVICE DEFAULT FONT       */
//          PIMSG=(NO,0),                /* SUPPRESS ERROR MESSAGES   */
//          DACK=BLOCK,                  /* BLOCK DATA CHECKS        */
//          DISCINTV=120,                 /* END SESSION W/ PRINTER INTVL */
//          MGMTMODE=OUTAVAIL,            /* OUTPUT AVAILABLE          */
//          RELMODE=NONE,                 /* RELEASE MODE=IDLE         */
//          RELINTV=15,                   /* RELEASE INTERVAL          */
//          NORESP=NOTIFY,                 /* WHAT WHEN RESPTIME EXPIRES */
//          RESPTIME=45,                   /* MAX WAIT TIME FOR RESPONSE */
//          TIMEOUT=REDRIVE,              /* NO OUTPUT AVAILABE ACTION */
//          IPADDR='9.12.2.8',            /* IP ADDRESS OF PRINTER     */
//          PORTNO=5001                   /* IP PORT OF PRINTER        */
//IPDSPOK  ENDCNTL

```

Figure 3-26 PSF FSS IPDSWAY JCL procedure

3.6 Defining a general printer

Select a General type for printers that are not IP PrintWay or PSF for OS/390 printers. An example of a General type is a line printer controlled by JES.

The first panel displayed when **Option 6** is selected from the Primary menu is shown in Figure 3-2 on page 61.

AOPIPC3 General Printer Definition		
Command ==> _____		
Printer definition name . _____		
Description . _____		(extend)
Location. . . _____		(extend)
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> _____
Processing	=> _____	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> _____
NetSpool LU name . _____ LU classes . . . _____		(extend)
Spooling mode. . . 1. Line 2. Stream		

Figure 3-27 General printer type panel

The Spooling mode shown in the panel is a value that indicates how Print Interface writes the input data to the output data set on the JES spool. Select a value that is suitable for the specific printer. This option is for General type printers only.

- 1 = Line:** Print Interface writes the data in records (default). This value is suitable for JES and AFP printers.
- 2 = Stream:** Print Interface writes the data as a data stream, with control characters to indicate the end of lines.

3.7 Defining components

A component can be defined for any of the six sections in any of the printer types you define in the Printer Inventory. The purpose of defining components is to save time in defining printer definitions. If you are defining a large number of printers and many of them have the same attributes, for those attributes that are the same, create a component with these attributes and give it a name. Then, when you define each printer, specify the component name in the appropriate section on the Printer Definition panel, as shown in Figure 3-28.

Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> *
Processing	=> <u>PCL Printer</u>	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> *
Protocol	=> _____	=> *

Figure 3-28 Component name pcl_printer

3.7.1 Creating a component name definition

You are creating a component that contains attributes that are common to several printer definitions. You can define one or more components for each section of a printer definition, such as the Allocation section and the Processing section or any of the other Sections.

To create the component for a Processing section, place the cursor on the Processing line under the Component name, press enter, and Figure 3-29 on page 88 is displayed.

Component List	
Command ==> <u>a</u>	Scroll ==> <u>CSR</u>
Processing	
Actions:	
A-Add B-Browse C-Copy D-Delete E-Edit /-Select P-Show use R-Rename	
A Component Name	Description
= =====	
***** Bottom of data *****	

Figure 3-29 Create a new component

Figure 3-29 shows that no components exist at the moment. To add a component, place an *a* for Add on the command line, press Enter, and the Processing section panel appears, as shown in Figure 3-30.

Edit	Processing
Command ==> _____	
Component name. <u>pcl_printer</u>	
Description . <u>Generic PCL options</u>	
	More: +
Document code page . . . _____	
Printer code page. . . <u>ISO8859-1</u>	
Print Interface supported data formats and associated filters:	
Data format:	Filter:
/ Line data	<u>afp2pcl.dll</u> (extend)
/ M0:DCA-P	<u>afp2pcl.dll</u> (extend)
- PostScript	_____ (extend)
/ Text	<u>aopfiltr.so</u> (extend)
/ PCL	_____ (extend)
- PDF	_____ (extend)
- SAP	_____ (extend)
- Other	_____ (extend)
/ Resubmit for filtering	
Conversion between SCS and Line:	
Margins: Top . . . _____	Bottom . . . _____
Line length . . . _____	Page length . . . _____
Tabs: Vertical . . . _____	_____ (extend)
Horizontal . . . _____	_____ (extend)
IP PrintWay Line-to-Text Conversion:	
/ Pagination	
Margins: Top . . . _____	Bottom . . . _____
Page height . . . <u>58</u>	

Figure 3-30 Component definitions for a Processing section

This component definition panel has the fields filled in for a `pcl_printer` component name and description, the Data formats and filters and the Resubmit for filtering. These attributes are to be the same for every new printer definition when the name `pcl_printer` is filled into the component field for the Processing section. When you use PF3 to return, the component entry is created, as shown in Figure 3-31 on page 89.

Component List		Row 1 to 1 of 1
Command ==>		Scroll ==> CSR
Processing		
Actions:		
A-Add B-Browse C-Copy D-Delete E-Edit /-Select P-Show use R-Rename		
A Component Name	Description	
= =====		
pcl_printer	Generic PCL options	
***** Bottom of data *****		

Figure 3-31 New component name entry just created

Using the component name

To use the component just created, when you are adding a new printer definition, the add printer definition panel shown in Figure 3-32 is displayed.

IP PrintWay Printer Definition		
AOPIPC1 Command ==> _____		
Printer definition name . PCL5 _____		
Description . PCL_printer _____ (extend)		
Location. . . Room 606 Bldg 740 _____ (extend)		
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> _____
Processing	=> _____	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> _____
Protocol	=> _____	=> *
_ Use DEST, CLASS, and FORMS for IP PrintWay printer selection		
NetSpool LU name . _____ LU classes . . _____ (extend)		

Figure 3-32 Using the component name in a new printer definition

First, when adding a new printer definition, fill in the printer name, description, and location fields. Of course, you could create components for each of the Sections you need to define. In our example, place the cursor on the Processing section component name field. When you now press Enter, Figure 3-33 is displayed.

Component List		Row 1 to 2 of 2
AOPIPC1 Command ==> _____		Scroll ==> CSR
Processing		
Actions:		
A-Add B-Browse C-Copy D-Delete E-Edit /-Select P-Show use R-Rename		
A Component Name	Description	
= =====		
/ pcl_printer	Generic PCL options	
- ASCII-pcl-EBDIC	Ditto	
***** Bottom of data *****		

Figure 3-33 A Component list panel of all Processing components

Select the Processing component you want to use for this printer definition with a /, press Enter and Figure 3-34 on page 90 is displayed.

AOPIPC1 IP PrintWay Printer Definition		
Command ==> _____		
Printer definition name . <u>PCL5</u> _____		
Description . <u>PCL printer</u> _____		(extend)
Location. . . <u>Room 606 Bldg 740</u> _____		(extend)
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> _____
Processing	=> <u>pcl printer</u> _____	=> <u>*</u> _____
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> _____
Protocol	=> _____	=> <u>*</u> _____
__ Use DEST, CLASS, and FORMS for IP PrintWay printer selection		
NetSpool LU name . _____ LU classes . . . _____		(extend)

Figure 3-34 Component name selected for Processing section

Using components eliminates much typing on the various panels during a printer definition. You can use the same component name for each Section. The component name only has to be unique for an individual Section.

Figure 3-35 shows a display of the component from the OMVS shell command line using the **pidu** command. For more information on this command, see “Using the pidu command” on page 215.

```

ROGERS @ SC43: />pidu -c 'display processing pcl_printer ; '

# display processing pcl_printer
  filters = {
    line -> afp2pcl.dll
    modca -> afp2pcl.dll
    text -> aopfilttr.so
  }
  document-formats-supported = {
    line
    modca
    text
    pcl
  }
  resubmit-for-filtering = yes
  printer-codepage = ISO8859-1
  description = "Generic PCL options"
ROGERS @ SC43: />
===>

```

INPUT

Figure 3-35 A display of a component using the pidu command

See *z/OS Infoprint Server Operation and Administration*, S544-5744 for additional information on using components.

3.8 Display defined printers

To display all the printers that are defined in the Printer Inventory, use Option 2 from the Primary Menu, shown in Figure 3-2 on page 61. This displays a printer definition panel as shown in Figure 3-36. With the action characters shown, you can Browse, Copy, Delete, Edit, or Change the type of the new definition. The following display is the printer used in our example.

AOPIPDLI		Printer Definition List				Row 13 to 36 of 41	
Command ==>						Scroll ==> CSR	
Actions:							
A-Add	B-Browse	C-Copy	D-Delete	E-Edit	X-Chg	type	
A	Printer Name	Type	Description	Location	DEST	C FORMS	LU Name
=	=====	=====	=====	=====	=====	=====	=====
_	pokeps	IPPW	3130 Postscr	Room 2c-16 B	POKEPS	J STD	

Figure 3-36 Printer Definition List display panel

If you need to Add, Browse, Copy, or Edit the new definition, you can use the following:

- ▶ ISPF application with the action characters shown in Figure 3-36.
- ▶ PIDU utility

3.8.1 Display defined printer using PIDU

The Printer Inventory Definition Utility (PIDU) is a program to manage objects in the Printer Inventory. Inventory objects include:

- ▶ Printer definitions
- ▶ Printer pool definitions
- ▶ Components
- ▶ FSS definitions and FSA definitions

Instead of using the Infoprint Server ISPF panels to manage objects in the Printer Inventory, you can use the PIDU program. The ISPF panels display Inventory objects in a readable format, check for required attributes, and provide defaults for some attributes.

However, you might find the PIDU program useful for creating and editing many objects at the same time. Also, the PIDU program lets you perform some functions that you cannot perform with Infoprint Server ISPF panels.

For example, you can export and dump Inventory objects to a file, and you can perform more powerful searches of the Printer Inventory.

The output from this command is shown in Figure 3-37 on page 92. You can use this display to verify all of the options chosen, which includes any defaults for those characteristics that have defaults on the panel when it is first displayed during the definition phase.

Note: For a complete description of the PIDU utility, see “Printer Inventory Definition Utility” on page 213.

You can use the PIDU utility to display the definition using the following command from the OMVS command line:

```
pidu -c 'display printer pokeps ; '
```

You receive the following on your UNIX screen:

```
# display printer pokeps ;
  printer-codepage = ISO8859-1
  filters = {
    text -> aopfiltr.so
  }
location = 2c-16
description = "3130 Postscript"
printer-type = ip-printway
retry-limit = 1
retry-time = 0000:00:10
failure-retention-period = 0000:01:00
successful-retention-period = 0000:00:00
print-queue-name = afccu2
printer-ip-address = 9.12.2.4
normal-output-disposition = purge
hold = no
forms = STD
output-class = J
burster-trimmer-stacker = no
destination = POKEPS
document-formats-supported = {
  line
  postscript
  text
  pcl
}
lpr-print-banner = no
printway-formatting = none
dcf-routing = yes
duplex = yes
```

Figure 3-37 Display of defined printer using the pidu command



Implementing IP PrintWay

This chapter describes how IP PrintWay transmits output data sets from JES2 or JES3 to remote printers or to host systems in a TCP/IP network. A print server can be running on the host system. The remote printer or host system must support either the LPR/LPD protocol, the Internet Printing Protocol (IPP), or direct socket printing.

IP PrintWay also transmits output data sets from JES2 or JES3 to VTAM-controlled printers that are defined to VTAM as LU0, LU1, or LU3 printers. The supported output data streams are Data Stream Compatibility/Data Stream Extended (DSC/DSE) and SNA Character String (SCS). This support in IP PrintWay requires that you install the Coax Printer Support feature of Infoprint Server Transforms.

This chapter contains the following:

- ▶ IP PrintWay customization
- ▶ Defining an IP PrintWay functional subsystem
- ▶ IP PrintWay data set customization
- ▶ IP PrintWay sample exits
- ▶ IP PrintWay port ranges
- ▶ Defining JES devices for IP PrintWay printers
- ▶ Printing to IP PrintWay printers using JCL

4.1 IP PrintWay

The IP PrintWay component of Infoprint Server operates as a functional subsystem (FSS) that is started by JES and can transmit output data sets from the JES spool to the following types of printers:

- ▶ TCP/IP-attached printers, using any of the following TCP/IP transmission protocols:
 - LPR to LPD
 - Direct socket printing
 - Internet Printing Protocol (IPP). Java is no longer required after the Infoprint Server PTF UW74452 and the PTF for APAR OW46515.
- ▶ VTAM-controlled printers defined to VTAM as one of the following types of logical units:
 - LU0 (Data Stream Compatibility (DSC)/Data Stream Emulation (DSE))
 - LU1 (SNA Character String (SCS))
 - LU3 (DSC/DSE)

VTAM printers require the Coax Printer Support feature of the Infoprint Server Transforms product with PTF UW74922 applied and Infoprint Server PTFs UW73909 and UW73913.

4.1.1 IP PrintWay TCP/IP considerations

If IP PrintWay is to print to TCP/IP-attached printers, you must verify that the TCPIP.DATA data set contains the following statements used by IP PrintWay:

TCPIPJOBNAME This statement specifies the name of the TCP/IP started task. If you have several TCP/IP instances with different names, the administrator can specify the name of the TCP/IP started task that IP PrintWay is to use in the FSS definition for the IP PrintWay FSS in the Printer Inventory. If you specify a TCP/IP name in both IP PrintWay and TCPIP.DATA, IP PrintWay uses the name in the FSS definition.

If you do not specify a name in either the TCPIPJOBNAME statement or in the FSS definition, IP PrintWay uses the default name TCPIP.

DATASETPREFIX The DATASETPREFIX statement specifies the high-level qualifier for TCP/IP data sets. IP PrintWay uses this high-level qualifier when searching for TCP/IP translation tables.

If you create a unique TCPIP.DATA data set for use by IP PrintWay, you can point to that data set in the IP PrintWay startup procedure with the //SYSTCPD DD statement or you name the data set to hlq.TCPIP.DATA where hlq is the name of the IP Printway started task.

TCPIP.DATA search sequence

IP PrintWay uses the following search sequence to find the TCPIP.DATA:

1. The data set defined by the SYSTCPD DD statement in the IP PrintWay startup procedure
2. A data set named hlq.TCPIP.DATA, where hlq is the name of the IP PrintWay started task
3. A data set named SYS1.TCPPARMS(TCPDATA)
4. A data set named hlq.TCPIP.DATA, where hlq. is the TCP/IP high-level qualifier found earlier in the search sequence or the system default value of TCPIP

If you change any of the values in the TCPIP.DATA data set, restart TCP/IP to pick up the changes.

4.2 IP PrintWay functional subsystem

A functional subsystem is an extension of JES that runs in its own address space separate from the JES address space. An FSS provides support for a function peripheral to JES processing, such as a peripheral device or other component. A functional subsystem application (FSA) executes within the functional subsystem address space and is defined to provide application-specific support to peripheral functions. An FSA allows devices to operate outside of direct JES control. The support contained in the FSS/FSA, then, is specific to the task. In Figure 4-1, the JES address space starts the FSS address space when the operator starts a writer. The writer is defined in the JES initialization statements and actually becomes the FSA portion of the FSS address where IP PrintWay executes.

IP PrintWay executes in an FSS address space as an FSA, as shown in Figure 4-1. .

Note: There is no limit to the number of FSSs that you can define, nor is there a limit to the number of FSAs that you can define to run under an individual FSS.

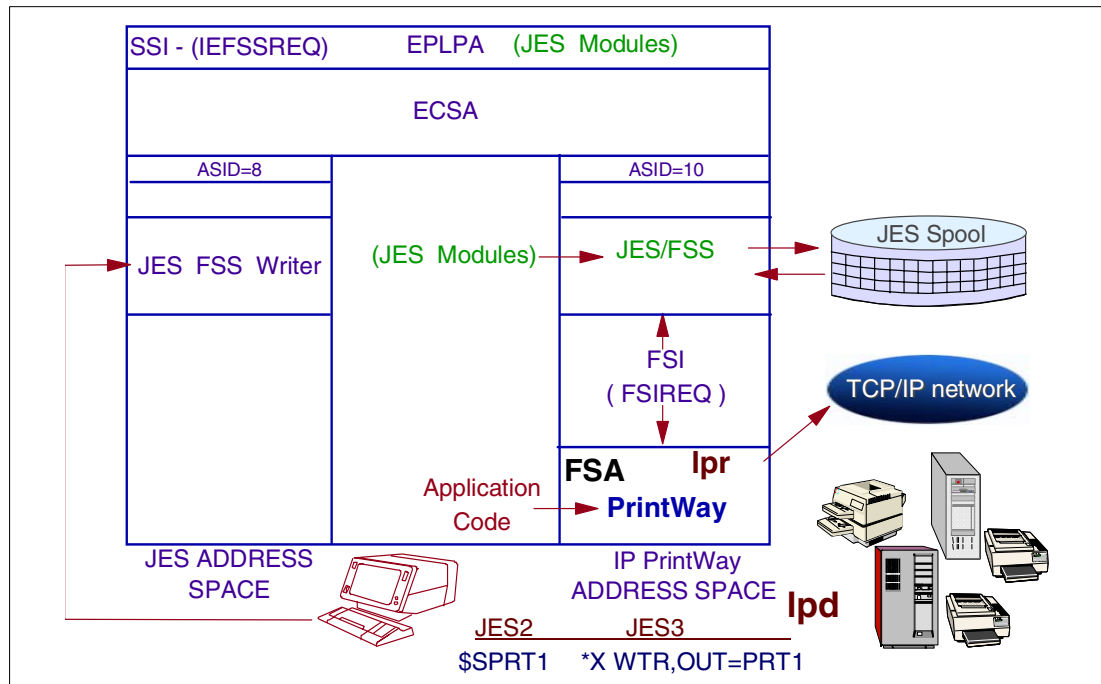


Figure 4-1 JES address space communicating with IP PrintWay FSS address space

You define one or more IP PrintWay functional subsystem applications (FSAs) in the JES initialization statements. Once started, the FSA uses the support facilities of the FSS to communicate with JES.

An FSA is started by the operator as follows (see Figure 4-1):

```
JES2: $SPRT1
JES3: *X WTR,OUT=PRT1
```

Several IP PrintWay FSAs can run in the same FSS address space. Figure 4-2 shows three FSAs that have been started in the FSS named PRTWYA1. Each FSA, once started, selects a data set to print by requesting the data set from JES. The data set is then read from the JES spool by the FSS address space and passed to IP PrintWay through the FSI, and IP PrintWay uses the **lpr** command to pass the data set through TCP/IP to the LPD.

IP PrintWay keeps track of the work in process in the transmission queue data set. The IP PrintWay transmission queue is a VSAM cluster and contains an entry for each data set that IP PrintWay is transmitting to the remote systems or that IP PrintWay is retaining on the JES spool.

Only one IP PrintWay FSS is needed. However, for improved throughput and more efficient use of system resources up to 2000 FSSs can be defined. If you define more than one IP PrintWay FSS, specify the same transmission queue data set in the startup procedure for each FSS to ensure that data sets are always transmitted in the same order as IP PrintWay acquires them from the JES spool.

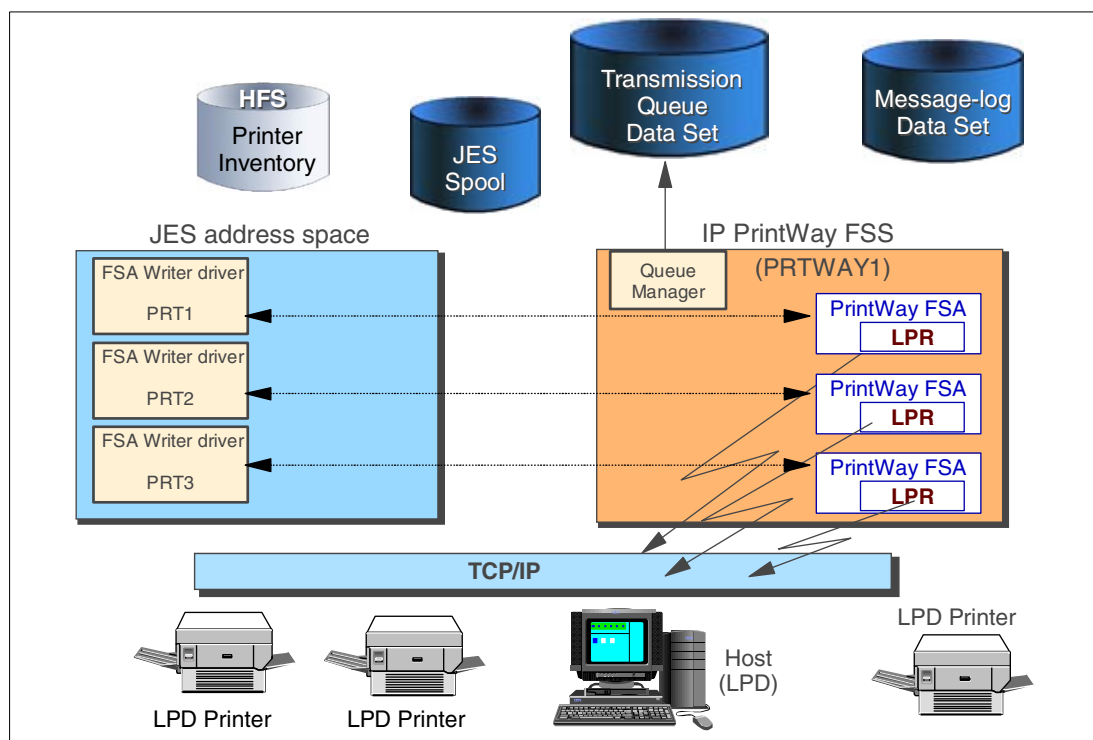


Figure 4-2 IP PrintWay and JES component overview

When multiple IP PrintWay FSSs share a transmission queue data set, as shown in Figure 4-3 on page 97, you must allocate the data set with SHAREOPTIONS (4 3). When you define the FSSs in this way, each FSS reads all of the transmission queue entries and ensures that data sets are sent in the right order.

Each IP PrintWay FSA can transmit data sets, one data set at a time, to many printers or print servers. When an FSA selects a data set from the JES spool, the IP address determines which printer the FSA passes the data set to.

For improved throughput and more efficient use of system resources, you should define more than one FSA for each IP Printway FSS. IBM recommends that you do not create more than 35 FSAs per FSS. You can define up to 64 FSAs in an FSS. As a guideline for the number of FSAs, you should estimate how many jobs must print concurrently and specify at least that many FSAs.

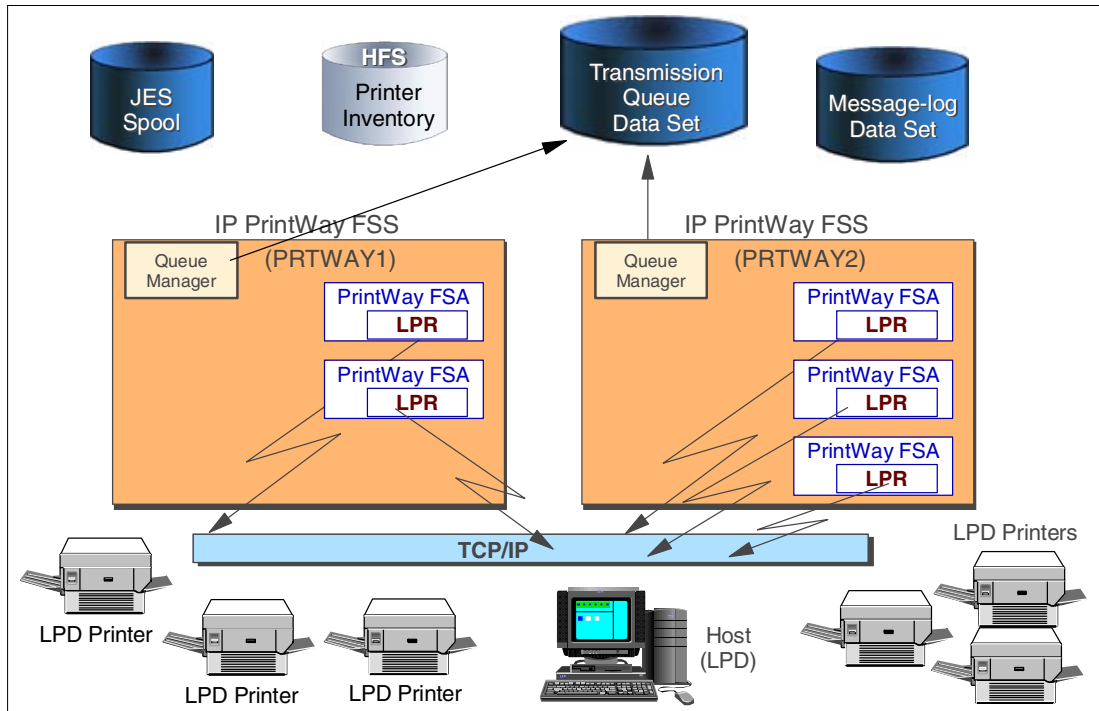


Figure 4-3 Multiple IP PrintWay FSS address spaces

For more information about JES functional subsystems, refer to *z/OS JES2 Initialization and Tuning Guide*, SA22-7532 or *z/OS JES3 Initialization and Tuning Guide*, SA22-7549.

4.3 IP PrintWay customization

To support the FSS/FSA environment for IP PrintWay, you need to do the following:

- ▶ Define one or more FSS address space definitions.
- ▶ Define devices in the JES initialization statements.

The JES initialization streams must include statements for printer devices that will be needed to start an FSA in a specified IP PrintWay FSS address space. In the example in Figure 4-3 there are five JES printer devices started in the JES address space (not shown), which then starts the three FSAs in one FSS address space and two in the other.

- ▶ Define work selection criteria for the FSA to select data sets.
- ▶ Define one or more IP PrintWay procedures.

The IP PrintWay procedure must include statements for each device FSA that can be started in the FSS address space. See “IP PrintWay procedure” on page 114.

4.3.1 Defining the IP PrintWay functional subsystem in JES2

JES2 FSS definitions can be done either with JES2 initialization statements or, after initialization, with the \$ADD FSS and \$ADD PRT operator commands. FSS definitions for IP PrintWay with the JES2 initialization statements require:

- ▶ One FSS(fss_name) statement for each IP PrintWay FSS
- ▶ One PRTnnnnn statement for each FSA under the control of an FSS

Other JES2 initialization statements that have considerations for IP PrintWay:

- ▶ On the JES2 SPOOLDEF statement, specify TRKCELL=5 for improved performance. Also, specify TRKCELL=YES on the PRTnnnnn statements.
- ▶ On the JES2 OUTCLASS statements, specify BLNKTRNC=NO for data sets in an output class that IP PrintWay processes to suppress trailing blank truncation. The setting of this parameter affects the formatting and transformation of the printed output.

Figure 4-4 shows sample JES2 statements for one FSS named PRINTWAY and one associated FSA named PRT123.

```
JES2 initialization statements:
  FSS(PRINTWAY)  PROC=ANFWPROC,AUTOSTOP=YES
  PRT123         CLASS=J,FSS=PRINTWAY,MODE=FSS,PRESLECT=YES,
                 START=NO,TRKCELL=YES,WS=(Q/)

JES2 operator commands:
  $ADD FSS(PRINTWAY),PROC=ANFWPROC,AUTOSTOP=YES
  $ADD PRT123,CLASS=J,MODE=FSS,FSS=PRINTWAY,TRKCELL=YES,WS=(Q/)
```

Figure 4-4 Sample JES2 definitions for an FSS and one associated FSA

FSS statement

The following briefly discusses the JES2 initialization statements shown in Figure 4-4. Some additional parameters that you might want to specify for an IP PrintWay FSS and FSA are also covered. For a complete list of JES parameters and detailed descriptions of each of the parameters, refer to *OS/390 JES2 Initialization and Tuning Reference*, SA22-7533.

FSS(fss_name) Define this statement for each FSS you want to use. Each FSS must have a unique name specified as a 1- to 8-character name. In the example, the FSS name is PRINTWAY. This FSS name must match the FSS=name on the PRTnnnnn statement for each associated FSA.

PROC Specifies the name of the procedure for starting this FSS. Different FSS statements can refer to the same startup procedure. This parameter is required.

AUTOSTOP Specifies YES or NO for whether or not the FSS address space is to be stopped automatically if all FSAs under control of the FSS are stopped. This parameter is optional; if you omit it, the default is NO.

HASPFSSM Specifies the 1- to 8-character name of the load module to be loaded into the functional subsystem address space. This load module contains the various JES2-supplied FSI service routines. If your installation uses the default libraries and module names for the JES2 component, as distributed by IBM, omit this parameter or code HASPFSSM=HASPFSM.

PRTnnnnn statement

Each FSA you want to start must be defined in the JES2 initialization statements with the PRTnnnn statement as follows:

PRTnnnnn The PRTnnnnn statement defines the FSA to JES2. Each FSA must have a unique name; therefore, nnnnn must be a unique number from 1 to 32767.

CL I QUEUE Specifies the output classes processed by this FSA. List all classes to be selected by this FSA; do not separate classes with a comma. If you designate CL or Q as a work-selection criterion on the WS parameter, the FSA processes data sets whose class matches one of the values specified here.

FORMS	Specifies the 1- to 8-character form names processed by this FSA. List from one to eight different form names that can be selected by this FSA; separate each form name with a comma.
FSS	Specifies the name of the FSS associated with this FSA. This parameter is required.
MODE	MODE=FSS specifies that the FSA is managed by an FSS. This parameter is optional. The default is FSS if you code the FSS parameter.
PRESELECT	YES NO specifies whether output data sets are preselected for this FSA. This parameter is optional. The default is YES.
ROUTECD	Specifies the destinations processed by this FSA. List from one to four different destination names that can be selected by this FSA; separate the destination names with a comma.
START	YES NO specifies whether or not JES2 is to automatically start this FSA whenever JES2 starts. If you specify START=NO, the operator must start the FSA. Consider specifying START=NO so that TCP/IP can be started before IP PrintWay. This parameter is optional. START=YES is the default.
TRKCELL	YES NO specifies whether or not track-cell despooling is to be used with this FSA. You specify the size of the track cell, in terms of buffers, in the TRKCELL parameter of the JES2 SPOOLDEF statement. For improved performance, IBM recommends that you specify TRKCELL=YES on this statement and TRKCELL=5 on the JES2 SPOOLDEF statement. The default is NO.
WS	Specifies the work-selection criteria for this FSA; separate the values with a comma. For an IP PrintWay FSA, consider the following values:
CL Q	Specifies that the FSA selects only those data sets with the class specified in the CLASS parameter of this statement.
R	Specifies that the FSA selects only those data sets with the destination specified in the ROUTECDE parameter of this statement.
F	Specifies that the FSA selects only those data sets with the form name specified in the FORMS parameter of this statement. This parameter is optional but recommended. If you omit it, the FSA selects output data sets for processing according to default work-selection criteria.

Note: If you create an FSA definition in the Printer Inventory for this FSA, you must specify this FSA name in the FSA definition. Although JES2 might let you use different formats to specify the FSA name in this statement, for example, PRT(nnnnn), *always* specify the FSA name as PRTnnnnn in the FSA definition. See “FSS definitions” on page 78.

4.3.2 Defining the IP PrintWay functional subsystem in JES3

JES3 FSS definitions can only be done with JES3 initialization statements. Once the initialization deck is updated, JES3 global should be restarted with the *hot start with refresh* initialization option. The hot start with refresh processing reads the initialization stream. FSSDEF and DEVICE statements belong to the subset of the initialization statements that are processed during a hot start with refresh.

To define a functional subsystem for IP PrintWay in the JES3 environment, the JES3 initialization deck needs to be updated with:

- FSSDEF statements to define each IP PrintWay functional subsystem (FSS)

- **DEVICE** statements for each functional subsystem application (FSA) under the control of the FSS

JES3 also provides operator commands for dynamic FSSDEF and DEVICE statement definitions.

On the JES3 SYSOUT statements, specify TRUNC=NO if you do not want JES3 to truncate trailing blanks for data sets in output classes that IP PrintWay processes. The setting of this parameter can affect the formatting and transformation of the printed output. The default is the value set in the TRUNC parameter of the JES3 BUFFER statement or TRUNC=YES.

Figure 4-5 shows sample JES3 statements for one FSS named PRINTWAY and one associated FSA named PRT123.

```
FSSDEF,FSSNAME=PRINTWAY,PNAME=ANFWPROC,TYPE=WTR
DEVICE,DTYPE=PRTAFP1,FSSNAME=PRINTWAY,JNAME=PRT123,JUNIT=(,SYS1,UR,ON),
MODE=FSS,WC=(J),WS=(CL,F)
```

Figure 4-5 Sample JES3 definitions for an FSS and one associated FSA

FSSDEF statements

The following section describes the JES3 initialization statements shown in Figure 4-5, including some additional parameters that you might want to specify for an IP PrintWay FSS and FSA. For a complete list of parameters and for detailed descriptions of each of the parameters, refer to *OS/390 JES3 Initialization and Tuning Reference*, SA22-7550.

The FSSDEF statement parameters are:

FSSNAME	Specifies the name of this FSS. Each FSS must have a unique 1- to 8-character name. This FSS name must match the FSS name in the DEVICE statement for each associated FSA. This parameter is required.
MSGDEST	Specifies the console destination class for messages concerning this FSS. This parameter is optional; if you omit it, JES sets a default.
PNAME	Specifies the name of the procedure for starting this FSS. Different FSSDEF statements can refer to the same startup procedure. This parameter is required.
SYSTEM	Specifies the JES3 system name on which the FSS will run. This parameter is optional; if you omit it, JES determines the default from the DEVICE statement.
TERM	YES NO specifies that the FSS will terminate if the JES3 global address space is terminated by a *RETURN or *DUMP operator command. This parameter is optional. The default is NO.
TYPE=WTR	Specifies that the FSS is an output writer. This parameter is required.

DEVICE statement

The DEVICE statement defines each FSA to JES. This section describes DEVICE parameters that you might want to specify for an IP PrintWay FSA.

DGROUP	Specifies a 1- to 8-character destination processed by this FSA. If you designate destination as a work-selection criterion on the WS parameter, this FSA selects data sets that match the value specified. This parameter is optional. Specify it only if you specify destination as a work-selection criteria on the WS parameter.
---------------	--

DTYPE	Specify PRTAFP1 as the device type for an IP PrintWay FSA. This parameter is required.								
DYNAMIC	YES/NO specifies whether JES3 starts and stops the FSA dynamically.								
FSSNAME	Specifies a unique FSS associated with this FSA. This value must match the value coded for the FSSNAME parameter in the corresponding FSSDEF statement. This parameter is optional. The default is the name of this FSA, as specified in the JNAME parameter.								
JNAME	Specifies the unique 1- to 8-character name of this FSA. This parameter is required.								
JUNIT	(devnum,main,msgdest,ON OFF) <table> <tr> <td>devnum</td><td>For IP PrintWay, do not specify a device address; use a comma as a position holder.</td></tr> <tr> <td>main</td><td>Name of the processor on which IP PrintWay is running.</td></tr> <tr> <td>msgdest</td><td>Destination for messages about the FSA.</td></tr> <tr> <td>ON OFF</td><td>Specifies whether the FSA is initially online or offline. This parameter is required.</td></tr> </table>	devnum	For IP PrintWay, do not specify a device address; use a comma as a position holder.	main	Name of the processor on which IP PrintWay is running.	msgdest	Destination for messages about the FSA.	ON OFF	Specifies whether the FSA is initially online or offline. This parameter is required.
devnum	For IP PrintWay, do not specify a device address; use a comma as a position holder.								
main	Name of the processor on which IP PrintWay is running.								
msgdest	Destination for messages about the FSA.								
ON OFF	Specifies whether the FSA is initially online or offline. This parameter is required.								
MODE	MODE=FSS specifies that this FSA is managed by an FSS. This parameter is required.								
WC	WC=(classes) specifies the output classes processed by this FSA. List all classes to be selected by this FSA; separate them with a comma. If you designate CLASS as a work-selection criterion on the WS parameter, this FSA selects data sets that match the values specified here. This parameter is optional. If you omit it, this FSA selects data sets with any output class.								
WS	WS=(work_selection_criteria) specifies the work-selection criteria for this FSA; separate each value with a comma. For an IP PrintWay FSA, consider the following values: <table> <tr> <td>C</td><td>Causes JES3 to pass the FCB attribute of the print data set to the FSA. If you do not specify FCB as a work-selection criterion, JES3 does not pass the FCB to the IP PrintWay FSA.</td></tr> <tr> <td>CL</td><td>Specifies that the FSA selects only those data sets with the class specified in the WC parameter of this statement.</td></tr> <tr> <td>D</td><td>Specifies that the FSA selects only those data sets with the destination in the DGROUP parameter of this statement.</td></tr> <tr> <td>F</td><td>Requests that JES3 pass the form attribute of the print data set to the FSA. If you do not specify forms as a work-selection criterion, JES3 does not pass the form to the IP PrintWay FSA.</td></tr> </table>	C	Causes JES3 to pass the FCB attribute of the print data set to the FSA. If you do not specify FCB as a work-selection criterion, JES3 does not pass the FCB to the IP PrintWay FSA.	CL	Specifies that the FSA selects only those data sets with the class specified in the WC parameter of this statement.	D	Specifies that the FSA selects only those data sets with the destination in the DGROUP parameter of this statement.	F	Requests that JES3 pass the form attribute of the print data set to the FSA. If you do not specify forms as a work-selection criterion, JES3 does not pass the form to the IP PrintWay FSA.
C	Causes JES3 to pass the FCB attribute of the print data set to the FSA. If you do not specify FCB as a work-selection criterion, JES3 does not pass the FCB to the IP PrintWay FSA.								
CL	Specifies that the FSA selects only those data sets with the class specified in the WC parameter of this statement.								
D	Specifies that the FSA selects only those data sets with the destination in the DGROUP parameter of this statement.								
F	Requests that JES3 pass the form attribute of the print data set to the FSA. If you do not specify forms as a work-selection criterion, JES3 does not pass the form to the IP PrintWay FSA.								

4.3.3 IP Printway FSA work-selection criteria

You define work-selection criteria for each IP PrintWay FSA in the JES initialization statements. These criteria determine which output data sets each FSA selects from the JES spool. The work-selection criteria is specified on the WS parameter of one of the following:

- ▶ JES2 PRTnnnnn initialization statement
- ▶ JES3 DEVICE initialization statement

JES2 and JES3 let you specify a variety of work-selection criteria, which correspond to JCL parameters. Some work-selection criteria that you might consider for data sets to be processed by IP PrintWay are:

- ▶ SYSOUT class
- ▶ Form name
- ▶ Destination name

Work-selection criteria considerations

Do not specify destination as a work-selection criterion if your installation plans to specify the host name or IP address in the DEST=IP parameter on the OUTPUT JCL statements or on dynamically defined output descriptors. JES does not use the host name or IP address when determining whether an output data set meets the work-selection criteria defined for an FSA.

JES3 considerations

In a JES3 environment, specify *form* as a work-selection criterion if your installation wants IP PrintWay to use the form name on OUTPUT JCL statements or on dynamically defined output descriptors to select any printer definitions in the Printer Inventory. JES3 passes the form name data set attribute to the IP PrintWay FSA only if form is a JES3 work-selection criterion. Although you specify form as a work-selection criterion, the IP PrintWay FSA selects jobs with all form names for processing.

However, if your IP Printway FSA's JES3 DEVICE initialization statement specifies DYNAMIC=YES and its WS criterion includes form, JES3 will not start the FSA dynamically unless a print data set's form attribute (and all other attributes) match the FSA's work selection criterion.

In a JES3 environment, specify FCB as a work-selection criterion if you want IP PrintWay to perform FCB processing. JES3 passes the FCB name specified as an OUTPUT JCL parameter or as dynamically defined output descriptor to the PrintWay FSA only when FCB is a JES work-selection criterion.

JES initialization statement examples

The JES2 initialization statements required to start the three FSAs, shown in Figure 4-6, are as follows:

```
FSSDEF (PRTWAY1) PROC=PRINTWA2,   HASPFSSM=HASPFSSM
*-----*
PRT1 FSS=PRINTWAY, MODE=FSS,
      CLASS=J,
      START=NO, WS=(Q/)

PRT2 FSS=PRINTWAY, MODE=FSS,
      CLASS=J,
      START=NO, WS=(Q/)

PRT3 FSS=PRINTWAY, MODE=FSS,
      CLASS=J,
      START=NO, WS=(Q/)
```

Figure 4-6 JES2 initialization statements for the FSA devices

When you define the PRINTER statement, you determine which FSS that device starts the FSA in.

The JES3 initialization statements required to start the three FSAs, shown in Figure 4-7, are as follows:

```
*-----*
FSSDEF,TYPE=WTR,FSSNAME=PRTWAY1,PNAME=PRINTW43,TERM=YES
*-----*
DEVICE,DTYPE=PRTWAY,
  JNAME=PRT1,DGROUP=PRTWAY,
  JUNIT=(,SC50,UR,,SC49,UR,,SC43,UR),
  FSSNAME=PRINTWAY,
  WC=(J),WS=(CL)
*-----*
DEVICE,DTYPE=PRTWAY,
  JNAME=PRT2,DGROUP=PRTWAY,
  JUNIT=(,SC50,UR,,SC49,UR,,SC43,UR),
  FSSNAME=PRINTWAY,
  WC=(J),WS=(CL)
*-----*
DEVICE,DTYPE=PRTWAY,
  JNAME=PRT3,DGROUP=PRTWAY,
  JUNIT=(,SC50,UR,,SC49,UR,,SC43,UR),
  FSSNAME=PRINTWAY,
  WC=(J),WS=(CL)
*-----*
```

Figure 4-7 JES3 initialization statements for the FSA devices

When you define the DEVICE statement, you determine which FSS that device starts the FSA in.

4.3.4 APF-authorize SYS1.IMAGELIB

If you plan to use FCB support in IP PrintWay, you must link-list and APF-authorize the SYS1.IMAGELIB data set. To temporarily authorize this data set, use the SETPROG APF system command. To permanently authorize this data set:

- ▶ Update the PROGxx member of SYS1.PARMLIB to include the data sets in the LNKLIST statement.
- ▶ If the LNKAUTH parameter that is placed in the IEASYSxx member of SYS1.PARMLIB does not specify LNKLIST, then add the data sets to the APF statement in the PROGxx member.

As an alternative to updating the PROGxx member, you can update the LNKLISTxx and IEAAPFxx members of SYS1.PARMLIB; however, IBM recommends updating the PROGxx member. For more information, refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

4.3.5 Allocate and initialize the transmission queue data set

The transmission queue data set is a required data set that contains an entry for each data set that IP PrintWay is processing. IP PrintWay maintains the entries in the transmission queue data set. You can use the Infoprint Server ISPF panels to browse, modify, and delete entries, but you cannot add entries. For information about how to use the Infoprint Server ISPF panels to view the transmission queue data set, refer to *z/OS Infoprint Server Operation and Administration*, S544-5744.

You should allocate one shared transmission queue data set for all IP PrintWay functional subsystems (FSSs) running at the same software level (V2R8 or later). To share the transmission queue data set between all FSSs is not a must requirement; you can create a separate data set for each IP PrintWay FSS. If more than one FSS can transmit output data sets to the same set of printers, IBM recommends sharing the transmission queue data set so that data sets are printed in the correct order.

The IP PrintWay transmission queue data set is a VSAM key-sequenced data set. Use the DFSMS IDCAMS program to define the transmission queue. Figure 4-8 shows sample attributes for the transmission queue DEFINE CLUSTER command.

Refer to *OS/390 DFSMS Access Method Services for Catalogs*, SC26-7394 for more information about allocating VSAM data sets.

```
DEFINE CLUSTER
      (NAME(INFOPRNT.SC43.QUEUE) -
       VOLUMES(volser) -
       INDEXED -
       SPEED -
       SHAREOPTIONS(4 3)) -
      DATA -
      (NAME(INFOPRNT.SC43.QUEUE.DATA) -
       TRACKS(30 5) -
       KEYS(20 0) -
       RECORDSIZE(238 2048) -
       FREESPACE(10,10) -
       CISZ(4096)) -
      INDEX -
      (NAME(INFOPRNT.SC43.QUEUE.INDEX))
```

Figure 4-8 Model DEFINE CLUSTER command for IP PrintWay transmission queue

The IP PrintWay default data set name is ANF.QUEUE. You can use any name for the IP PrintWay transmission queue data set. If you change the name for the transmission queue, as shown in Figure 4-8 on page 104, you could update the ANFIPQS panel in the AOP.SAOPPENU data set to initially display the correct transmission queue name, as shown in Figure 4-9 on page 106.

Note: Infoprint Server provides sample JCL in SYS1.SAMPLIB(ANFDEAL) to delete and redefine the IP PrintWay transmission queue data set.

Initialize the transmission queue data set

After allocating space for the transmission queue data set, use the sample JCL in SYS1.SAMPLIB(ANFQINIT) to initialize the data set with binary zeroes. If you did not use the default name, ANF.QUEUE, change the name of the data set in the sample JCL.

transmission queue data set considerations

When allocating the transmission queue data set, consider the following:

- ▶ Change the data set name to comply with your installation standards. Replace volser with a correct volume serial if the data set will be on a non-SMS-managed volume. VOLUMES can be specified or modeled for a data set that is to be SMS-managed or, instead of the VOLUMES, you can use the SMS construct name STORAGECLASS(class).
- ▶ Use the values for the RECORDSIZE, SHAREOPTIONS, and KEYS options shown in the model definition.

- Adjust the value of the FREESPACE option based on experience with the data set. The size of the PrintWay transmission queue data set changes dynamically, according to the number of data sets retained after successful or unsuccessful transmission.

Note: After allocating space for the transmission queue data set, use the sample JCL in SYS1.SAMPLIB(ANFQINIT) to initialize the data set with binary zeroes.

Restricting access to the transmission queue data set

All ISPF users who have a valid OMVS segment defined in the System Authorization Facility (SAF) user profile, can use the Infoprint Server ISPF panels to update the IP PrintWay transmission queue data set. You can, however, set up an SAF data set profile, for example using Resource Access Control Facility (RACF), to restrict access to the data set.

```
ADDSD 'INFOPRNT.**' UACC(NONE) GENERIC
```

The data sets INFOPRNT.** that are being protected are:

INFOPRNT.SC43.QUEUEAs shown in Figure 4-8 on page 104 and Figure 4-9 on page 106

INFOPRNT.SC43.MSGFILEAs shown in Figure 3-3 on page 63

If you have SAF-protected the transmission queue data set, the Infoprint Server ISPF panels display only the actions that a user is allowed to perform. For example, if a user has READ access to the transmission queue data set, that user is allowed only to browse entries in the data set and is not allowed to modify or delete entries. If the access is NONE, the user is not allowed to browse in the data set. To create an access list for users, do the following:

```
PERMIT 'INFOPRNT.**' ACC(UPDATE) ID(AOPADMIN AOPOPER)
PERMIT 'INFOPRNT.**' ACC(READ) ID(VAINI ROGERS)
```

Viewing the transmission queue data set

An administrator or operator can view data sets that are being processed by IP PrintWay. The queue can be entered by selecting Option 5 on the Primary Menu, shown in Figure 3-2 on page 61. The panel, shown in Figure 4-9, is displayed when you select this option.

When a data set is selected to be printed, IP PrintWay creates an entry in the transmission queue data set. Using this panel, you specify the display criteria to select the queue entries you wish to see.

In Figure 4-9, the Printer name (poke) and the CLASS (J) were entered to display all data sets with that criteria.

```

ANFIPQS                               IP PrintWay Transmission Queue Selection
Command ===> _____

Enter optional display criteria below:

Printer Name:           ===> poke_____
or
DCF Routing Key:
  DEST                 ===> _____
  CLASS                ===> J_____
  FORMS                ===> _____

Jobname                 ===> _____
Arrival Date           ===> _____ (YYYY/MM/DD)
Arrival Time           ===> _____ (HH:MM:SS)
Transmission Status    ===> _____ (One or more of: A S F R H Z)
Port Number            ===> _____ (100 to 65535)
Print Q Name/Logmode    ===> _____

Host Name/IP Address/
URL/LU Name            ===> _____

Queue Data Set name    ===> INFOPRNT.SC43.QUEUE_____

Change Sort Criteria ===> _

```

Figure 4-9 IP PrintWay Transmission Queue selection panel

After the selection criteria is entered, and you press Enter, the data sets are displayed. You can use the action characters shown to browse, edit, hold, reset, and delete entries, as shown in Figure 4-10.

```

ANFIPQL                               IP PrintWay Transmission Queue           Row 1 to 2 of 2
Command ===> _____                               Scroll ===> CSR
Actions: B-Browse D-Delete E-Edit H-Hold R-Reset --repeat
Status : Z-New R-Retry H-Held F-Failed S-Successful A-Active
Change Sort Criteria ===> _____ (/ or blank)

A Route Name          T S Jobname  Arrival Date/Time  Host/IP/URL/LUName
_ -poke               P A BPXAS   2001/05/30 15:45:09 9.12.2.4
_ -poke               P Z BPXAS   2001/05/30 15:45:09 9.12.2.4
***** Bottom of data *****

```

Figure 4-10 IP PrintWay Transmission Queue with an active print data set

Browse the transmission queue entry

If you need to know information about the current entry on the transmission queue, place a **b** next to the entry and hit Enter. The panel shown in Figure 4-11 on page 107 is displayed.

```

ANFIPQR ----- IP PrintWay Transmission Queue Entry -----
Command ==> _____

More:      +

Route Name      .... -poke
Route Name Type .... P (Printer Name)
Jobname         .... BPXAS
Job ID          .... PS067203
Data Set Name    .... WTSCPLX9.ROGERS.ROGERS.JOB14986.D000000E.TEST#JCL
Transmission Status .... Z (Queued for first attempt)
Last Activity : Date      .... 0000/00/00   Time      .... 00:00:00
Next Activity : Date      .... 2001/05/30   Time      .... 17:57:15
Retry          : Time      .... 0000:00:10   Retries     .... 0      of 1
Retain Time    : Success   .... 0000:00:00   Failure     .... 0000:01:00

Transmission Protocol .... 1  1. LPR  2. Direct Sockets  3. IPP  4. VTAM

Host/IP Addr/URL/LUName .... 9.12.2.4

Print Queue/Logmode .... afccu2

Port Number      ....                  (100 to 65535)
Response timeout ....                  Number of seconds to wait

```

Figure 4-11 Browse a transmission queue data set entry

4.3.6 Allocating and initializing the IP PrintWay message log data set

IP PrintWay writes messages that track data set transmissions in an IP PrintWay message log data set. This is a required data set. The message log is organized as a circular queue. When a message extends to the end of the data set, the next message starts at the beginning. The size of the message data set remains constant. The oldest messages are automatically overlaid by newer messages.

You can create one message log data set to be shared by all IP PrintWay functional subsystems (FSSs) or you can create separate data sets for each FSS. To allocate space for the data set and initialize it, use the sample JCL in SYS1.SAMPLIB(ANFMIJCL) to execute the ANFMFILE initialization program, as shown in Figure 4-12.

```

//MSGINIT JOB ...
//*-----*//
/* INITIALIZE THE IP PRINTWAY MESSAGE QUEUE FILE.          *//
//*-----*//
//REXXJCL EXEC PGM=IRXJCL,PARM='ANFMFILE'
//SYSTSPRT DD SYSOUT=*
//SYSEXEC DD DSN=AOP.SAOPEXEC,DISP=SHR
//MSGDD DD DSN=INFOPRNT.SC43.MSGFILE,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,1),
//          DCB=(DSORG=PSU,RECFM=FB,LRECL=080,BLKSIZE=0)
/*

```

Figure 4-12 Sample JCL to allocate and initialize the message queue data set

The message log data set has the following characteristics:

- Physical sequential unmovable organization (PSU)
- Fixed block format (FBA)

- ▶ Logical record length (LRECL) of 57 at a minimum; 80 is recommended
- ▶ Block size (BLKSIZE) of about 10 records per block; 800 is recommended

The ANFMFILE program formats all available space.

Note: A successful execution of the ANFMFILE program ends with a system abend code D37.

All ISPF users can use the ISPF browse and view to access the IP PrintWay message log data. You should set up an SAF data set profile to restrict access to the data set.

Refer to *z/OS Infoprint Server Operation and Administration*, S544-5744 for more information.

Message log file

IP PrintWay issues a message to the IP PrintWay message log data set when:

- ▶ A data set is received from JES.
- ▶ It is successfully or unsuccessfully transmitted to the target system.
- ▶ It is released to JES.

The data set may also contain other IP PrintWay messages, including messages written by IP PrintWay installation exits.

An administrator or operator can view data sets that are being processed by IP PrintWay. The queue can be entered by selecting Option 6 on the Primary Menu, shown in Figure 3-2 on page 61. The panel, shown in Figure 4-14 on page 109, is displayed when you select this option.

The message log is organized as a circular queue. When a message extends to the end of the data set, the next message starts at the beginning. In this way, the size of the message data set remains stable, with old messages automatically overlaid by newer messages.

The following line indicates the current position in the data set with the previous line being the last message written to the log.

```
007593 2001032207351436 <<<<=====>>>>
```

Figure 4-13 Current position in message log data set for next message


```

File Edit Edit_Settings Menu Utilities Compilers Test Help
ISREDDE2 INFOPRNT.SC43.MSGFILE Columns 00001 00072
Command ==> Scroll ==> HALF
007567 IAZFSS
007568 2001032007544008-ANFM604I Data set:
007569 WTSCPLX9.VAINI.VAINI.JOB12988.D0000039.CLISTHAA
007570 The data set is being released to JES
007571 2001032207232853-ANFM700I Data set:
007572 WTSCPLX9.VAINI.VAINI.JOB12988.D000003B.#ONELINE
007573 The data set has been acquired by PRTWAY
007574 2001032207233432-ANFM601I Data set:
007575 WTSCPLX9.VAINI.VAINI.JOB12988.D000003B.#ONELINE
007576 The data set was successfully transmitted to host and q
007577 9.12.2.4
007578 afccu2
007579 2001032207233434-ANFM604I Data set:
007580 WTSCPLX9.VAINI.VAINI.JOB12988.D000003B.#ONELINE
007581 The data set is being released to JES
007582 2001032207351332-ANFM700I Data set:
007583 WTSCPLX9.VAINI.VAINI.JOB12988.D000003C.#ONELINE
007584 The data set has been acquired by PRTWAY
007585 2001032207351433-ANFM601I Data set:
007586 WTSCPLX9.VAINI.VAINI.JOB12988.D000003C.#ONELINE
007587 The data set was successfully transmitted to host and q
007588 9.12.2.4

PF03 set to CANCEL - exit View. PF06 set to LREFRESH - refresh View.
KEYLIST is set OFF.

007593 2001032207351436 <<<<=====>>>>

```

Figure 4-14 IP PrintWay Message log data set

4.3.7 Create an IP PrintWay startup procedure

Before starting an IP PrintWay FSS, you must prepare the startup JCL procedure. The same startup procedure can be used to start several IP PrintWay FSSs if all the IP PrintWay FSSs share the same transmission queue data set. You specify the name of the startup procedure in the JES initialization statement for each FSS.

If you change the IP PrintWay startup procedure, you must stop all IP PrintWay FSAs within the FSS that uses this startup procedure and then restart the FSAs to pick up the changes. If JES2 does not stop the FSS automatically, you also need to stop the FSS before restarting the FSAs.

Refer to *OS/390 Infoprint Server Operation and Administration*, S544-5744 for information about how to stop and start IP PrintWay.

IP PrintWay provides a sample startup procedure in SYS1.IBM.PROCLIB(ANFWPROC), which is shown in Figure 4-15 on page 110.

```

//ANFWPROC PROC HLQ=ANF,
//      TCPHLQ=TCPIP
//* -----
//* this is a sample procedure for starting IP PrintWay
//* -----
//IEFPROC EXEC PGM=ANFFIEP,REGION=1M,TIME=NOLIMIT,PARM='INV=AOP1'
//* -----
//* AOP1 is the default name of the printer inventory
//* server. This must be modified if this is not the
//* name being used for the inventory server
//* -----
//ANFQUEUE DD DSN=&HLQ..QUEUE,DISP=SHR
//SYSTCPD DD DSN=&TCPHLQ..TCPIP.DATA,DISP=SHR
//ANFMMSG DD DSN=&HLQ..MSGFILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*

```

Figure 4-15 IP PrintWay sample startup procedure

IP PrintWay procedure considerations

When creating an IP PrintWay JCL procedure, consider the following:

- ▶ If you use an IP PrintWay exit, add a STEPLIB statement to identify the library containing the exit unless the library is part of the standard OS/390 search order.
- ▶ If you did not add the Language Environment run-time library, CEE.SCEERUN, to the system LNKLIST, add the SCEERUN data set to the STEPLIB DD statement.
- ▶ If you plan to use the IPP protocol to transmit data to any printers and did not add the C++ run-time library, CBC.SCLBDLL, to the system LNKLIST, add the SCLBDLL data set to the STEPLIB DD statement.
- ▶ If you plan to use the IPP protocol to transmit data to any printers, you might need to add the STDENV DD statement to the startup procedure.

Note: The PARM='INV=inventory' on the EXEC statement refers to the Printer Inventory *name* that is assigned in the **aopd.conf** configuration file. Specify AOP1 (the default name) if you have not created an **aopd.conf** configuration file or if the inventory attribute is omitted. This name is case sensitive. This parameter is required.

IP PrintWay procedure DD statements

The IP PrintWay JCL procedure DD statements are as follows:

- //ANFQUEUE** The IP PrintWay transmission queue data set. This statement is required. To ensure that data sets are always transmitted in the same order in which IP PrintWay acquires them from the JES spool, share the same transmission queue data set between each IP PrintWay FSS.
- //SYSTCPD** The TCPIP.DATA data set. This statement is optional.
- //ANFMMSG** The IP PrintWay message log data set. This statement is required.

STDENV DD statement

This DD statement is optional, but should be considered if the defaults for the environment variables are not adequate.

- //STDENV** Specifies an MVS variable blocked (VB) record format data set or an HFS file that contains z/OS UNIX environment variables. This DD statement is used only

if IP PrintWay uses the Internet Printing Protocol (IPP) to transmit data to remote printers. The environment variables that IP PrintWay might use are:

- LANG** Defines the language in which you want messages. This variable is required if the NLSPATH variable contains %L and you do not want to receive messages in English.
- LIBPATH** Defines the directory where Infoprint Server DLLs are located. If you installed Infoprint Server DLLs in the default directory, /usr/lpp/Printsrv/lib, you do not need to specify this variable.
- NLSPATH** Defines where Infoprint Server and OS/390 UNIX message catalogs can be found. Specify the same catalogs as you specify in the OS/390 UNIX NLSPATH environment variable in the /etc/profile file.
- PATH** Defines the directory where Infoprint Server executables are located. If you installed Infoprint Server executables in the default directory, /usr/lpp/Printsrv/bin, you do not need to specify this variable.

_BPXK_SETIBMOPT_TRANSPORT

Specifies the TCP/IP job name of the TCP/IP stack you want the IP client in IP PrintWay to use. See also “Starting Infoprint daemons from /etc/rc” on page 49. If you do not specify this variable, TCP/IP uses its standard search mechanism to determine the TCP/IP stack to use. For example, if the TCP/IP job name is TCPIPOE, specify:

_BPXK_SETIBMOPT_TRANSPORT=TCPIPOE

The TCP/IP job name for an IP PrintWay FSS can also be defined on the IP PrintWay FSS definition panel. Refer to Figure 3-20 on page 80.

Note: If you omit the STDENV DD statement or do not specify all of the environment variables, the following default values are used:

```
LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib
NLSPATH=/usr/lpp/Printsrv/%L/%N:/usr/lib/nls/msg/%L/%N
PATH=/usr/lpp/Printsrv/bin:/bin:/usr/bin
```

4.3.8 Setting IP PrintWay LE runtime options

The IP PrintWay runtime options for the LE environment are shown on Figure 4-16. If you want to change the LE runtime options, create a CEEUOPT object file as described in *z/OS Language Environment Programming Guide*, SA22-7561. Then create an SMP/E usermod to link-edit the object file with ANFICONV.

```
POSIX(ON)
ALL31(ON)
STACK(8K,64K,ANY,KEEP)
HEAP(8K,8K,,FREE)
ANYHEAP(8K,8K)
THREADHEAP(8,4K,ANY,FREE)
BELOWHEAP(8,4K,FREE)
LIBSTACK(8,0)
ENVAR(''_CEE_ENVFILE=//DD:STDENV'')
MSGFILE(,,,ENQ)
```

Figure 4-16 LE runtime options for IP PrintWay

4.4 IP PrintWay sample exits

IP PrintWay provides the following sample exits written in assembler language in SYS1.SAMPLIB:

ANFUXRTG	Changes the IP address of the target system.
ANFUXRT1	Changes the name of the components that contain transmission options.
ANFUXRT2	Changes the URL.
ANFUXRT3	Changes the IP address and print queue name.
ANFUXRT4	Changes the printer definition name to DEST, CLASS, and FORMS values.
ANFUXBD1	Adds a PCL5 printer setup string to the beginning of a data set.
ANFUXBD2	Adds a separator page containing line data to the beginning of a data set. ANFUXBD2 uses the Block Letter program (ANFUBLK) to create block letters.
ANFUXBD3	Requests printing a banner page before the first data set in a job but not before subsequent data sets in the job.
ANFUXED1	Adds a separator page containing line data to the end of a data set. ANFUXED1 uses the Block Letter program (ANFUBLK) to create block letters.
ANFUXRC1	Writes lines in a boustrophedon manner; that is, it writes alternate lines in opposite directions, from left to right and from right to left.
ANFUXSMF	Suppresses the SMF record for the JESNEWS data set.
ANFUXMSG	Suppresses messages ANFM700I and ANFM604I. See also Appendix A, “Dynamic IP PrintWay message exit sample” on page 243 for an example of an ANFUXMSG exit that writes all IP PrintWay input messages to the hardcopy log. This sample exit exploits the MVS dynamic exit facility services.
ANFUXRSP	Takes action according to the response notification code.

4.4.1 Installing exits

To install IP PrintWay exits, compile and link-edit an exit as follows:

- ▶ Name the exit:
 - For the Routing exit, use the name ANFUXRTG.
 - For the SMF exit, use the name ANFUXSMF.
 - For the Message exit, use the name ANFUXMSG.
 - For the Response Notification Exit, use the name ANFUXRSP.
 - For the Begin Data Set exit, the Record Exit, and the End Data Set exit, use any program name allowed by your system. Specify the exit name in one or more option entries in the printer definition.
- ▶ Link-edit the exit with the RENT and AMODE 31 attributes into an APF-authorized library.
- ▶ If the Begin Data Set or End Data Set exit program calls the Block Letter program (ANFUBLK), use the sample JCL in SYS1.SAMPLIB(ANFULINK) to link-edit the exit with the Block Letter program (ANFUBLK) and the tables (ANFUBTBL).

Identify the library containing an exit in one of the following:

- ▶ A STEPLIB statement in the IP PrintWay startup procedure
You can provide a different exit program for each IP PrintWay FSS by using a unique STEPLIB for each startup procedure.

- A library concatenated to LNKLIST

If you place the exit in a LNKLIST library, then you must use the same exit program for every IP PrintWay startup procedure.

Modifying exits:

To modify any exit, replace the exit and then restart the IP PrintWay FSS.

4.4.2 Activating IP PrintWay exits

“Sample ANFUXBD2 begin data set exit” on page 252 is provided as an example of modifications to the exit ANFUXBD7 which is available at the following Web site:

<http://www.printers.ibm.com/R5PSC.NSF/Web/tools>

To activate this exit for a specific printer using the ISPF panels, place the exit name on the panel, as shown in Figure 4-17, in the Begin data set field.

```

AOPIPDIO                                IP PrintWay Options
Command ==> _____

Printer definition name . NP24 _____

Retention period:                                     More:  +
  Successful. . . . 0000:01:00   Failure . . 0000:01:00
  Retry time . . . . 0000:00:10
  Retry limit. . . . 1 _____

Response timeout . . _____
Exits:
  Begin data set. . ANFUXBD7   End data set. . _____   Record. . _____
  Document header . . <ESC>E<ESC><ESC>811S<ESC> _____ (extend)
    / Translate document header
  Document trailer . . <ESC>E<ESC> _____ (extend)
    / Translate document trailer
  Dataset grouping . . 2  1. None  2. Job  3. Concatenate job

Formatting:
  Line termination. . . . 0D25 _____
  Transparent data char . 35 _____
  Carriage control type . -  1. None  2. Machine  3. ANSI
  Delete form feed. . . . 1  1. None  2. Leading  3. Trailing  4. Both
  Formatting. . . . . 2  1. None  2. Standard
                        3. Translate only  4. Use FCB
  PostScript header . . . -  1. Add  2. Ignore
                        3. Landscape  4. Always landscape
  _ Omit line termination at EOF

```

Figure 4-17 Panel to enter separator page exit name

4.5 IP PrintWay port ranges

In OS/390 releases before Release 8, the IP PrintWay LPR, by default, restricted itself to OS/390 ports 721 through 731. This meant that only eleven transmissions could occur simultaneously. The administrator could specify the USERPORTS LPR option to cause the IP PrintWay LPR to use any available port.

In the OS/390 Release 8 version of Infoprint Server, to increase the probability of finding a free port, the IP PrintWay LPR, by default, uses any available port. If you want to continue to use 721 through 731, the administrator can select the Restrict ports field in the printer definition to cause the IP PrintWay LPR to restrict itself to those ports.

In Figure 4-18, a slash (/) in the Restrict ports field indicates that the OS/390 port must be in the range of 721 to 731. Select this option when the remote LPD enforces this port restriction.

A blank indicates that any OS/390 port can be used.

Note: When this option is not selected, IP PrintWay can use any free port. This increases the probability of finding an available port. This indicator is used only when IP PrintWay uses the LPR transmission protocol; otherwise, it is ignored.

AOPIPDIL

LPR Protocol

Command ==>

Printer definition name . pokeps

Printer IP address . 9.12.2.4 (extend)

Print queue name . afccu2 (extend)

LPR Processing Options:

Mode 2

1. Control file first 2. Control file last

3. Stream 4. Remote PSF

Restrict ports

Print banner page

Banner class. .

Banner job name (extend)

Filename

Indent

Owner.

Print function . . . f

Title. (extend)

Width.

User options (extend)

Figure 4-18 LPR protocol panel

4.5.1 LPR to LPD considerations

Whether the LPR must use a source port in the range of 721 to 731. Some LPDs require that the source port be in this range, which is defined by RFC 1179. However, the Print Interface LPD *does not require* that the LPR be in this range. Generally, performance on the system where the LPR runs is improved if you do not restrict the range for the source port.

4.5.2 IP PrintWay procedure

To start any IP PrintWay FSS, you must create a cataloged startup procedure. If you require more than one FSS, the same startup procedure can be used.

Note: The operator cannot start this procedure. JES starts the FSS address space when the operator issues a START command for a printer that is defined to start an FSA in the FSS.

In the IP PrintWay procedure, there are three key data sets as follows:

- ▶ IP PrintWay Transmission queue
- ▶ IP PrintWay Message log file
- ▶ TCPDATA

TCPDATA

If you create a unique TCPIP.DATA data set for use by IP PrintWay, you can identify that data set in the IP PrintWay startup procedure. Use a //SYSTCPD DD statement to do this.

As shown in Figure 4-19, use the data set naming convention that includes the system name of the system where the IP PrintWay startup procedure executes.

```
//PRINTWY  PROC HLQU='INFOPRNT.SC43'
//*          HLQ=ANF,TCPHLQ=TCPIP
//* -----
//* THIS IS A SAMPLE PROCEDURE FOR STARTING IP PRINTWAY
//* -----
//IEFPROC  EXEC  PGM=ANFFIEP,REGION=3M,TIME=NOLIMIT,
//          PARM='INV=AOP1'
//ANFQUEUE DD DSN=&HLQU..QUEUE,DISP=SHR
//SYSTCPD  DD DSN=TCPIPOE.&SYSNAME..TCPPARMS(TCPDATA),DISP=SHR
//ANFMMSG  DD DSN=&HLQU..MSGFILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*SYSABEND DD SYSOUT=*
//*SYSUDUMP DD SYSOUT=*
//*SYSDUMP DD DSN=DUMP.PRINTWAY.MDUMP,DISP=SHR
//PRT1 CNTL
//PRT1 PRINTDEV TRACE=NO
//PRT1 ENDCNTL
//PRT2 CNTL
//PRT2 PRINTDEV TRACE=NO
//PRT2 ENDCNTL
//PRT3 CNTL
//PRT3 PRINTDEV TRACE=NO
//PRT3 ENDCNTL
```

Figure 4-19 IP PrintWay procedure in SYS1.PROCLIB

4.6 Printing to IP PrintWay printers with JCL

A printer definition is not required for a printer if the printer is controlled by IP PrintWay and the job submitter can specify the IP address of the target printer in the JCL that creates the output. The job submitter can specify the IP address and either print queue name or port number of a remote printer on an OUTPUT JCL statement, as shown in Figure 4-20. A job submitter can also specify the IP address and print queue name of a remote printer in an Infoprint Server job attribute.

```
//ROGERSM  JOB  (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//R1 OUTPUT DEST='IP:9.12.2.4',PRTQUEUE='afccu2',
//          RETAINF='01:00:00',
//          RETAINS='0',
//          RETRYL=4,
//          RETRYT='00:10:00',
//PRINT    EXEC  PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2   DD SYSOUT=J,OUTPUT=(*.R1)
//SYSUT1   DD DISP=SHR,DSN=ROGERS.JES.PS
//SYSIN    DD DUMMY
//
```

Figure 4-20 User JCL to print to an IP PrintWay printer

The keywords that apply to IP PrintWay options on the OUTPUT JCL statement are as follows:

- ▶ DEST='IP:'
- ▶ PRTOPTNS
- ▶ PRTQUEUE
- ▶ RETAINF | RETAINS
- ▶ RETRYL | RETRYT
- ▶ PORTNO
- ▶ FSSDATA

Note: See *z/OS MVS JCL Reference*, SA22-7597 for the details for these keywords.

When you define IP PrintWay printers in the Printer Inventory, the OUTPUT JCL statements keywords can be specified for the printer, as shown in Figure 4-21 and Figure 4-22 on page 117.

```

AOPIPDI0                                IP PrintWay Options
Command ==> _____

Printer definition name . pokeps_____

Retention period:
  Successful. . . . 0000:00:00   Failure . . 0000:01:00
Retry time . . . . 0000:00:10
Retry limit. . . . 1_____

Response timeout . . _____
Exits:
  Begin data set. . _____   End data set. . _____   Record. . _____

Document header . . _____ (extend)
  / Translate document header
Document trailer . . _____ (extend)
  / Translate document trailer
Dataset grouping . . 2 1. None 2. Job 3. Concatenate job

Formatting:
  Line termination. . . . 25_____
  Transparent data char . 35_____
  Carriage control type . _ 1. None 2. Machine 3. ANSI
  Delete form feed. . . . 1 1. None 2. Leading 3. Trailing 4. Both
  Formatting. . . . . 1 1. None 2. Standard
                        3. Translate only 4. Use FCB
  PostScript header . . . _ 1. Add 2. Ignore
                        3. Landscape 4. Always landscape
  _ Omit line termination at EOF

```

Figure 4-21 IP PrintWay Options panel

AOIPDIL		LPR Protocol	
Command ==> _____			
Printer definition name . <u>pokeps</u> _____			
Printer IP address . <u>9.12.2.4</u>		(extend)	
Print queue name . . <u>afccu2</u>		(extend)	
LPR Processing Options:			
Mode	<u>2</u>	1. Control file first	2. Control file last
		3. Stream	4. Remote PSF
/ Restrict ports			
_ Print banner page			
Banner class . .	_____		
Banner job name	_____ (extend)		
Filename	_____		
Indent	_____		
Owner	_____		
Print function . . .	<u>f</u>		
Title	_____ (extend)		
Width	_____		
User options	_____ (extend)		

Figure 4-22 IP PrintWay Protocol panel

4.6.1 Users using IP PrintWay printers

Therefore, there are three ways that a user may use IP PrintWay printers by submitting user JCL to print data sets on an IP printer. While the examples showing the JCL do not show all the JCL statements, they do show the ones necessary for IP PrintWay use.

JCL example 1

The submitting user has been notified with the following information to be able to use the IP PrintWay printer:

- ▶ Specify a specific SYSOUT class J because all the IP PrintWay FSAs select by class J.
- ▶ The IP address of the printer is 9.12.2.4.
- ▶ The queue name of the printer is afccu2.

The IP PrintWay definitions for the defined printer in the Printer Inventory are shown in Figure 3-7 on page 69 and Figure 3-13 on page 75.

In the JCL example in Figure 4-23, an OUTPUT statement is used to print the SYSPRINT data set on an IP PrintWay printer that is defined in the Printer Inventory.

```
//ROGERSM JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//PW OUTPUT DEST='IP:9.12.2.4',
// PRTOPTNS=POSTSCRIPT,
// PRTQUEUE='afccu2',
// RETAINF='01:00:00',
// RETAINS='01:00:00',
// RETRYL=4,
// RETRYT='00:10:00',
// .....
//SYSPRINT DD SYSOUT=J,OUTPUT=(*.PW)
// .....
```

Figure 4-23 User JCL to print a data set using an IP PrintWay printer

Note: The example shown in Figure 4-23 on page 117 does not use the Printer Inventory definition for the IP PrintWay printer.

JCL example 2

In Figure 4-24, the user specifies the Printer Inventory printer name on the SYSOUT DD statement. The only information the user is required to know is:

- ▶ The specific SYSOUT class J that the IP PrintWay FSAs select by
- ▶ The printer name specified in the Printer Inventory

```
//ROGERSM JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
.....
//SYSPRINT DD SYSOUT=J,DEST=POKEPS
.....
```

Figure 4-24 User submits a SYSOUT data set for IP PrintWay printer pokeps

Note: The example shown in Figure 4-24 uses the Printer Inventory definition for the IP PrintWay printer pokeps.

JCL example 3

IP PrintWay allows users to select a printer definition by specifying the name of the printer defined in the Printer Inventory using the FSSDATA parameter of an OUTPUT JCL statement, as shown in Figure 4-25. Because users can select a printer definition by name, administrators could eliminate specifying the DEST, CLASS, and FORMS routing criteria, shown by the arrow in Figure 4-26 on page 119, for users who use IP PrintWay using JCL jobs. Administrators and users can, however, continue to use DEST, CLASS, and FORMS routing criteria if desired.

FSSDATA parameter

If users submit jobs that specify the DEST=IP parameter on an OUTPUT JCL statement, as shown in Figure 4-23 on page 117, without also specifying the FSSDATA parameter on the OUTPUT JCL statement, you *must* create an IP PrintWay default printer definition to support such users. See “Define the default printer in Printer Inventory” on page 118.

```
//JOBFSSK JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//R1 OUTPUT FSSDATA='printer=INFO40'
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD SYSOUT=J
//SYSUT1 DD DISP=SHR,DSN=ROGERS.INFO.PS
//SYSIN DD DUMMY
//
```

Figure 4-25 JCL example using the FSSDATA parameter

4.6.2 Define the default printer in Printer Inventory

IP PrintWay uses the values in the default printer definition only when the OUTPUT JCL statement contains the DEST=IP parameter and the FSSDATA=printer parameter is omitted. If the FSSDATA parameter is also specified, IP PrintWay uses values in the printer definition specified in that parameter.

To create an IP PrintWay default printer definition, create an IP PrintWay printer definition, as shown in Example 4-26, and specify the attributes that apply to IP PrintWay as follows:

- Specify **DFLTNTRY** as the Printer Definition name, using uppercase characters.
- Do not select the field for Use DEST, CLASS, and FORMS for IP PrintWay printer selection.

AOPIPDC1 IP PrintWay Printer Definition		
Command ==> _____		
Printer definition name . <u>DFLTNTRY</u> ←		
Description . <u>IP: Defaults</u>		(extend)
Location . . . <u>Sysplex</u>		(extend)
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> *
Processing	=> _____	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> *
Protocol	=> _____	=> *
_ Use DEST, CLASS, and FORMS for IP PrintWay printer selection ←		
NetSpool LU name . _____ LU classes . . . _ _ _ _ _		(extend)

Figure 4-26 Create a default entry for IP PrintWay

Allocation options panel

For the allocation options, shown in Figure 4-27, make sure the CLASS is specified as a SYSOUT class that IP PrintWay FSAs are selecting by.

AOIPDAL Allocation	
Command ==> _____	
Printer definition name . <u>DFLTNTRY</u>	
More: +	
Spool allocation values:	
CLASS <u>J</u>	LINECT. _____
DEST. <u>DFLTNTRY</u>	PRMODE. _____
JES node. _____	PRTY. _____
FCB _____	SEGMENT _____
FLASH count _____	THRESHLD. _____
FLASH name. _____	UCS _____
FORMS <u>STD</u>	WRITER. _____
GROUPID _____	
USERDATA	
(extend)	
BURST <u>2</u> 1. Yes 2. No	
HOLD. <u>2</u> 1. Yes 2. No	
OUTDISP <u>5</u> 1. Purge 2. Leave 3. Keep 4. Hold 5. Write	
Values for Separator Pages:	
Address	
(extend)	
Building _____	
Department _____	
Name _____	
Room _____	
Title _____	
Resource Related Values:	
Form definition _____	
Character sets _____	

Figure 4-27 DFLTNTNTRY allocation options

Processing options panel

Do not select the Resubmit for filtering option. If you select this option, IP PrintWay ignores the IP address that the job submitter specifies in the DEST=IP parameter of the OUTPUT JCL statement and transmits the data set to the IP address specified in the printer definition.

AOPIPDPRR		Processing	
Command ==> _____			
Printer definition name . <u>DFLTNTRY</u>			
			More: +
Document code page . . _____			
Printer code page. . . <u>ISO8859-1</u>			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
<input checked="" type="checkbox"/> Line data			(extend)
<input type="checkbox"/> M0:DCA-P			(extend)
<input checked="" type="checkbox"/> PostScript			(extend)
<input checked="" type="checkbox"/> Text	<u>aopfiltr.so</u>		(extend)
<input checked="" type="checkbox"/> PCL			(extend)
<input type="checkbox"/> PDF			(extend)
<input type="checkbox"/> SAP			(extend)
<input checked="" type="checkbox"/> Other			(extend)
<input type="checkbox"/> Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top . . . _____		Bottom . . _____	
Line length . . . _____		Page length . . _____	
Tabs: Vertical . . _____		(extend)	
Horizontal . . _____		(extend)	
IP PrintWay Line-to-Text Conversion:			
<input checked="" type="checkbox"/> Pagination			
Margins: Top . . _____		Bottom . . _____	

Figure 4-28 DFLTENTRY entry processing options

IP PrintWay options

In the IP PrintWay default printer definition, you can specify default values for the following OUTPUT JCL parameters:

RETAINS, RETAINF, RETRYL, RETRYT, and PRTOPTNS.

AOIPDIO		IP PrintWay Options	
Command ==> _____			
Printer definition name . <u>DFLTNTRY</u> _____			
			More: +
Retention period:			
Successful	<u>0000:00:00</u>	Failure . .	<u>0000:01:00</u>
Retry time	<u>0000:00:10</u>		
Retry limit	<u>1</u>		
Response timeout . . _____			
Exits:			
Begin data set. .	_____	End data set. .	_____ Record. . _____
Document header . .	_____	(extend)	
/ Translate document header			
Document trailer . .	_____	(extend)	
/ Translate document trailer			
Dataset grouping . .	<u>2</u>	1. None 2. Job 3. Concatenate job	
Formatting:			
Line termination. . . .	<u>25</u>		
Transparent data char .	<u>35</u>		
Carriage control type .	<u>1</u>	1. None 2. Machine 3. ANSI	
Delete form feed. . . .	<u>1</u>	1. None 2. Leading 3. Trailing 4. Both	
Formatting.	<u>2</u>	1. None 2. Standard	
		3. Translate only 4. Use FCB	
PostScript header . . .	<u>1</u>	1. Add 2. Ignore	
		3. Landscape 4. Always landscape	
<u> </u> Omit line termination at EOF			

Figure 4-29 DFLTNTRY entry IP PrintWay options

DFLTNTRY protocol options

Select either LPR or Direct sockets; however, IP PrintWay can change the protocol based on the parameters specified on the OUTPUT JCL statement. IP PrintWay uses the LPR protocol if the PRTQUEUE parameter is specified, and uses the direct sockets protocol if the PORTNO parameter is specified.

For the Printer IP address, Print queue name, Port number, and URL fields: leave these fields blank because IP PrintWay uses the IP address and print-queue name or port number specified in JCL.

FSSDATA

When the FSSDATA parameter is specified, users can also specify the DEST=IP, PRTQUEUE, and PORTNO parameters on the OUTPUT JCL statement to override the IP address, print queue name, and port number in the selected printer definition. This support means that users can use the FSSDATA parameter to select a printer definition with the desired IP PrintWay options, but print to printers whose IP addresses are not defined in the Printer Inventory.

AOPIPDIL		LPR Protocol	
Command ==> _____			
Printer definition name . <u>DFLTNTRY</u> _____			
Printer IP address . _____		(extend)	
Print queue name . . _____		(extend)	
LPR Processing Options:			
Mode	<u>2</u>	1. Control file first	2. Control file last
		3. Stream	4. Remote PSF
<u>/</u> Restrict ports			
_ Print banner page			
Banner class. .	_____	(extend)	
Banner job name	_____	(extend)	
Filename	_____		
Indent	_____		
Owner.	_____		
Print function . . .	<u>f</u>		
Title.	_____	(extend)	
Width.	_____		
User options	_____	(extend)	

Figure 4-30 DFLTNTRY entry protocol options



Implementing NetSpool

NetSpool intercepts print data from VTAM applications, such as CICS and IMS. NetSpool converts the data into line data and creates output data sets on the JES2 or JES3 spool. JES or PSF for OS/390 can print the output data sets, or JES can transmit them to another location for printing. Alternatively, IP PrintWay can transmit the data sets to a remote printer in your TCP/IP network.

This chapter describes the following:

- ▶ NetSpool customization
- ▶ Defining VTAM APPL statements
- ▶ Defining NetSpool printers
- ▶ Allocating and initializing the NetSpool message log
- ▶ Defining a NetSpool startup procedure
- ▶ NetSpool exit programs
- ▶ Starting NetSpool

5.1 NetSpool

NetSpool is a component of Infoprint Server that receives print data from VTAM applications, such as CICS and IMS, converts the data into line data, and places the data on the JES spool. The data set can then be printed by JES, IP PrintWay or PSF for OS/390-controlled printers that are defined in the Printer Inventory.

Each printer definition can be associated with only one NetSpool printer LU name; therefore, if your installation needs to print to the same actual printer using different printer LU names, you need to create more than one printer definition for the same printer.

Note: This allows you to specify a printer as a destination for output submitted via NetSpool.

5.2 NetSpool customization

You can configure NetSpool so that you do not need to change existing VTAM applications; that is, existing VTAM applications can send print requests to NetSpool in the same manner as they currently send print requests to SNA network printers.

VTAM applications establish communication sessions with NetSpool printer logical units (LUs) instead of with SNA network printers, as shown in Figure 5-1. Each NetSpool printer LU must be defined to VTAM. NetSpool supports the following types of VTAM data streams:

- ▶ SNA character string (SCS) data over an LU type 1 session
- ▶ 3270 data over an LU type 3 or LU type 0 session
- ▶ A binary data stream over an LU type 0, type 1, or type 3 session

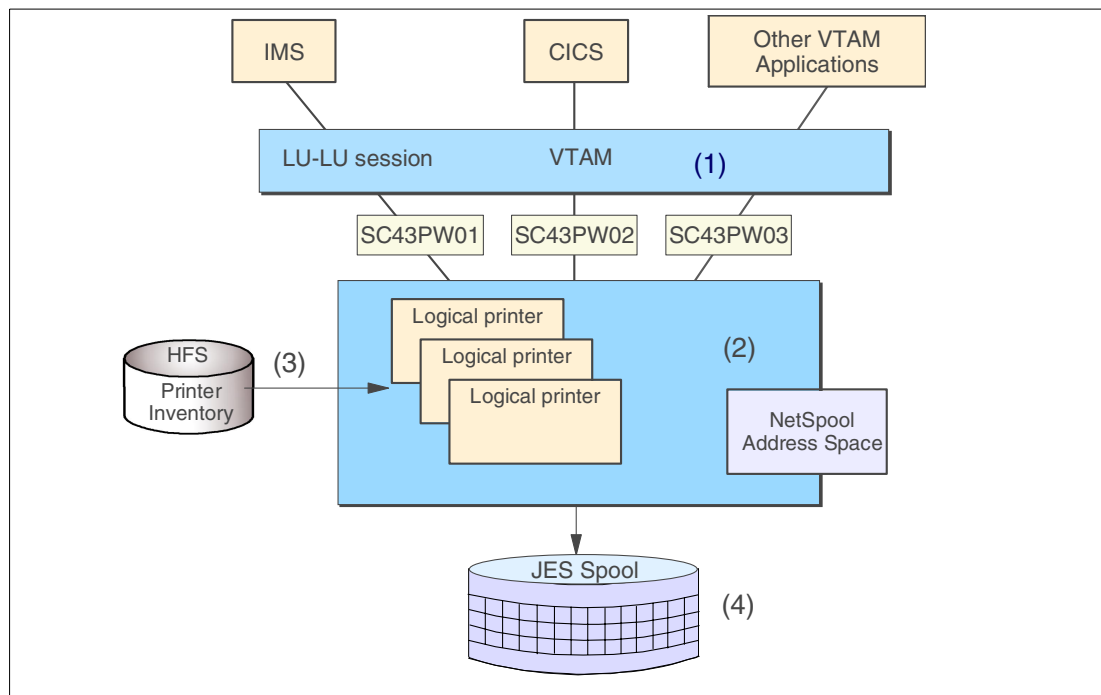


Figure 5-1 NetSpool configuration with VTAM applications

NetSpool runs as a VTAM application program, maintaining separate LU-LU sessions for each NetSpool printer LU. Each LU-LU session is between:

- ▶ A primary LU, which is the VTAM application that initiates the VTAM session and sends print requests
- ▶ A secondary LU, which is the NetSpool printer LU

Each application program must be defined within an application program major node. Each application program represents a minor node.

5.2.1 Create NetSpool VTAM APPL statements

You define an application program major node with one VBUILD definition statement, and you define application program minor nodes with one APPL definition statement for each application program in the major node.

You must create a VTAM APPL definition statement for each NetSpool printer LU. Create the APPL statement in a new or existing application major node definition.

Figure 5-2 shows sample APPL statements under a major node named NETSPOOL.

```
NETSPOOL  VBUILD TYPE=APPL
*
LUPRT0*   APPL  MODETAB=ISTINCLM,DLOGMOD=S3270,EAS=1,SESSLIM=YES
IMSPRO*   APPL  MODETAB=ISTINCLM,DLOGMOD=SCS,EAS=1,SESSLIM=YES
SC43PW0?  APPL  MODETAB=ISTINCLM,DLOGMOD=SCS,EAS=1,SESSLIM=YES
```

Figure 5-2 Sample APPL statements for NetSpool LUs

Notes on the VTAM APPL definitions:

statement-name Specifies the secondary LU name used by VTAM applications to establish a VTAM session. If this name does not match the NetSpool LU name specified in the printer definition, the ACBNAME parameter must be defined to match the LU name in the printer definition.

ACBNAME Specifies the NetSpool LU name. This name must match the LU name specified in a printer definition in the Printer Inventory. This parameter is optional. If you omit it, VTAM defaults the ACBNAME to the name specified as the *statement-name*.

MODETAB Specifies the name of a VTAM logon mode table to be used to associate each logon mode entry name with a set of session parameters. If not specified, the default name is ISTINCLM.

DLOGMOD Specifies the name of an entry in the VTAM logon mode table that contains session parameters to be used when the primary LU does not provide other parameters. If this entry is used to establish the session parameters, it must specify correct LU type, FM profile, and TS profile parameters. If not specified, the default is the first entry in the VTAM logon mode table specified in the MODETAB parameter.

The DLOGMOD entry name must be in the VTAM logon mode table named in the MODETAB parameter or in the default table named ISTINCLM.

The following shows the valid combinations of the BIND parameters:

LU3 3270 print data -- FM profile **2** TS profile **2**

LU3 3270 print data -- FM profile **3** TS profile **3**

LU1 SCS print data -- FM profile **3** or **4** TS profile **3** or **4**

EAS=1 Specifies the number of active sessions for this printer. Always specify 1.

SESSLIM	Specify YES to allow VTAM to queue pending sessions for the secondary LU, if the secondary LU already has an active session with another primary LU.
PARSESS	Specifies that multiple sessions are not allowed. PARSESS=NO is the default, so it is not shown in the example.

Defining LU names

The logical unit (LU) name that NetSpool uses to identify a printer must be a unique LU name in the Printer Inventory. This field is required for NetSpool to start a session with this printer. The LU names are defined in SYS1.VTAMLST as shown in Figure 5-3.

```

NETSPOOL  VBUILD  TYPE=APPL
*
&SYSNAME.PW0? APPL MODETAB=ISTINCLM,DLOGMOD=SCS,SESSLIM=YES,EAS=1

```

Figure 5-3 NetSpool definitions in SYS1.VTAMLST

5.2.2 Define NetSpool printers

Each NetSpool printer LU must also be defined in a printer definition in the Printer Inventory. NetSpool converts the data stream into S/390 line-data format and groups the data into output data sets using information in the printer definition.

Each printer definition can be associated with only one NetSpool printer LU name; therefore, if your installation needs to print to the same actual printer using different printer LU names, you need to create more than one printer definition for the same printer.

AOPIPDC1

IP PrintWay Printer Definition

Command ==>

Printer definition name .

NS

Description .

VTAM NetSpool SCS Printer

(extend)

Location. . .

2C-16

(extend)

Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> * _____
Processing	=> _____	=> * _____
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> _____
Protocol	=> _____	=> * _____

_ Use DEST, CLASS, and FORMS for IP PrintWay printer selection

NetSpool LU name .

SC43PW01

LU classes . .

(extend)

Figure 5-4 Adding a printer for a NetSpool request

The NetSpool Options section contains attributes that tell NetSpool whether or not to format the input data before writing it to the JES spool.

NetSpool Options	
AOIPDNO	Command ==> _____
Printer definition name	. <u>NS</u> _____
Formatting	<u>2</u> 1. None 2. Standard
Record size	_____
RECFM	<u>-</u> 1. VB 2. VBA 3. VBM

Figure 5-5 NetSpool Options panel

The NetSpool End-of-File section contains attributes that tell NetSpool when to close the output data set on the JES spool so that the data can be printed. To print VTAM application data to the same actual printer with different requirements (for example, to print to the same printer with different NetSpool end-of-file rules), you would create two printer definitions with two different NetSpool printer LU names.

NetSpool End-of-File Rules	
AOIPDNE	Option ==> _____
Printer definition name	. <u>NS</u> _____
Default rules	1 All LUs 2 LU0 3 LU1 4 LU3 More: +
PLU name _____	5 All LUs 6 LU0 7 LU1 8 LU3
PLU name _____	9 All LUs 10 LU0 11 LU1 12 LU3
PLU name _____	13 All LUs 14 LU0 15 LU1 16 LU3
PLU name _____	17 All LUs 18 LU0 19 LU1 20 LU3
PLU name _____	21 All LUs 22 LU0 23 LU1 24 LU3
PLU name _____	25 All LUs 26 LU0 27 LU1 28 LU3
PLU name _____	29 All LUs 30 LU0 31 LU1 32 LU3
PLU name _____	33 All LUs 34 LU0 35 LU1 36 LU3
PLU name _____	37 All LUs 38 LU0 39 LU1 40 LU3
PLU name _____	41 All LUs 42 LU0 43 LU1 44 LU3
PLU name _____	45 All LUs 46 LU0 47 LU1 48 LU3
PLU name _____	49 All LUs 50 LU0 51 LU1 52 LU3
PLU name _____	53 All LUs 54 LU0 55 LU1 56 LU3
PLU name _____	57 All LUs 58 LU0 59 LU1 60 LU3
PLU name _____	61 All LUs 62 LU0 63 LU1 64 LU3
PLU name _____	65 All LUs 66 LU0 67 LU1 68 LU3
PLU name _____	69 All LUs 70 LU0 71 LU1 72 LU3
PLU name _____	73 All LUs 74 LU0 75 LU1 76 LU3
PLU name _____	77 All LUs 78 LU0 79 LU1 80 LU3
PLU name _____	81 All LUs 82 LU0 83 LU1 84 LU3
PLU name _____	85 All LUs 86 LU0 87 LU1 88 LU3
PLU name _____	89 All LUs 90 LU0 91 LU1 92 LU3
PLU name _____	93 All LUs 94 LU0 95 LU1 96 LU3

Figure 5-6 NetSpool End-of-File Rules panel

The logical unit (LU) class for this NetSpool LU. If you specify more than one class, the NetSpool LU is assigned to all of the specified classes. This field is not required; the default value is class 1. To extend the length of this field, place the cursor on the word "extend" and press Enter.

Format: 1 - 64; you can specify up to 64 classes.

Note: When NetSpool starts, it starts LUs according to LU class; therefore, specify the same LU class for all NetSpool LUs that you want to start at the same time.

A0PIPDIV		VTAM Protocol
Command ==>		
Printer definition name .	NS	
Printer LU name. . .	SC43PW02	
VTAM Processing Options:		
Printer logmode. . .		
Checkpoint pages . .	5	

Figure 5-7 VTAM protocol panel for NetSpool

Multiple instances of NetSpool can run simultaneously in separate address spaces; each instance of NetSpool can process VTAM print requests sent to different NetSpool printer LUs.

For print requests NetSpool dynamically allocates output data sets on the JES spool using JES allocation attributes specified in the printer definition in the Printer Inventory, which include:

- ▶ JES work-selection parameters, such as class, forms name, and destination
These parameters cause JES to direct the output data sets to the correct JES output writer or functional subsystem application (FSA), such as PSF for OS/390 or IP PrintWay.
- ▶ Advanced Function Presentation (AFP) parameters, such as the name of a form definition and page definition
PSF for OS/390 uses these parameters when printing data on IBM AFP printers.
- ▶ Distribution information, such as name and address, which can be printed on output header pages

NetSpool can also broadcast the same data to a printer pool (groups of printers). To broadcast data, NetSpool allocates multiple output data sets on the JES spool. Printer pools are defined in the Printer Inventory.

5.2.3 Allocate and initialize the NetSpool message log

NetSpool writes informational and error messages in a NetSpool message log data set. The NetSpool message log data set is organized as a circular queue -- the oldest messages are overlaid by the newest messages. The NetSpool message log data set is optional.

Note: NetSpool can share the IP PrintWay message log.

If you do not share the IP Printway message log with NetSpool, you must allocate space for the data set and initialize it. Use the sample JCL in SYS1.SAMPLIB (APIMIJCL) to allocate space and run the ANFMFILE program to initialize the data set.

The NetSpool message log data set attributes are:

- ▶ Sequential organization (PS)
- ▶ Fixed block format (FBA)
- ▶ Logical record length (LRECL) of 57 at a minimum; 80 is recommended
- ▶ Block size (BLKSIZE) of about 10 records per block; 800 is recommended

The APIMIJCL program formats all available space.

Note: A successful execution of the APIMIJCL program ends with a system abend code D37.

All ISPF users can use the ISPF browse and view to access the NetSpool message log data. You should set up an SAF data set profile to restrict access to the data set.

5.2.4 Create a NetSpool startup procedure

Before starting NetSpool, you must create a startup JCL procedure. The startup procedure identifies:

- ▶ Program name, region size, time-out value, classes of printer LUs, tracing option, console name, and the name of the Printer Inventory
- ▶ The location of the message log data set
- ▶ The location of the trace options file
- ▶ The location of exit programs

Note: If you change the NetSpool startup procedure, you must restart NetSpool to pick up the changes

You can run multiple instances of NetSpool simultaneously, each instance running in a different address space. The following reasons explain why this might be useful:

- ▶ To spread processing across multiple address spaces
- ▶ To reduce region-size requirements by spreading the printer LUs across multiple regions
- ▶ To enable different sets of printer LUs to be started and stopped at different times

NetSpool startup procedure

NetSpool provides a startup procedure in SYS1.IBM.PROCLIB(APIJPJCL), which is shown in Figure 5-8. You can copy and modify this procedure for your installation. This procedure uses symbolic parameters with default values for some of the parameters that you might want to change for your installation.

```
//NETSPOOL PROC LUCLASS=(1),INV='AOP1',DURATION=NOLIMIT,
//          SIZE=31M,MSGFILE=USER.MSGFILE
//APIPPAAA EXEC PGM=APIPPAAA,REGION=&SIZE,TIME=&DURATION, X
//          PARM='LUCLASS=&LUCLASS,INV=&INV'
/* Modify the next STEPLIB to replace the transparent data exit or
/* the beginning of file exit. This library must be APF
/* authorized.
/*STEPLIB DD DSN=NETSPOOL.EXITS.LOADLIB,DISP=SHR
//APIMMSG DD DSN=&MSGFILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

Figure 5-8 NetSpool startup procedure in SYS1.IBM.PROCLIB(APIJPJCL)

Notes on the JCL statements and parameters that you can include in your NetSpool startup procedure:

- ▶ The EXEC statement PARM='LUCLASS=&LUCLASS,INV=&INV'

LUCLASS The classes of printer LUs that this instance of NetSpool is to start. The class of each printer LU is specified in the printer definition in the Printer Inventory.

Valid class values are 1 to 64. Enclose the classes in parentheses, even if you specify only one class value. To specify more than one class, separate each class with a comma. NetSpool starts all printer LUs that are assigned to any one of the classes specified. For example, if you specify `LUCLASS=(1,2)`, NetSpool starts all printer LUs assigned to class 1 and all printer LUs assigned to class 2. Refer to *OS/390 Infoprint Server Operation and Administration* for more information on using LU classes.

If you want to start NetSpool without starting any printer LUs, specify a class number, from 1 to 64, that does not match any classes defined for printer LUs in the Printer Inventory. To start individual printers, use the NetSpool `LUNAME ADD` command described in *OS/390 Infoprint Server Operation and Administration*.

This parameter is required.

You might want to group logical printers into classes for the following reasons:

To start classes of logical printers at different times. For example, you might want to process requests for one class of printers during the day and process requests for another class of printers during the night.

To spread processing of different classes of logical printers over different address spaces. You might want to do this if you have a large number of logical printers.

After NetSpool is started, the operator can dynamically start and stop individual logical printers that are in different classes from those specified when NetSpool was started.

Each NetSpool printer LU can belong to one or more LU classes. If you assign a printer LU to more than one class, NetSpool starts that printer LU when you start any one of the classes.

Grouping NetSpool LUs into classes is optional. If you do not specify an LU class, NetSpool assigns the printer LU to class 1. When you start NetSpool, simply start class 1 and NetSpool will start all printer LUs defined in the Printer Inventory.

INV The name that you assigned to the Printer Inventory in the inventory attribute in the `aopd.conf` configuration file. Specify `AOP1` (the default name) if you have not created an `aopd.conf` configuration file or if the inventory attribute is omitted. This name is case sensitive. This parameter is required.

CONSNAME The name of an MCS or extended MCS console where you want NetSpool to display its unsolicited messages. Unsolicited messages are issued by NetSpool as a result of an unexpected error, such as a VTAM or data error. For `console_name`, specify a name that is defined in a `CONSOLE` statement in the `CONSOLxx` member of `SYS1.PARMLIB`.

If the `CONSNAME` parameter is not specified, unsolicited NetSpool messages are displayed at the console that issued the `START NetSpool` command to start NetSpool.

LANGUAGE The language that NetSpool is to use for messages. NetSpool appends the `language_id` to the prefix `APIMT` to determine the name of the message data set to load. If you specify a language other than English, first install the NLV. If you omit the `LANGUAGE` parameter, or if NetSpool does not find the specified message data set, NetSpool issues messages in English. (NetSpool uses the default message data set `APIMTENU`.)

TRACE TRACE=ON specifies that NetSpool is to read the data set in the TRACEOPT DD statement to obtain the trace options. The tracing will start when NetSpool is started.

Note: The sample NetSpool APIJPJCL JCL procedure neither sets the TRACE symbolic variable nor defines the TRACEOPT DD statement.

NetSpool provides two types of traces:

An internal wrap trace

An external trace using the Generalized Trace Facility (GTF)

You can start either trace at initialization time or dynamically, while NetSpool is running. For more information on the NetSpool traces see *z/OS Infoprint Server Messages and Diagnosis*, G544-5747.

► STEPLIB DD DSN=dsname,DISP=SHR

This is the name of a data set that contains NetSpool exit programs written by your installation. The library must be APF-authorized. This statement is optional; specify it if you are replacing the IBM-supplied APPIPTD1 or APPIPTD2 exit programs.

► APIMMSG DD DSN=dsname,DISP=SHR

This is the name of the NetSpool message log data set. This statement is optional. If you include this statement, NetSpool writes messages to the message log data set as well as to the operator console. If you specify a language identifier other than ENU in the LANGUAGE parameter on the NetSpool startup procedure, the message log data set contains the translated messages.

► TRACEOPT DD DSN=dsname,DISP=SHR

This is the name of the trace options data set. Specify DISP=SHR to allow more than one instance of NetSpool to use the data set. The trace options data set can be either a member of a PDS or a sequential data set.

NetSpool Japanese or Spanish national language versions

NetSpool offers Japanese and Spanish versions of message data sets as separate function modification identifiers (FMIDs). To install the message data set associated with the language feature, refer to *OS/390 Program Directory*.

Note: Make certain that your MVS consoles support the national language you install. NetSpool issues all messages to both the console and an optional message log data set. Some NetSpool messages are issued before NetSpool loads the message data set. These messages will always be issued in English.

5.2.5 NetSpool Exit Programs

NetSpool provides two exits that allow you to customize the processing of SCS data streams:

► The Beginning of File exit (APIPPTD1)

This exit is called when NetSpool creates a new output data set.

► The Transparent Data Control exit (APIPPTD2)

This exit is called when NetSpool finds the Transparent Data control (TRN) in the input data stream.

These exits are called only when NetSpool is processing an SCS data stream.

5.3 Starting NetSpool

When NetSpool starts, it attempts to start any printer LUs already defined in the Printer Inventory and assigned to one of the started LU classes. If a new printer LU name is defined in the Printer Inventory while NetSpool is running, NetSpool automatically attempts to start that printer LU if it is in one of the started LU classes.

A NetSpool printer definition contains information about a printer. The printer can be a local printer or remote printer in your TCP/IP network or it can be a printer defined to VTAM. The printer definition contains all of the information that NetSpool needs to allocate data sets on the JES spool and that IP PrintWay or any other printer needs to transmit data sets from the JES spool to the specified printer.



Infoprint Server client printing

This chapter describes how Print Interface processes print requests received from both remote clients and local users. Users can submit print requests and query job status from remote clients in the TCP/IP network, using one of the following TCP/IP protocols, LPR protocol, IPP protocol, and native Windows SMB protocol. This chapter illustrates the use of these protocols and describes Infoprint Server client printing as follows:

- ▶ Sending print requests to the Print Interface
- ▶ Infoprint Server client programs
- ▶ Workstation users using the **lpr** command
- ▶ Internet Printing Protocol (IPP)
- ▶ DFS and SMB support
- ▶ z/OS users submitting batch jobs
- ▶ z/OS UNIX users using the **lp** command
- ▶ z/OS TSO/E users using the **lpr** command

6.1 Sending print to Print Interface

Figure 6-1 displays the various types of requests passed to the Print Interface component of Infoprint Server. Request types from users are as follows:

- ▶ OS/930 Printer Port Monitor from Windows workstations
- ▶ Workstation users using the **lpr** command
- ▶ Workstation users using IPP protocol
- ▶ Windows workstation users using SMB protocol
- ▶ z/OS or OS/390 users submitting batch jobs
- ▶ z/OS UNIX users using the **lp** command
- ▶ z/OS TSO users using the **lpr** command
- ▶ SAP users (See “Infoprint Server and SAP R/3” on page 229)

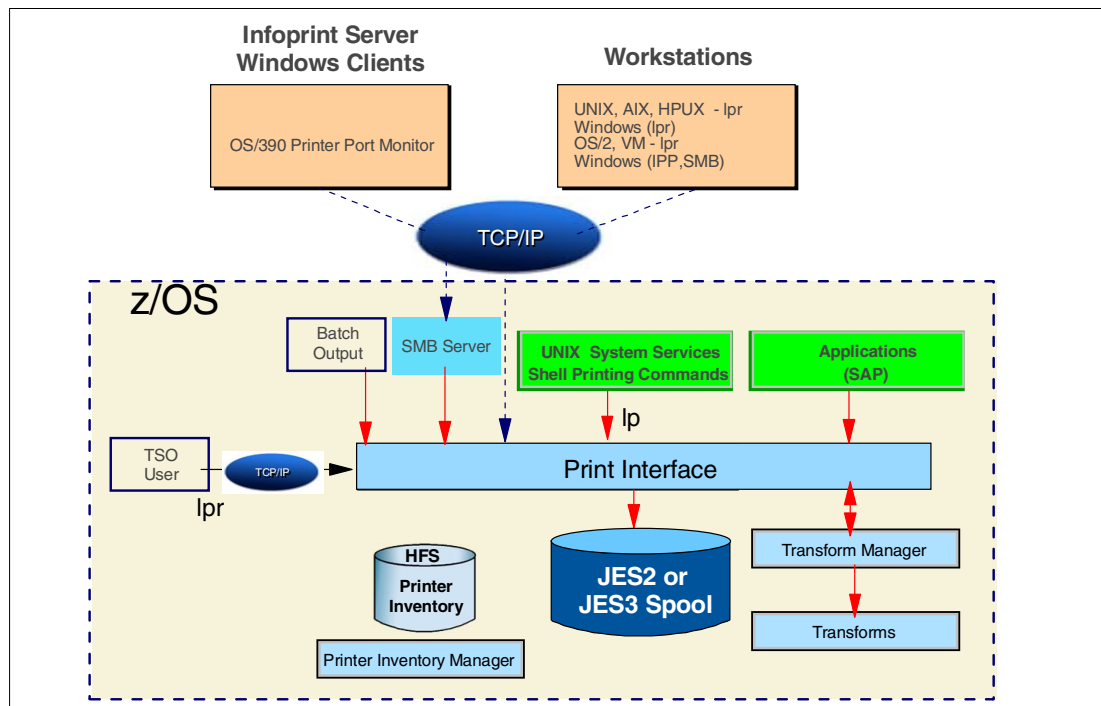


Figure 6-1 Print Interface processing requests

6.1.1 Print Interface receives data sets

In Figure 6-2 on page 135, a workstation user issues the **lpr** command and a z/OS UNIX user issues the **lp** command, to submit a PostScript file to be printed on printer *pokeps*. When the data set arrives, via TCP/IP on port 515 for the workstation user and by a forked address space for the z/OS UNIX user, Print Interface validates that the specified print data is valid and that the data stream is valid for the printer, by checking the Printer Inventory entry, see Figure 3-10 on page 71.

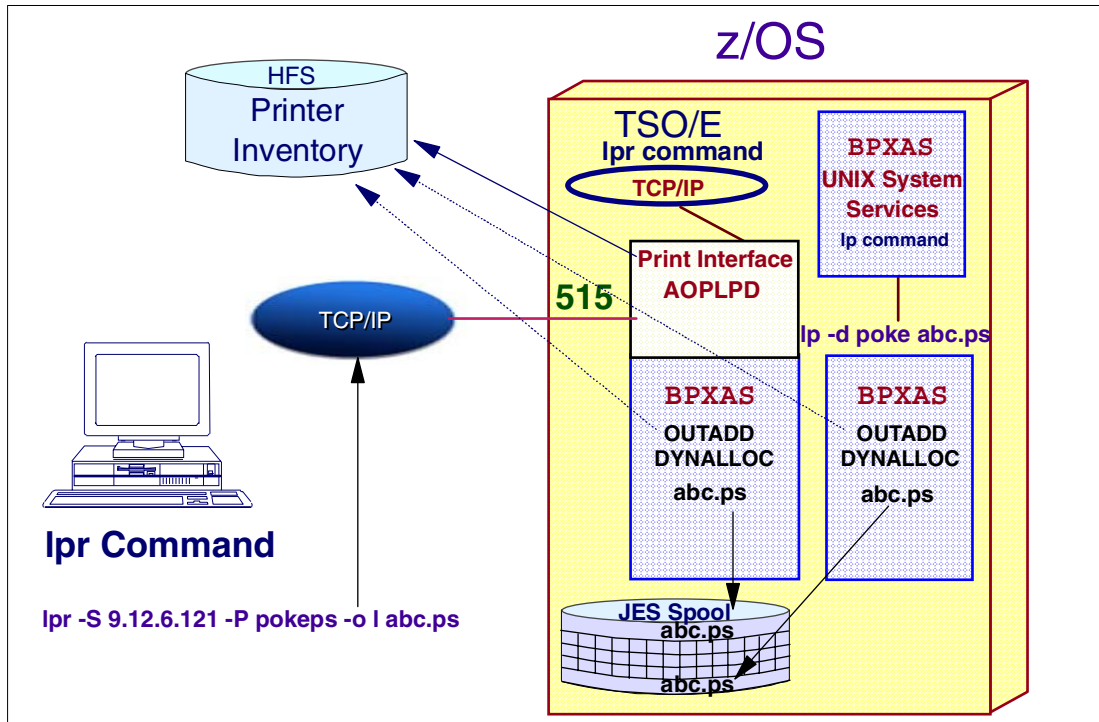


Figure 6-2 Print Interface receives data stream from user

Output characteristics

The OUTADD macro is used by Print Interface to create output descriptors from the characteristics in the Printer Inventory entry for *pokeps*. These are the characteristics that were specified in Figure 3-7 on page 69, Figure 3-8 on page 69, and Figure 3-9 on page 70.

Data sets allocated to spool

Print Interface sends the print requests to the JES spool using dynamic allocation. With this dynamic allocation of the SYSOUT, the data set characteristics from the Print Interface definition for *pokeps* such as class, forms, destination, and any of the other allocation options are passed to JES. JES then places the print request on the JES spool, where the appropriate print device selects it for output to the user specified printer.

Data set owner

From a JES perspective, the SYSOUT data set is queued to the BPXAS job that dynamically allocated the data set to the spool. The data set is assigned a job-ID using the job-prefix specified in the Infoprint Server configuration file shown in Figure 2-9 on page 33.

6.1.2 Viewing the data set on spool

When the data set is placed on the spool, it is assigned a job-ID. The job-ID job-prefix is taken from the **aopd.conf** configuration file, as shown in Figure 6-6 on page 137.

The data sets to be printed from the JES spool, that were placed there by the Print Interface, may be viewed using:

- ▶ SDSF with JES2 systems
- ▶ E(JES) with JES3 systems

Viewing data sets with JES2 SDSF

To view all data sets on the JES2 spool that have a job-prefix beginning with *PS*, set a filter on the JES2 SDSF Output display. Using the Filter Pop-up, specify the following as shown in Figure 6-3:

- ▶ Set the column to *JOBID*.
- ▶ Set the Oper to *EQ*.
- ▶ Set the Value to *PS**.

Now, when you display the Output panel, all data sets with a job-ID starting with *PS* are displayed.

Display	Filter	View	Print	Options	Help
SDSF OUT	Command ==> _____	Filter	Row 1 to 11 of 25	====> HALF	
COMMAND	Type filter criteria. Type a / in the Column or Oper fields for valid values. Press F11/23 to clear input.				ist,
PREFIX=*					Tot-Rec
ACTION=/ ACTION=P	AND/OR between columns	AND	(AND/OR)		1
NP JOB	AND/OR within a column	OR	(AND/OR)		1
ROG	Column	Oper	Value (may include * and %)		
ROG	JOBID	EQ	PS*		

Figure 6-3 Set a filter to view output with a job-ID starting with *PS*

Now that the filter is set, when you go to the Output display, you see the display in Figure 6-4.

Display	Filter	View	Print	Options	Help
SDSF OUTPUT ALL CLASSES ALL FORMS	LINES 2	LINE 1-2 (2)	SCROLL ==> HALF		
COMMAND INPUT ==> _____					
PREFIX=ROGERS DEST=LOCAL OWNER=*					
ACTION=//-Block,=-Repeat,+-Extend,?-JDS,A-Release,C-Cancel,H-Hold,L-List,					
ACTION=P-Purge,Q-Outdesc,S-Browse,X-Print					
NP	JOBNAME	JobID	Owner	PrtY C Forms	Dest
ROGERS	PS000204	ROGERS	144 J STD	POKE	Tot-Rec
ROGERS	PS000205	ROGERS	144 J STD	POKEPS	1

Figure 6-4 Display of all output with a job-ID starting with *PS*

Viewing data sets with JES3 (E)JES

To view all data sets on the JES3 spool that have a job-ID beginning with *PS*, set a filter on the E(JES) Output display. Using the Filter Pop-up, choose Option 3 Metafilters as shown in Figure 6-5 on page 137, and do the following:

- ▶ Set the Column to *JOBID*.
- ▶ Set the Oper to *=*.
- ▶ Set the Value to *PS**.

Then, when you display the Output panel, all data sets with a job-ID starting with *PS* are displayed.

```

Navigate Filter View Options Help
----- Metafilters -----
W  Navigate Filter View Options Help
C  FILTER WRITER (MIN) Row 1 of 1
C  Command ==> - Scroll ==> HALF
-  Cmd Column Op Value Bool
*  JobID = PS*
***** Bottom of Data *****

```

Figure 6-5 Set a filter to view output with a job-ID starting with PS

Now that the filter is set, when you go to the Output display, you see the display in Figure 6-6.

```

Navigate Filter View Options Help
-----
WRITER 997S 2J 0T 2 Records (0 Sched) Row 1 of 2
Command ==> Scroll ==> PAGE
Cmd JobName JobID /JP Status Process MaxComp C Pri H Dest Records
-----
ROGERS PS066736 15 J 15 N POKE 1
ROGERS PS066737 15 J 15 N POKEPS 1
***** Bottom of Data *****

```

Figure 6-6 E(JES) display of data sets on the JES3 spool

6.2 Infoprint Server client programs

The Infoprint Server Windows client, shown in Figure 6-7 on page 138, consists of the following programs that run on Windows 95, 98, NT, and 2000 systems when a document is accessed, Step 1, and printing on an AFP printer is required:

- ▶ AFP Printer Driver for Windows

The AFP Printer Driver creates the document into an output file in AFP format, so that it can be printed to an AFP printer in Step 2.
- ▶ AFP Viewer plug-in for Windows

This AFP Viewer plug-in allows the user to view the document in AFP format before printing, as shown in Step 3. Any AFP document can be viewed with this plug-in.
- ▶ Infoprint Port Monitor for Windows

The Infoprint Port Monitor for Windows allows the user to print the document using standard print submission methods from any Windows application that supports printing. A wizard printer must be defined in the user's workstation. See "Infoprint Port Monitor" on page 138.

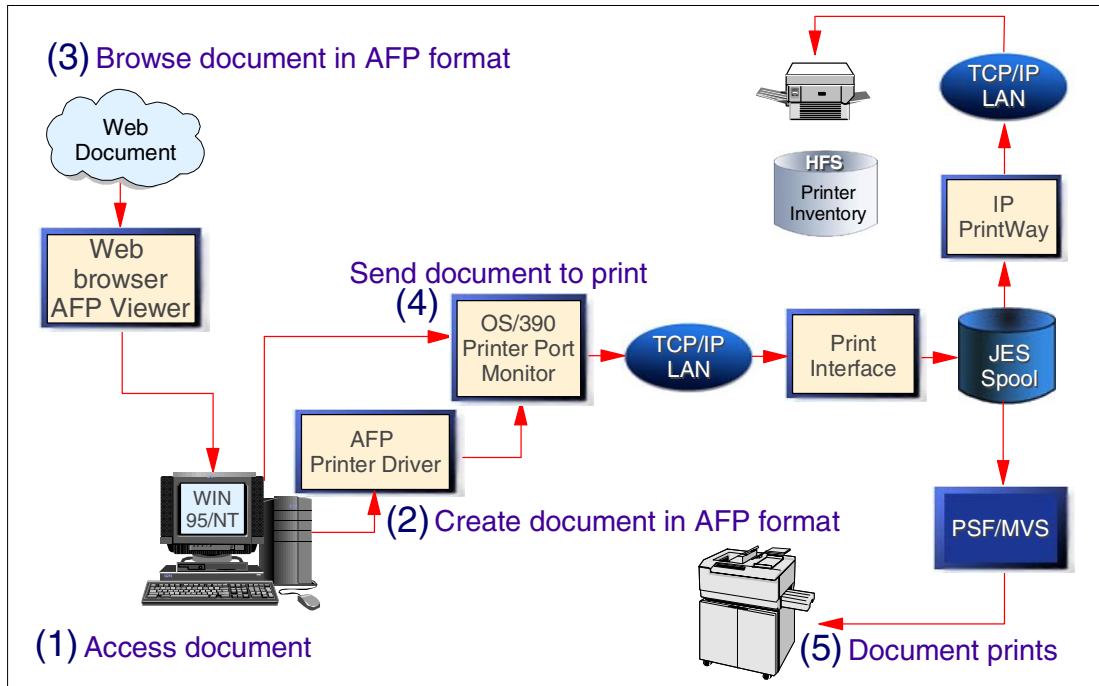


Figure 6-7 Using Infoprint Server client programs

6.2.1 Download of client programs

The client programs can be downloaded from one of the following:

- ▶ When Infoprint Server is installed, they are loaded into the HFS at `/usr/lpp/Printsrv/win/En_US`, as shown in Figure 6-8, as follows:
 - `afpdvnt.exe` (AFP driver for Windows NT)
 - `afpdv95.exe` (AFP driver for Windows 95)
 - `afpviewr.exe` (Windows AFP viewer)
 - `aopwin.exe` (Infoprint Port Monitor)

Directory List							
/usr/lpp/Printsrv/win/En_US/ Select one or more files with / or action codes.							
Type	Perm	Changed (GMT)	Owner	Size	File	Row 1 of 7	
Dir	755	08/04/2000 19:04	ROGERS	8192	.		
Dir	755	06/08/2001 18:00	ROGERS	8192	..		
File	644	06/20/2000 02:57	ROGERS	256607	afpdvnt.exe		
File	644	06/20/2000 02:57	ROGERS	177629	afpdv95.exe		
File	644	06/20/2000 02:57	ROGERS	1620476	afpviewr.exe		
File	644	08/04/2000 19:04	ROGERS	791476	aopwin.exe		
Dir	755	06/20/2000 02:57	ROGERS	8192	IBM		

Figure 6-8 Windows client programs in the HFS

- ▶ The Internet at:
<http://www.printers.ibm.com/R5PSC.NSF/Web/download>

6.2.2 Infoprint Port Monitor

You can download the **aopwin.exe** with ftp from the HFS on the host into a directory on your workstation. The Infoprint Server clients are in the following directory:

/usr/lpp/Printsrv/win/En_US

Then execute aopwin from your workstation command line, selecting any directory you choose to store the files, as follows:

```
aopwin c:\aop
```

This creates some new files and one of them is **setup.exe**. Execute this file to install the Infoprint Port Monitor in your workstation.

Create a printer with the Add Printer Wizard

When you enter the Windows NT Add Printer Wizard, the window shown in Figure 6-9 is displayed.

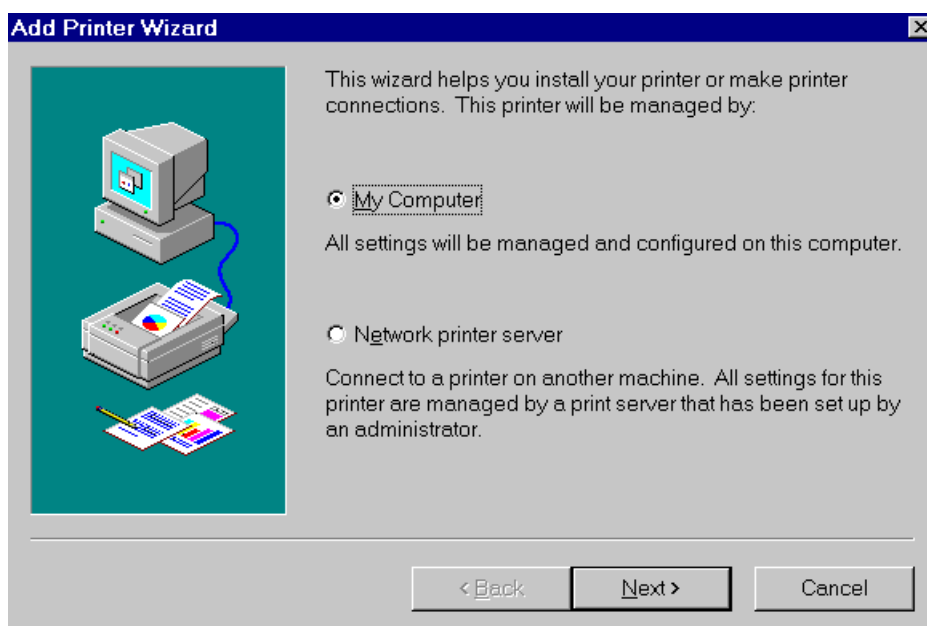


Figure 6-9 Add Printer Wizard primary panel

Click **Next** to continue. Figure 6-10 on page 140 is displayed.

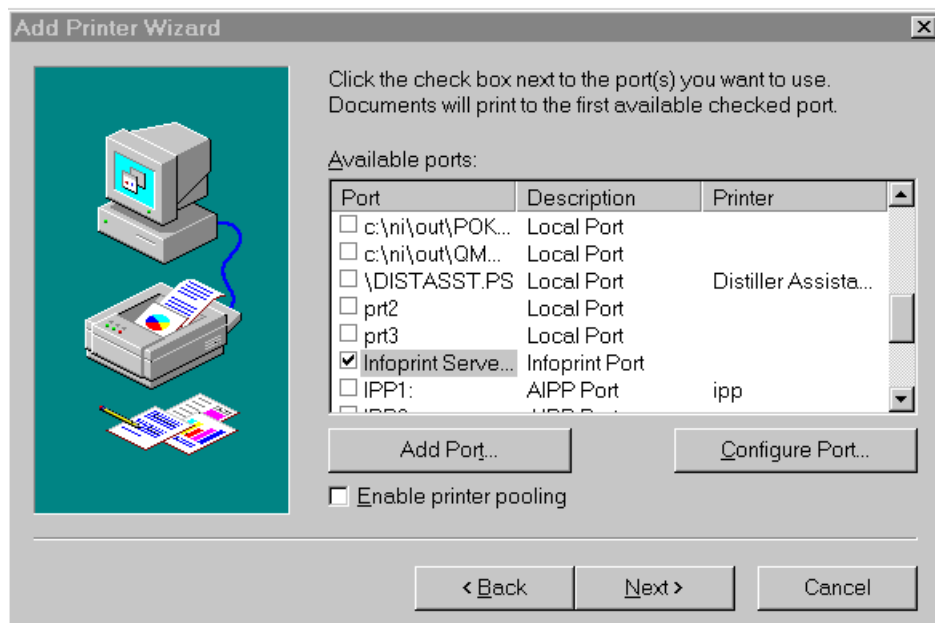


Figure 6-10 Select an Infoprint Server port

Select an Infoprint Server port, click **Configure Port** to continue, and Figure 6-11 on page 141 is displayed.

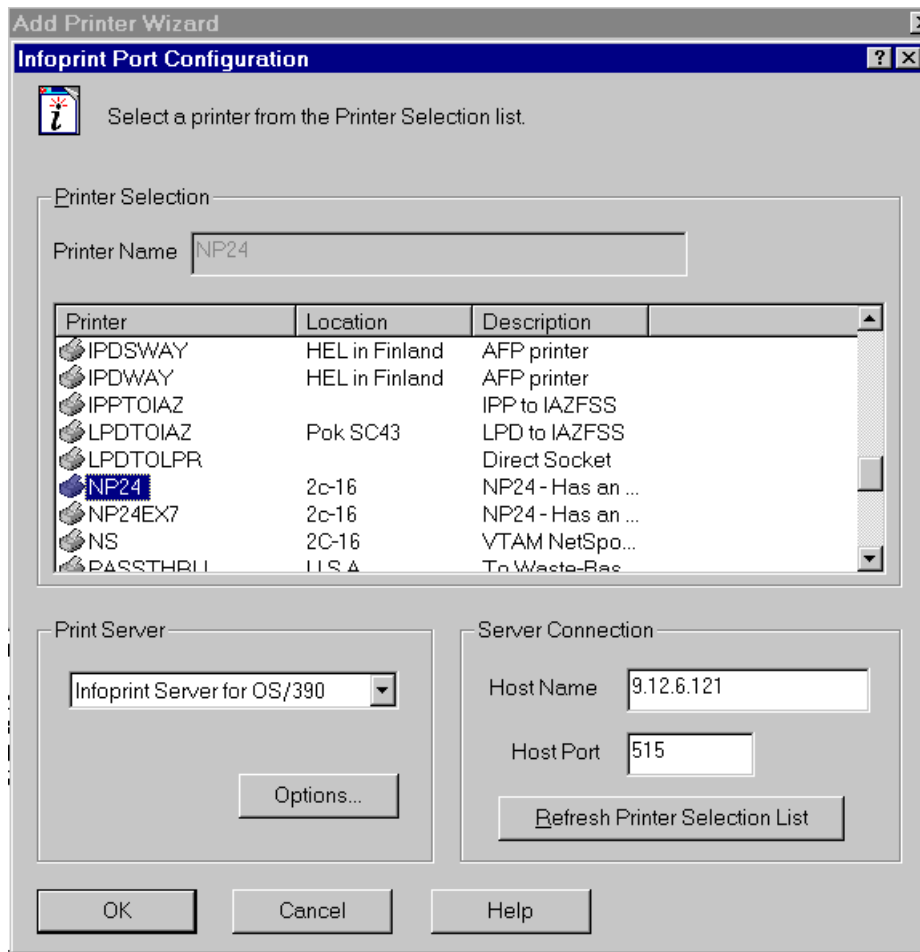


Figure 6-11 Select a printer on the host

Information required from administrator

The workstation user must receive the following local information to be able to add this printer definition and fill in the fields in Figure 6-11:

- ▶ The host name or IP address
- ▶ The host port number for Infoprint Server (This is defined in **aopd.conf** as 515.)
- ▶ Which printer defined in the Inventory that the user needs and the name of the printer driver for the selected printer

In Figure 6-11 on page 141, we entered the following for our configuration:

- ▶ 9.12.6.121, the host IP address
- ▶ 515, the Infoprint Server port
- ▶ We clicked **Refresh Printer Selection List**:
 - This causes communication with the host and the host returns the Printer Inventory entries into the window.
 - From that window, you find the printer you need to use and select it (NP24 in our case).
- ▶ Now we clicked **OK**:
 - This returned us to Figure 6-10 on page 140.
 - Now we clicked **Next** and Figure 6-12 on page 142 was displayed.

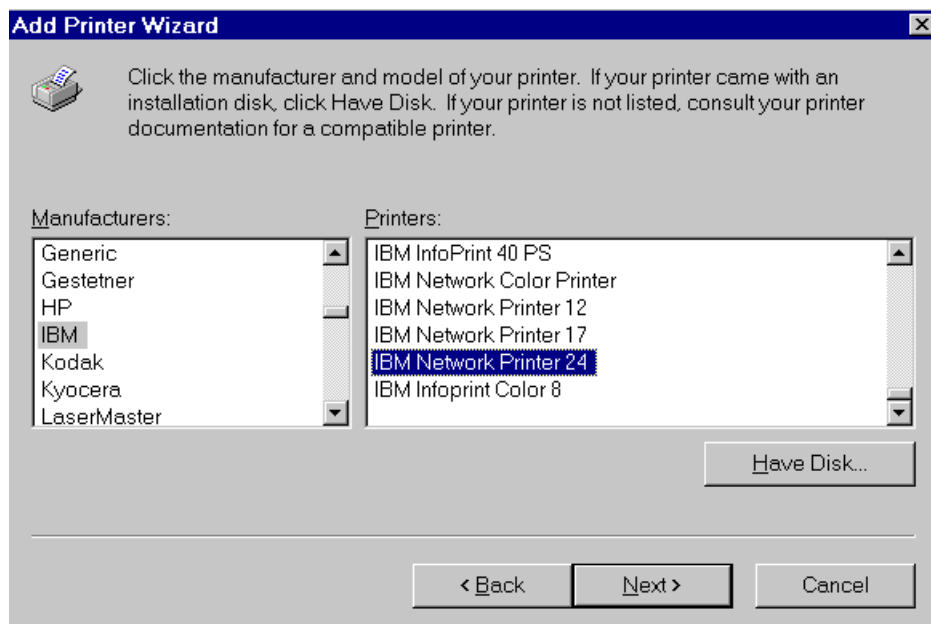


Figure 6-12 Select manufacturer and printer model

Select the manufacturer of the printer you have selected and then select the printer model from the list.

If your workstation does not have the printer driver installed, then click **Have Disk**, otherwise click **Next** and Figure 6-13 is displayed.

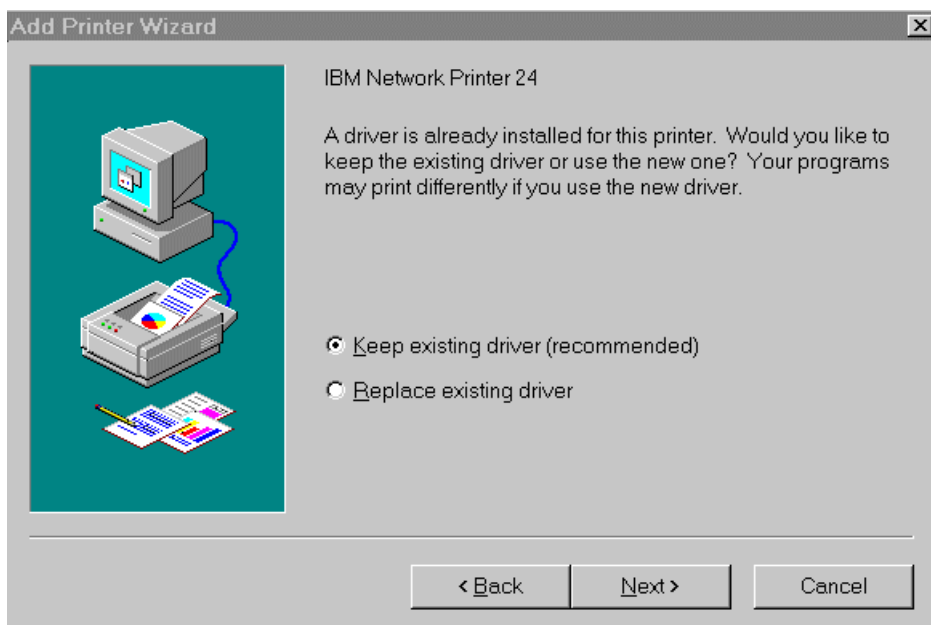


Figure 6-13 Use the existing driver that is installed

Click **Next** to continue and Figure 6-14 on page 143 is displayed.

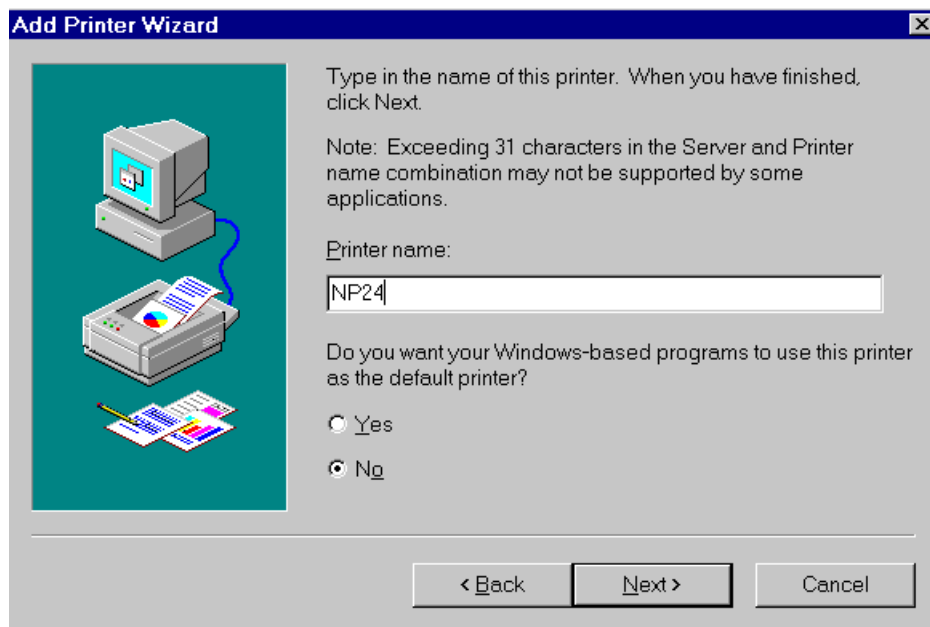


Figure 6-14 Enter the printer name you want

Enter the printer name you want to use on your workstation and choose whether or not you want this printer to be your default printer on your workstation, then click **Next** and Figure 6-15 is displayed.

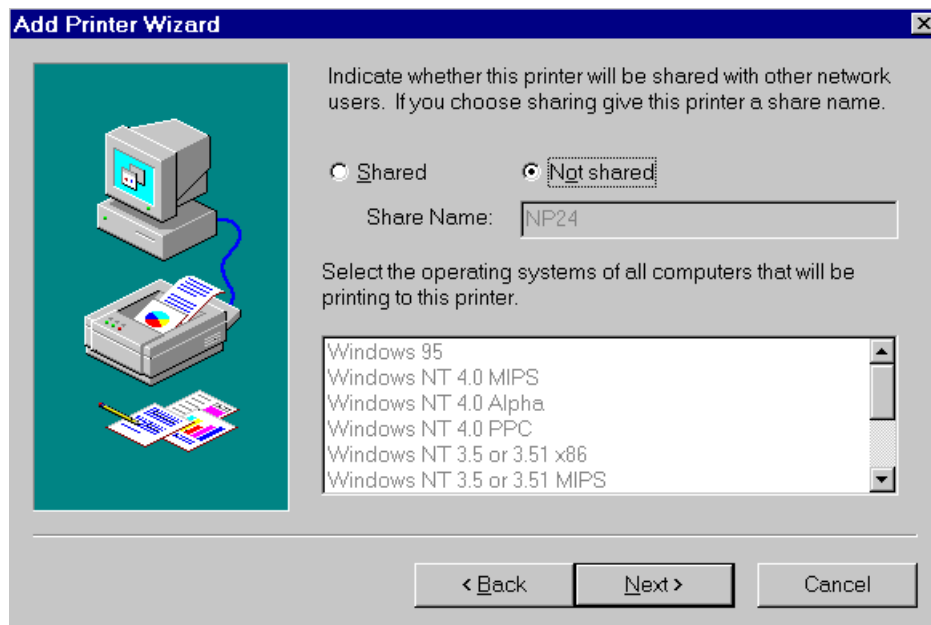


Figure 6-15 Select printer sharing option

Normally, you should choose **Not shared**, then click **Next** and Figure 6-16 on page 144 is displayed.

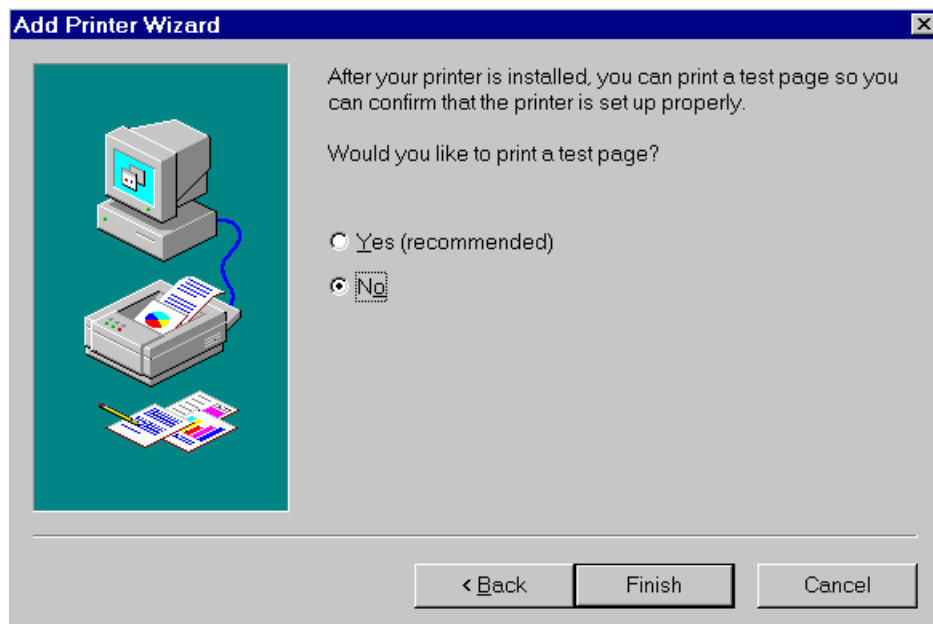


Figure 6-16 Print a test page

Select whether or not to print a test page when the printer is added and then select **Finish**. If you took the option to load a printer driver, it is loaded at this time and then the printer you have just added is now part of your printer options for printing from your workstation, as shown in Figure 6-26 on page 149.

6.2.3 AFP Printer Driver for Windows

If you require workstation documents to be converted into AFP format, you need to install the AFP Printer Driver for Windows on the user's workstation. This client program can be downloaded as shown in "Download of client programs" on page 138, from:

<http://www.printers.ibm.com/R5PSC.NSF/web/ipprintdriver>
<http://www.printers.ibm.com/R5PSC.NSF/Web/afpdr>

For example, download the appropriate Windows version into a directory, such as:

c:\temp

For Windows NT, the driver is AFPDRVNT.EXE.

Then place the files into a directory, as follows:

c:\temp\afpdrvnt c:\afpdrv

Next, you need to define an Add Wizard printer that contains the AFP printer driver.

Create the AFP printer driver

When you enter the Windows NT Add Printer Wizard, the window shown in Figure 6-17 on page 145 is displayed.

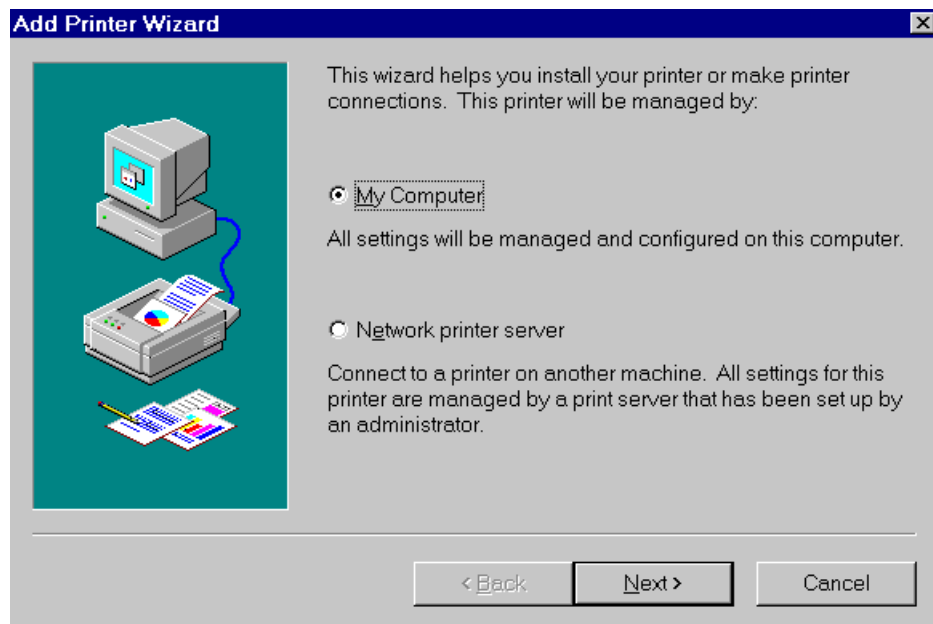


Figure 6-17 Add Printer Wizard primary panel

Select **My Computer**, then click **Next** to continue and Figure 6-18 is displayed.

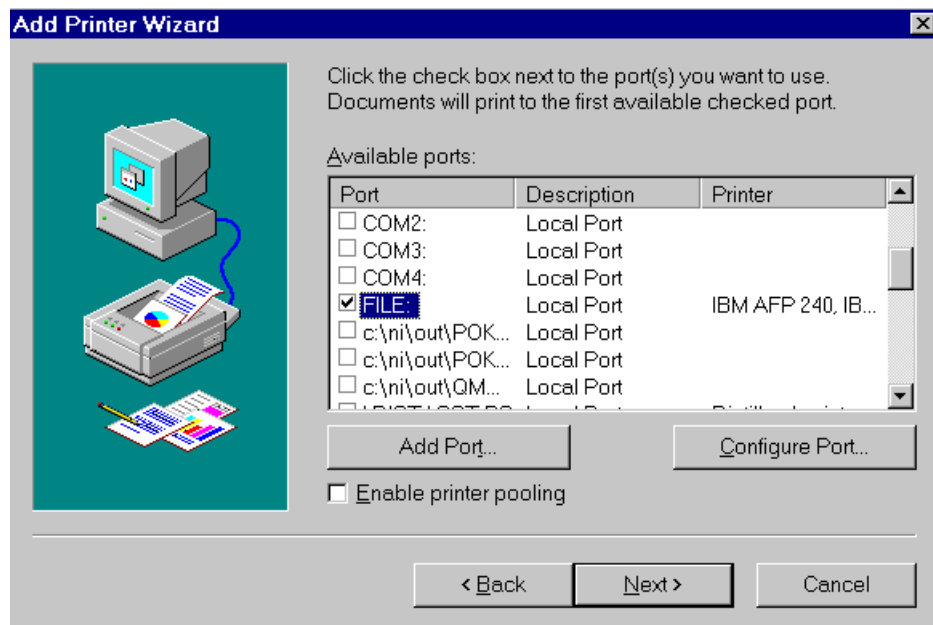


Figure 6-18 Define a port to print to a file

A port named FILE:, as shown in Figure 6-18, needs to be selected to use for the AFP printer driver. When any application uses this AFP printer driver printer we are now defining, the application data prints to a file in AFP format. Click **Next** and Figure 6-19 on page 146 is displayed.

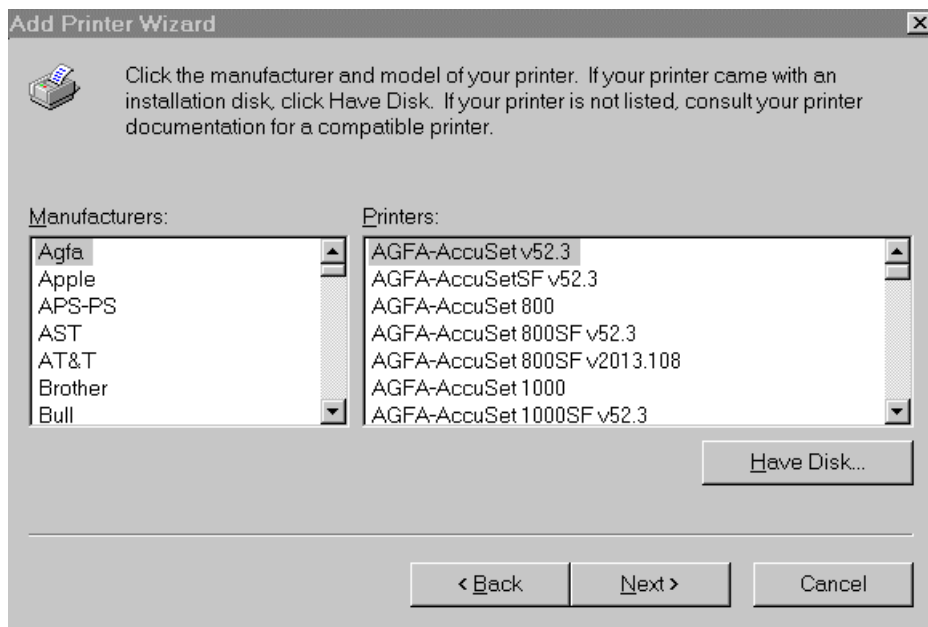


Figure 6-19 Click the Have Disk button

Click **Have Disk** and Figure 6-20 is displayed.

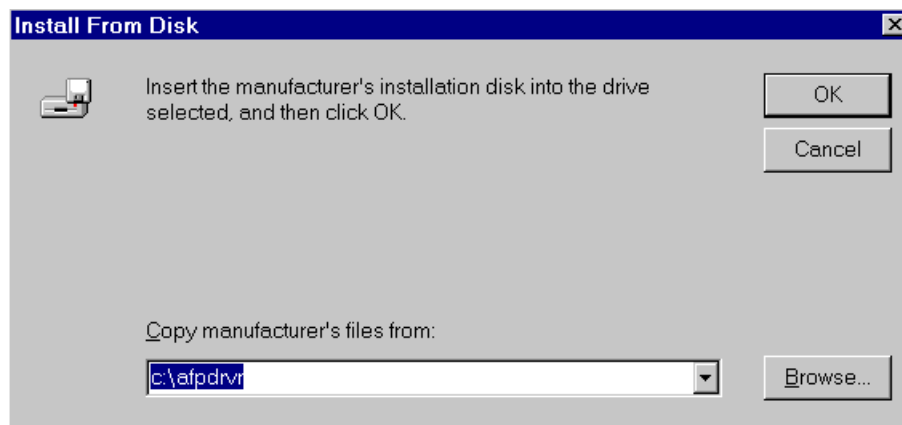


Figure 6-20 Specify the directory location of the downloaded files

In the window, type in the directory where you have placed the downloaded files, as shown in “AFP Printer Driver for Windows” on page 144, which is **c:\afpdrv**. Then click **OK** and Figure 6-21 on page 147 is displayed.

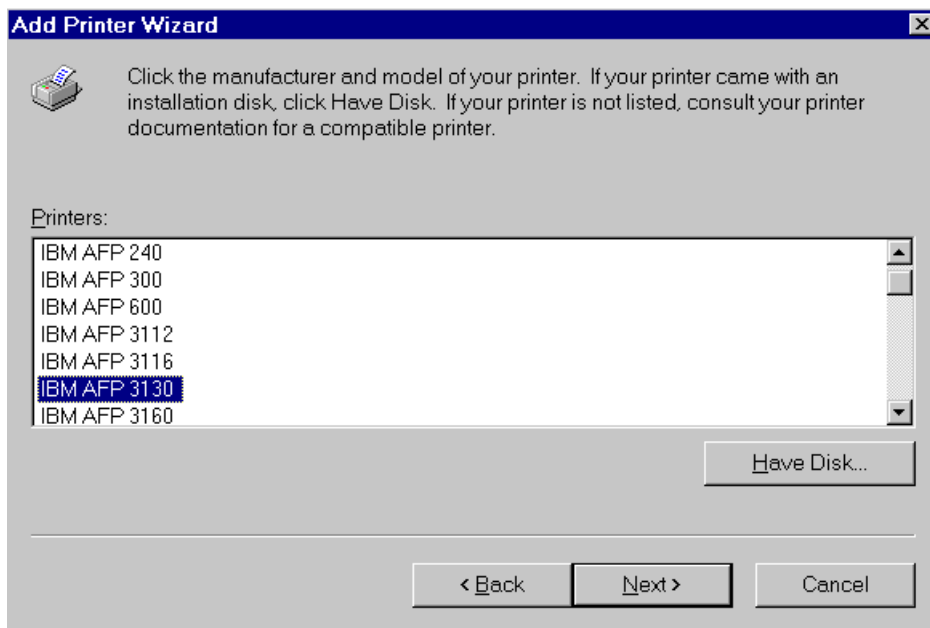


Figure 6-21 Select your AFP printer model from the list

Select your printer model from the list. If your AFP printer is not in the list, then Select IBM AFP 240 and see Step 6 of the readment.txt. Then, click **Next** and Figure 6-22 is displayed.

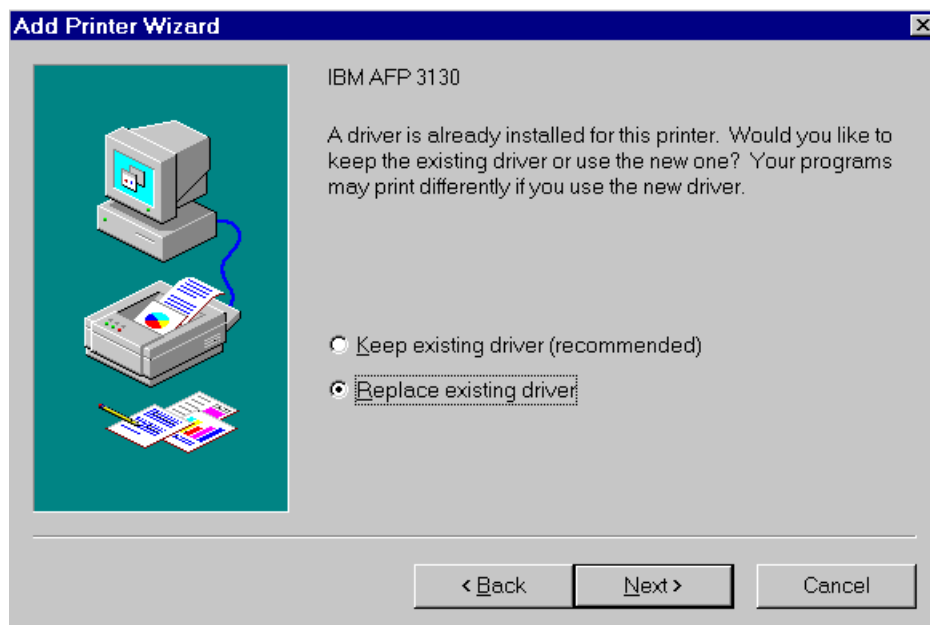


Figure 6-22 Select to replace an existing driver

Select **Replace existing driver**, then click **Next** and Figure 6-23 on page 148 is displayed.

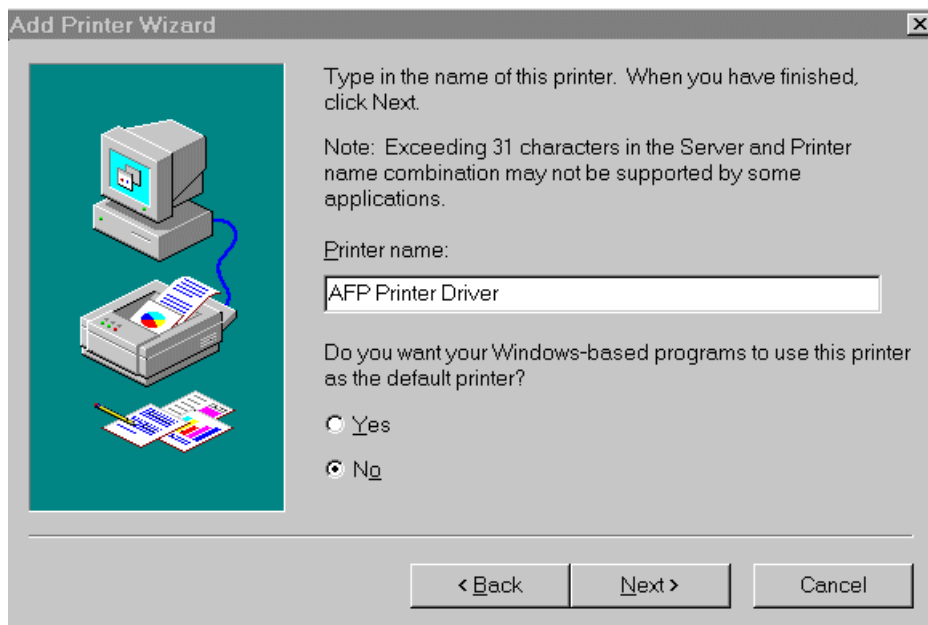


Figure 6-23 Add your printer name in the window

Choose a name for your AFP printer driver printer in the window and decide whether or not this printer should be your default printer at your workstation. Click **Next** and Figure 6-24 is displayed.

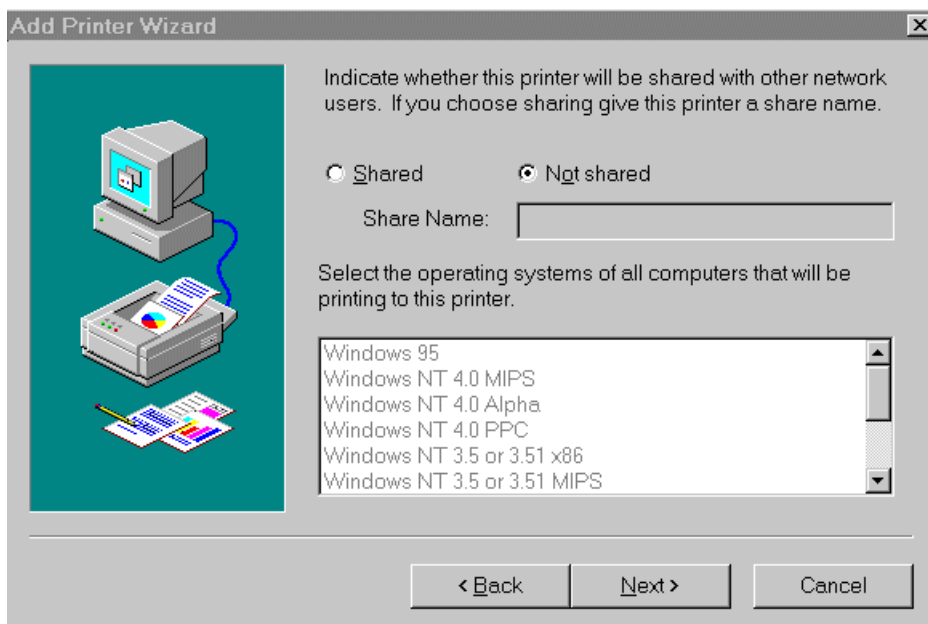


Figure 6-24 Select a print of a test page

Normally, you should select **Not shared**, then click **Next** and Figure 6-25 on page 149 is displayed.

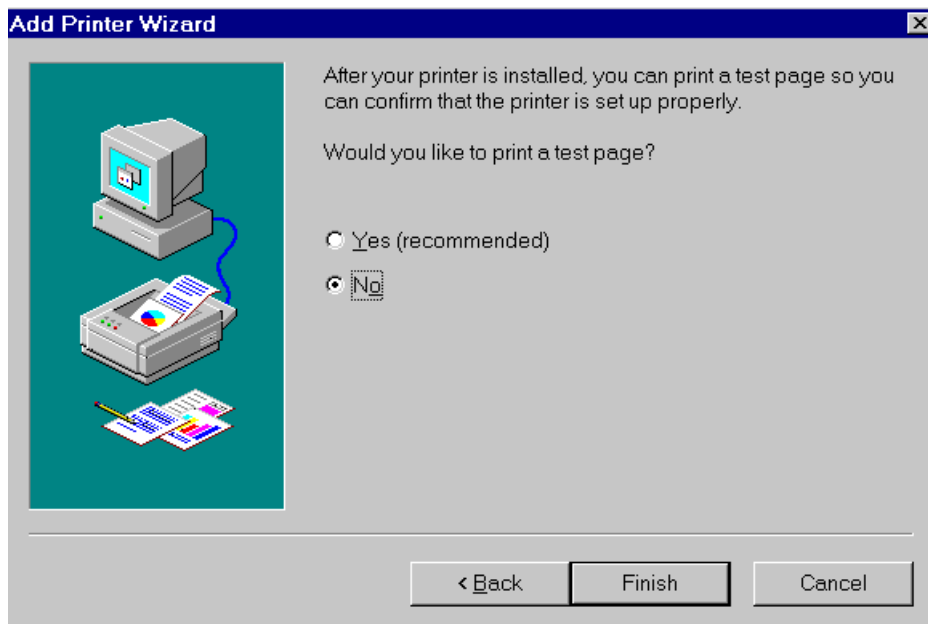


Figure 6-25 Click Finish to load the printer driver

Decide on whether you need to print a test page. Then, select **Finish** and the printer driver is loaded and the newly added Wizard printer appears in your Add Wizard printer, as shown in Figure 6-26.

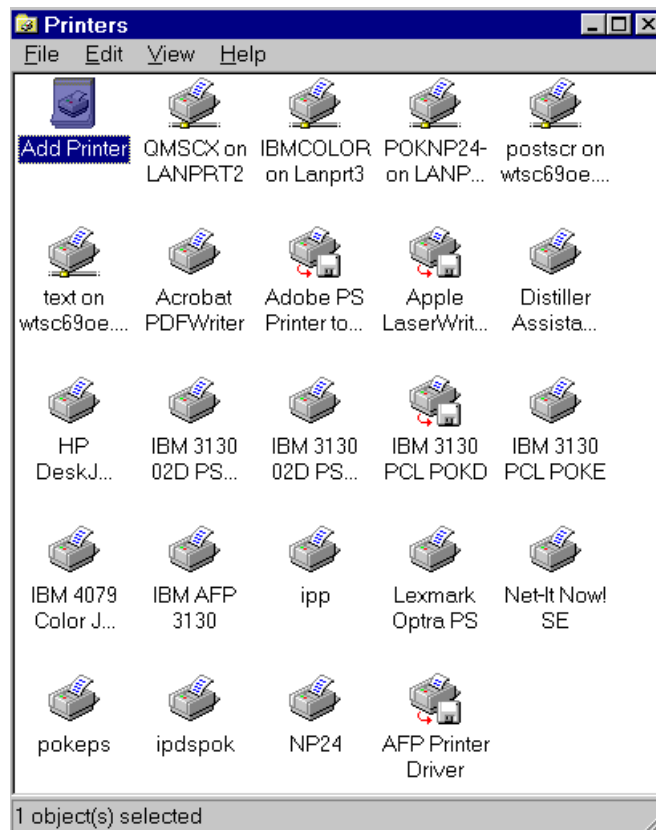


Figure 6-26 Printers added as workstation printers from Windows NT

6.3 Workstation Users using the lpr command

From workstations and command line interfaces where Transmission Control Protocol/Internet Protocol (TCP/IP) is installed, you can use standard printing commands such as the **lpr** command. The **lpr** command can be used to send print requests to the host system where the Print Interface is listening on port 515.

Figure 6-27 shows the various workstations that can use the **lpr** command to send print requests to the Infoprint Server and they include the following workstations and command lines:

- ▶ Windows 3.1, Windows Nt and Windows 2000 workstations
- ▶ VM/CMS command line
- ▶ AIX
- ▶ OS/2
- ▶ HPUX
- ▶ TSO/E ISPF command line

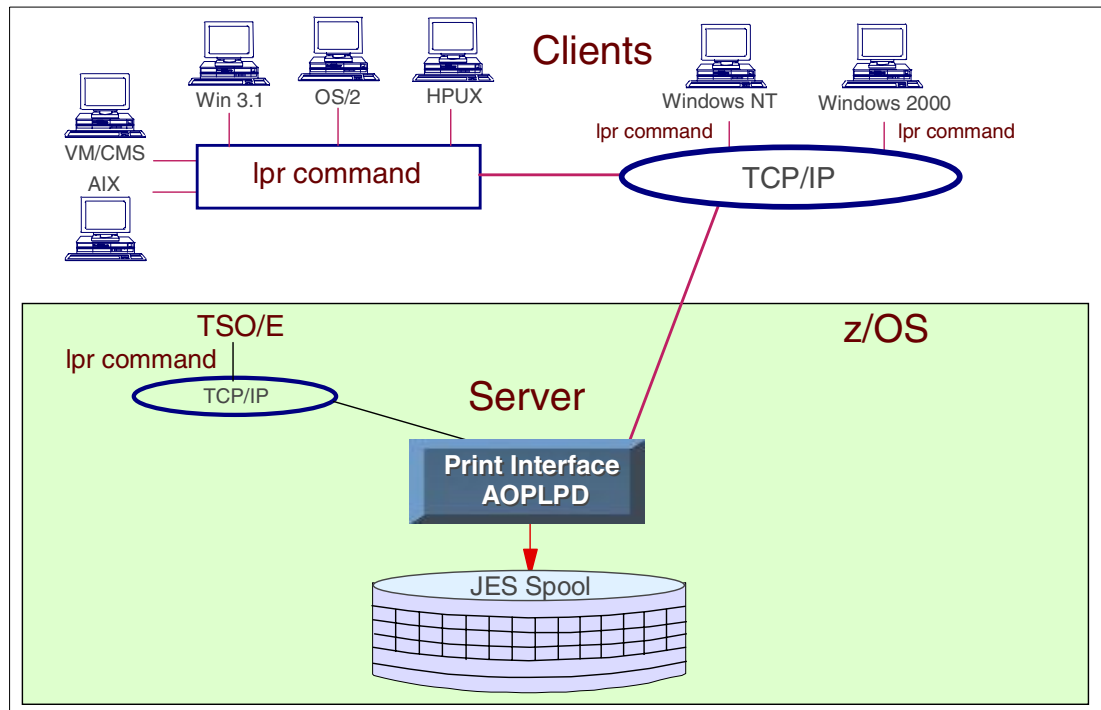


Figure 6-27 Workstation user using the **lpr** command

6.3.1 Using the lpr command

For this redbook, we tested the **lpr** command from the following workstations and command lines.

Note: Each workstation or command line interface has different **lpr** command options and methods of specifying them.

Windows workstations

You can use the Windows **lpr** command on a Windows NT or Windows 2000 system to print files directly, without using a Windows application that supports printing.

The **lpr** command options are as follows:

- S** Server - the name or IP address of the host providing lpd service
- P** Printer - the name of the printer in the Printer Inventory
- C** Class - the job class for use on the burst page and writer selection
- J** Job - the job name to print on the burst page
- o** Option - indicates the type of the file (by default assumes a text file)
Use "-o l" for binary (e.g., postscript) files.
- x** Compatibility with SunOS 4.1.x and prior
- d** Send data file first

An example of the **lpr** command to print the PostScript file **test.ps** is as follows:

```
lpr -S 9.12.6.121 -P NP24 -o l test.ps
```

VM CMS command line

From the VM CMS command line, you can print files from any VM filemode to a host system as follows:

```
lpr test jcl a (p NP24 at 9.12.6.121
```

OS/2 workstation

From an OS/2 workstation, you may enter the **lpr** command to print the data set **smpe.ps** as follows:

```
lpr -p pokeps -s 9.12.6.121 smpe.ps
```

6.4 Internet Printing Protocol (IPP)

Internet Printing Protocol (IPP) enables printing on the Internet using a client to server protocol using HTTP. An IPP client can be installed in a user's workstation and send a print request to a printer that has an IPP server installed.

Infoprint Server provides an IPP server on the host system that the IPP client can send print requests to, as shown in Figure 6-28. This IPP server runs as a JAVA application under a Web server and services print requests from IPP clients. The IPP server then forwards the print requests to z/OS Infoprint server.

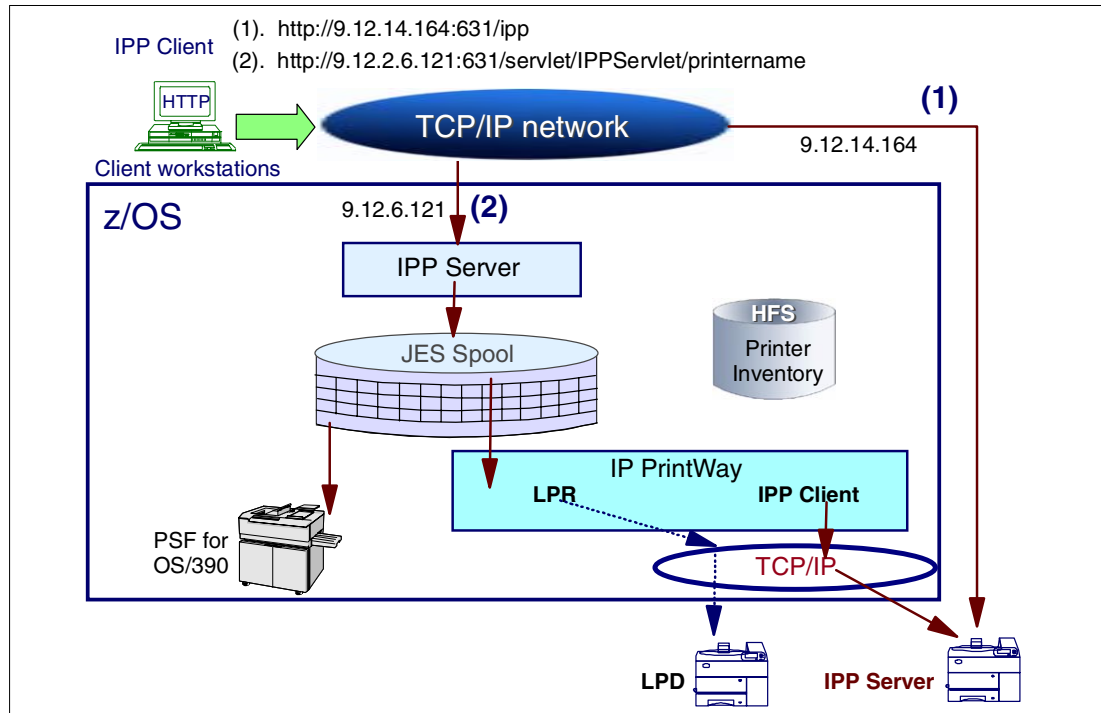


Figure 6-28 Client workstation using IPP protocol to IPP server

The IPP client in a workstation is not the same as the IP client that is part of the IP PrintWay component. The IPP client in IP PrintWay transmits output data sets from the JES spool to IPP servers running in a remote printer or host system.

6.4.1 IBM IPP client

The IBM IPP client prototype has been provided to further the development of the emerging IPP standard. Based on IBM's interpretation of the IPP specifications, the IBM prototype is being offered to software developers to test interoperability with their IPP server implementations and help validate the IPP specification. The client prototype generates simple IPP PrintJob Request data streams and receives IPP PrintJob Response data streams. The client prototype is written in Java and runs on multiple operating systems.

The IPP standard will enable vendors to develop print solutions that utilize pervasive and easy-to-use Internet tools, such as Web servers, to provide access to print services in a standard way across both public and private networks. Printing solutions which implement the new protocol will be able to take full advantage of Internet technologies and tools to extend the capabilities of the office to remote or mobile employees, business partners, and customers.

This IPP client can be downloaded from:

<http://www.printers.ibm.com/R5PSC.NSF/Web/ipp>

6.4.2 Defining a workstation IPP printer

In order to submit a print request over the Internet using the IPP protocol, the user's workstation must contain an IPP client. Using an IPP client, the job submitter can print any document (not only Web documents) on any printer defined in the Printer Inventory.

Note: Windows 2000 has an IPP client installed when you receive your operating system.

A user must first define a printer to the IPP client on the workstation. For example, on a Windows 2000 system, the user must define the printer to Windows as an Internet printer.

There are two options for IPP printing from the workstation, as shown in Figure 6-28 on page 152:

1. A user can print directly to the IPP server in a printer.
2. A user can print to the IPP Server on the host system and the print goes to the JES spool. The print data set can be selected by an IP PrintWay FSA writer and an IPP client sends the print data set to the IPP server in the printer.

The user must also specify the Universal Resource Indicator (URI) of the printer when defining an Add Wizard printer for either of the two options. The URI of any printer defined in the Printer Inventory has the following format:

`http://hostname OR ip address:port 631/servlet/IPPServlet/printername`

`http://9.12.6.121:631/servlet/IPPServlet/color` For option 1

`http://9.12.14.164:631/ipp` For option 2

See “Defining IP PrintWay printers with IPP protocol” on page 76 for additional information.

Define an IPP printer - option 1

The next set of figures describes how to add an IPP printer to a workstation list of printers. The Add Wizard screens are from a Windows 2000 system.



Figure 6-29 Add Wizard primary panel

Click **Next** to continue and Figure 6-30 is displayed.

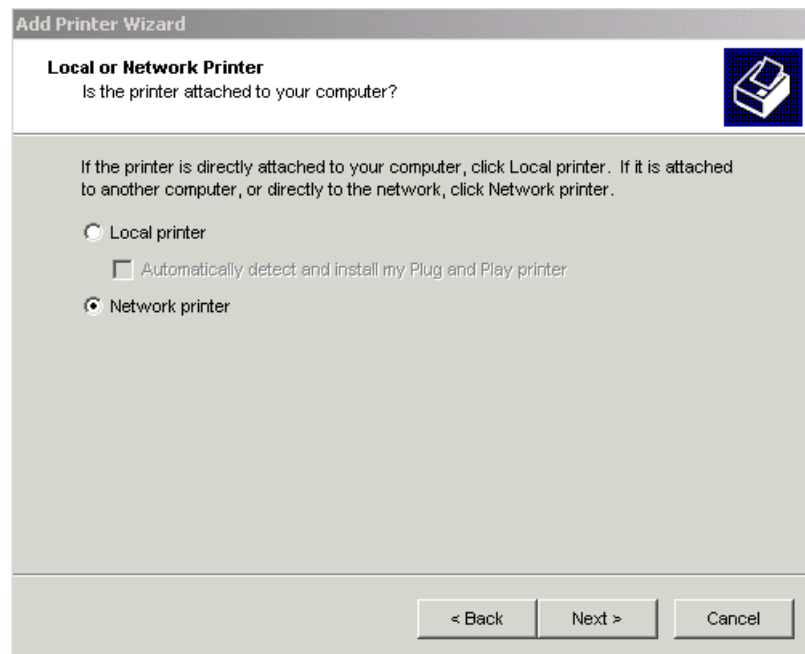


Figure 6-30 Select a Network printer for IPP printing

Select **Network printer**, click **Next** and Figure 6-31 on page 155 is displayed.

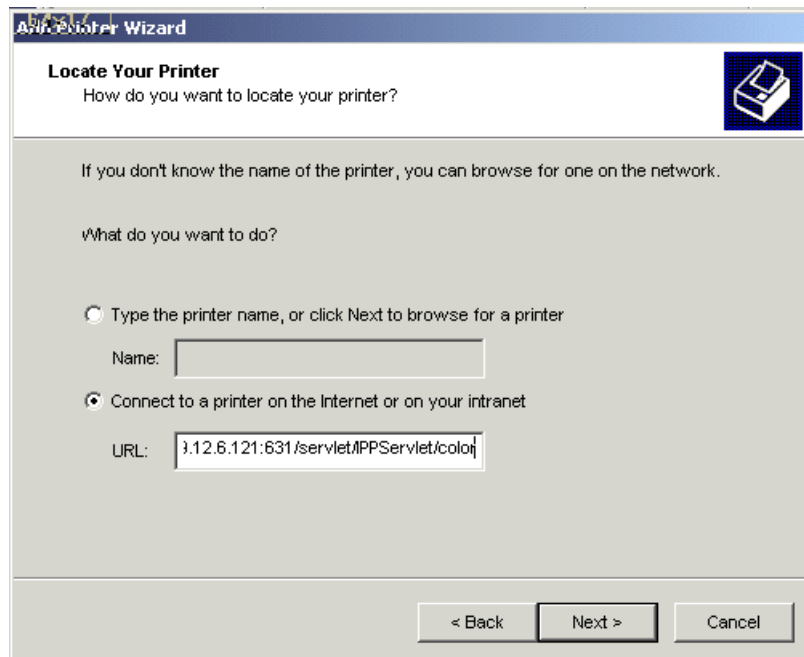


Figure 6-31 Enter a URI for a host system IPP server

Select **Connect to a printer on the Internet or on your intranet**, then enter the URL in the window as follows:

http://9.12.6.121:631/servlet/PPSrvlet/color

Then click **Next** and Figure 6-32 is displayed.

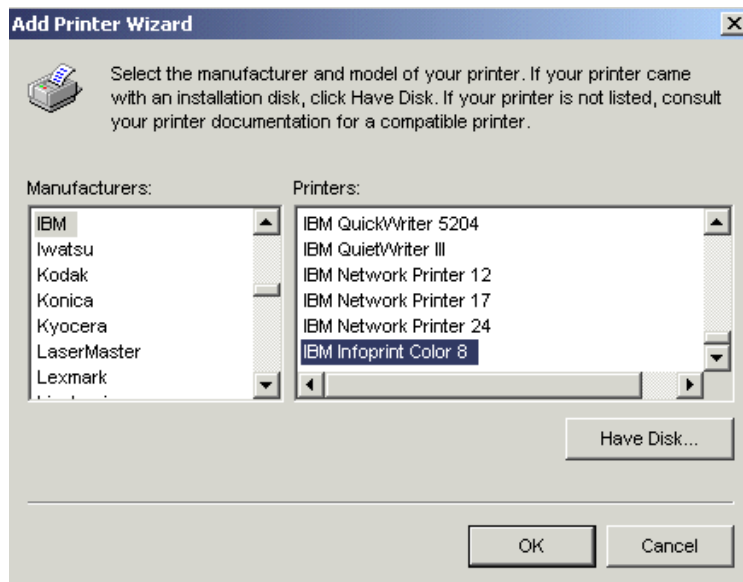


Figure 6-32 Select the printer manufacturer and type

Here you select the printer type and manufacturer of the printer you want to print your document to. Once you have selected both, if the printer driver is not currently used on your workstation, you need to click **Have Disk** and load the driver. Otherwise, if a driver already exists on your workstation, click **OK** and Figure 6-33 is displayed.

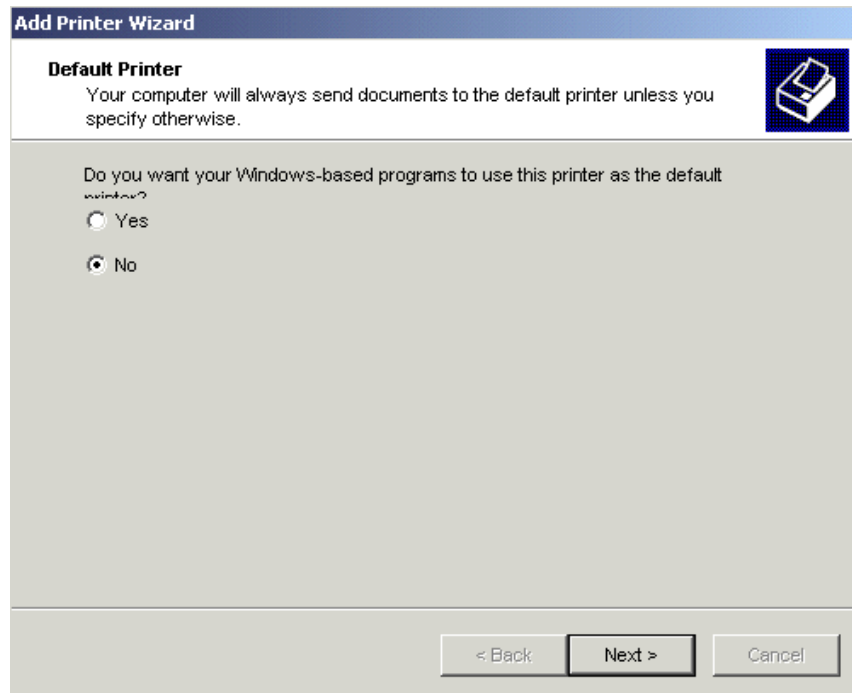


Figure 6-33 Select for use as a default printer for the workstation

Decide whether or not the printer you are defining is to be your default printer, click **Next** and Figure 6-34 is displayed.

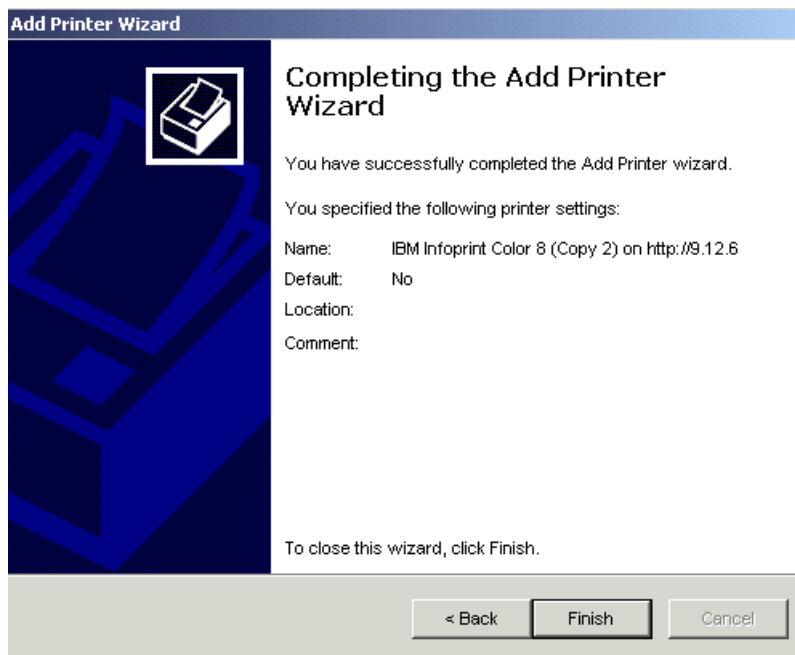
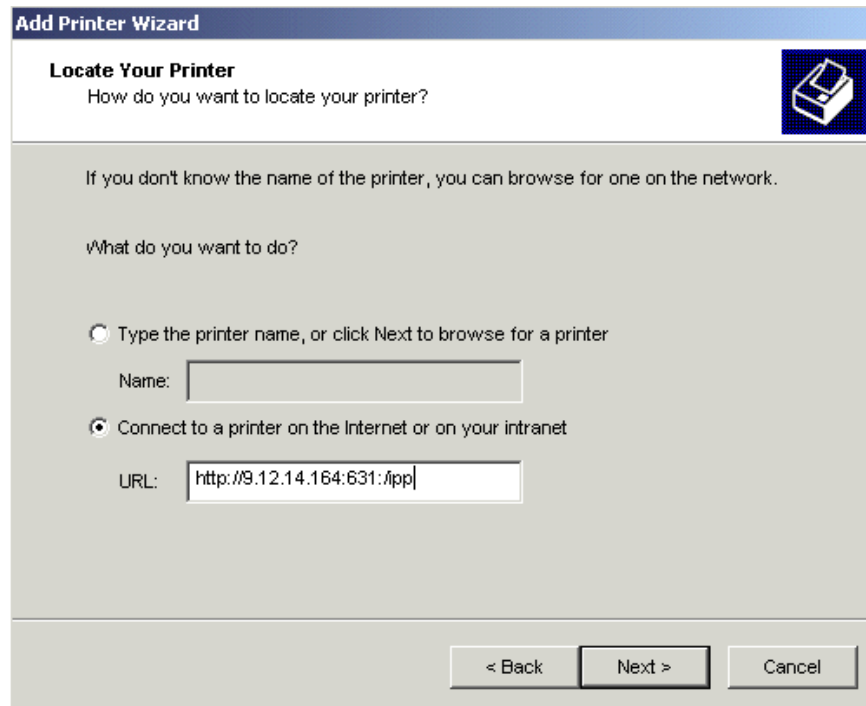


Figure 6-34 Select Finish to complete the Add Printer

Define an IPP printer - option 2

With this option, you print a document directly to the IPP server in the printer, bypassing the JES spool. In this example, we are defining an Add Wizard printer to the same printer as in option 1.

To not repeat all the panels of the Add Wizard, start with Figure 6-29 on page 154, followed by Figure 6-30 on page 154, and when you click **Next**, Figure 6-35 is displayed.



Add Printer Wizard

Locate Your Printer
How do you want to locate your printer?

If you don't know the name of the printer, you can browse for one on the network.

What do you want to do?

☐ Type the printer name, or click Next to browse for a printer

Name:

☒ Connect to a printer on the Internet or on your intranet

URL:

< Back Next > Cancel

Figure 6-35 Define the URL for printing directly to the IPP server

To print directly to the IPP server in the printer, use the following URL:

http://9.12.14.164:631/ipp

This is the only difference with the option 1 definitions, so continue as with option 1 with Figure 6-32 on page 155 through Figure 6-34 on page 157.

Printer Inventory IPP printer

To support the option 1 printing where the data set is passed to the spool, an IP PrintWay printer definition must be defined for the printer named *color*, as shown in the Add Wizard definition. Complete all the steps for defining the IP PrintWay printer and the protocol option, as shown in Figure 3-15 on page 76 and repeated in Figure 6-36.

Edit	IPP Protocol
Command ==>	<input type="text"/>
Printer definition name	. <u>color</u>
URL	<u>http://9.12.14.164:631/ipp</u> (extend)

Figure 6-36 IPP protocol panel for IP PrintWay printer color

An IPP client is used by IP PrintWay when the data set is selected to send it to the IPP server at IP address 9.12.14.164.

Figure 6-37 shows the JES3 messages from the SYSLOG.

Note: Message AOPD001 is a usermod on our system that shows the data set being received from the Windows 2000 workstation.

The FSA writer of IP PrintWay is named PRTWAY.

```
AOPD00I AOPIPPD (JOB15337) Prt: color          - WTSCPLX9 ROGERS02 PS067290
ANFM700I Data set: WTSCPLX9.ROGERS02.ROGERS02.JOB15337.D0000009.PS067290 The data
          set has been acquired by PRTWAY
*IAT7005 WTR (JOB15355), ON PRTWAY (    ), WAITING FOR WORK.
IAT7005 WTR (JOB15355), ON PRTWAY (    ), WS=(CL,F), WC=J.
IAT7005 WTR (JOB15355), ON PRTWAY (    ), F=STD.
```

Figure 6-37 Messages from a JES3 SYSLOG

Figure 6-38 displays the IP PrintWay message log messages for the processing of the data set by IP PrintWay.

```
2001062814561142-ANFM700I Data set:
          WTSCPLX9.ROGERS02.ROGERS02.JOB15337.D0000009.PS067290
          The data set has been acquired by PRTWAY
2001062814562708-ANFM1704W The destination http://9.12.14.164:631/ipp does not
          support the attributes { copies sides }.
2001062814562709-ANFM601I Data set:
          WTSCPLX9.ROGERS02.ROGERS02.JOB15337.D0000009.PS067290
          The data set was successfully transmitted to host and queue:
          http://9.12.14.164:631/ipp
2001062814562711-ANFM604I Data set:
          WTSCPLX9.ROGERS02.ROGERS02.JOB15337.D0000009.PS067290
          The data set is being released to JES
```

Figure 6-38 IP PrintWay

6.5 DFS and SMB support

OS/390 Distributed File Service (DFS) supports two distributed file protocols, DCE DFS and SMB. SMB stands for Server Message Block and is the file and print protocol of the Windows environment.

SMB support is based on X/Open CAE Specification Protocols for X/Open PC Interworking: SMB, Version 2. The SMB File/Print Server processes requests for printing, as shown in Figure 6-40 on page 161, from:

- ▶ Windows 95, 98, 2000
- ▶ Windows NT 4.0 workstations

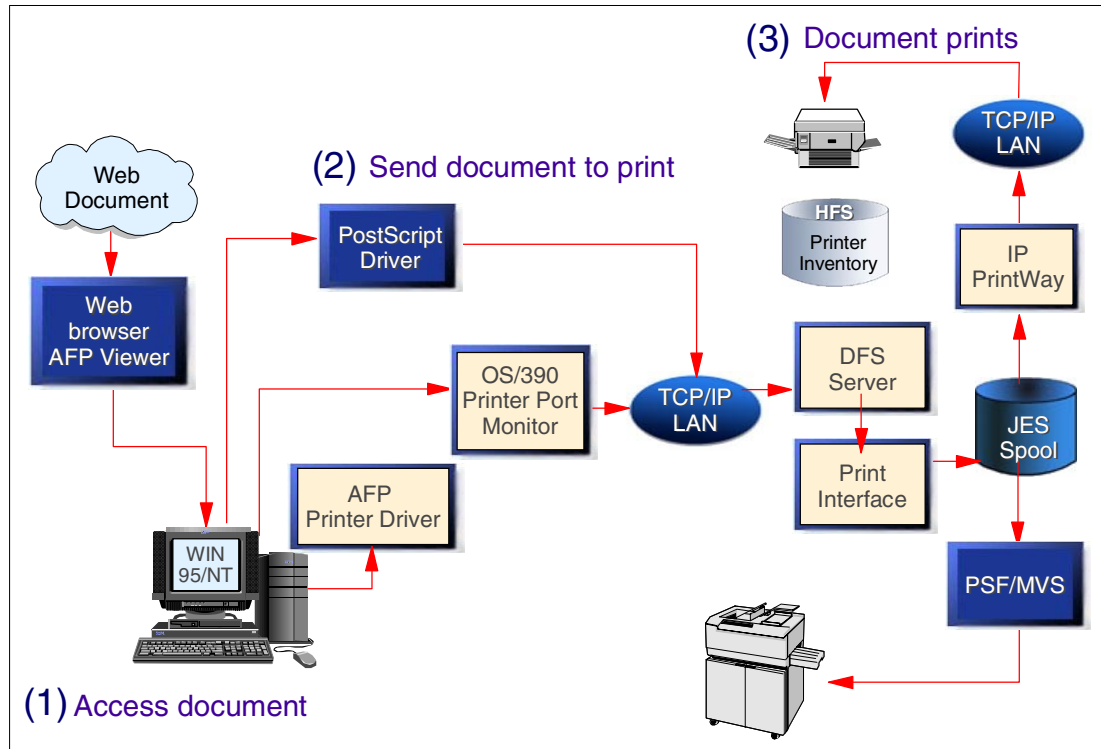


Figure 6-39 A user prints a document on the host system using SMB protocol

The SMB protocol is supported through the use of TCP/IP on z/OS. This communication protocol allows clients to access shared directory paths and shared printers. Personal Computer (PC) clients on the network use the file and print sharing functions that are included in their operating systems.

In addition, Windows SMB clients can make remote print requests to host printers that are connected to the Infoprint Server for z/OS or OS/390. In addition, OS/390 SMB support allows Windows workstation users to be able to use remote printers that are attached to the host system. Remote printers simply appear to be additional printers that are available to the workstation user.

Remote printers are installed on workstations using existing commands or install utilities.

Shared printers are assigned to JES print queues and printer names defined to the z/OS Infoprint Server. The JES queues are written to or queried by the SMB File/Print Server in response to SMB point requests. In turn, the Infoprint Server reads and processes print requests from the JES queue.

6.5.1 z/OS DFS Server

The z/OS SMB server acts as a print server that makes the services of the z/OS Infoprint Server available to workstation clients. This allows clients with the proper print drivers to submit print jobs to the z/OS Infoprint Server through the z/OS DFS server. The available print types include the following:

- ▶ Postscript
- ▶ PCL
- ▶ Text
- ▶ Advanced Function Printing (AFP)

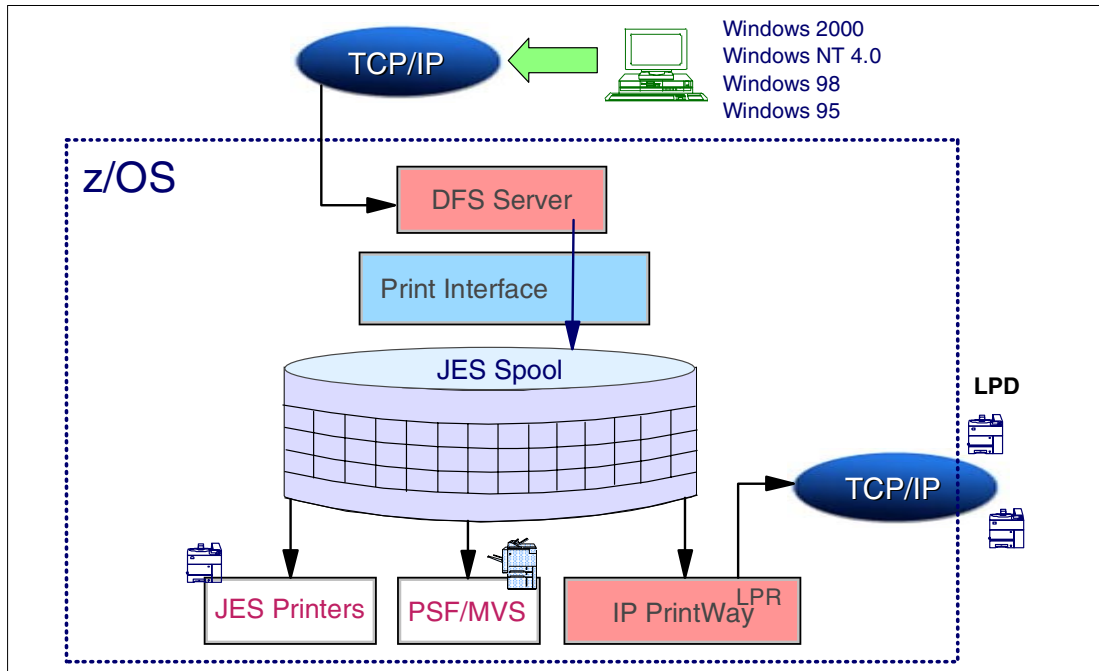


Figure 6-40 Windows workstation users use SMB protocol to process print data sets

You can access print drivers for supported Windows workstation clients in either of these two ways:

1. If the printer type specified on the smbtab (for example, Generic/Text Only) is available on the Windows system, Windows automatically uses that printer type (and print driver) when the printer is added by the Add Print Wizard.
2. Print drivers are available for free downloading from the IBM Printing Systems Company World Wide Web (WWW) site. It is located at:

<http://www.ibm.com/printers>

6.5.2 Accessing printers

Before a workstation client can print to an OS/390 Infoprint Server printer through the OS/390 SMB server, a shared printer must be created. A shared printer represents an OS/390 printer controlled by the OS/390 Infoprint Server.

Once the workstation client is connected to the shared printer, workstation users can print to that OS/390 printer as though it were a local printer.

An administrator needs to make printers available to workstation users and inform them as users how to access the printers. The steps necessary to provide SMB processing are as follows:

- First, we need to make sure that the OS/390 Distributed File Service server is started with SMB processing enabled. This is controlled by an environment variable.
- Next, we need to decide what data we want to make available to workstation clients.
- Then we need to update the export files and make the data available.

The smbtab export file in the HFS contains the smb printer definitions, as follows:

This tells the SMB server which directory is to be shared, the file system that the directory is in, and gives a name to the directory share for workstation users to use. The parameters used are as follows:

device name	A unique device name that refers to the file system device name in the dfstab
share name	The directory share name that the workstation user will use to “connect” to the HFS data
device type	ufs refers to UNIX File System (that is, HFS); prt refers to a printer device type
description	A text description that shows up on a net view for the OS/390 SMB server
permission	r/o means the share can only be accessed in a read-only fashion; r/w means that workstation users can read and write to the data (assuming they are authorized)
max users	The maximum number of users that can “connect” to the directory share; 0 means unlimited
directory	The directory name (within the file system referred to in the device name) to be shared with the workstation users

An example of an smbtab file in the HFS is shown in Figure 6-41. This file can be accessed in the HFS using the following command:

/etc/dfs/var/dfs/smbtab

#	device	share	type	share label	permissions	max users	path
#							
	/dev/ufs2	umikem	ufs	"Mike's home"	r/w	100	/
	/dev/ufs3	nichola	ufs	"Als share "	r/w	100	/
	#/dev/ufs4	root	ufs	"wtsc43oe root"	r/w	100	/
	/dev/prt2	postscr	prt	"A PS Printer"	pokeps	"3130IBM"	
	/dev/prt1	text	prt	"A text Printer"	poke	"TextOnly"	
	/dev/prt3	text1	prt	"A text Printer"	poke	"TextOnly"	

Figure 6-41 The smbtab file in the HFS

6.5.3 User requests for files and printing

When workstation users send requests (file or print) to the OS/390 SMB server, the requestor is identified by their SMB user ID. Since HFS data is authorized by a requestor's OS/390 user ID, the incoming SMB user ID must be mapped to its corresponding OS/390 user ID. This is accomplished by the OS/390 SMB server based on a configuration file called the smbimap file. The SMB server administrator creates this file and puts an entry for each SMB user ID and its corresponding OS/390 user ID.

The administrator can also specify that if the SMB user ID is not found in the smbimap file (or the password was invalid), the SMB user ID can be considered to be the OS/390 user ID. (The SMB user ID must be eight characters or less, in this case.) Finally, if the logon still fails, the administrator can specify that a default (or guest) user ID can be used. The default OS/390 user ID is specified in an environment variable and must, of course, be a valid OS/390 user ID. If there is no default OS/390 user ID, the SMB request is denied.

Authorization to files and directories is handled in the normal manner with the **chmod** command to change permissions or the **chown** command to change the owner.

The smbimap file is located via the _IOE_SMB_IDMAP environment variable.

The smbimap file is a text file that the administrator creates and maintains. It must be an HFS file. Any editor available on z/OS UNIX may be used (for example, oedit, vi, etc.). The smbimap file contains one or more mapping declarations and has the general format shown in Figure 6-42.

```
SMB-user-ID1
OS/390-user-ID1

SMB-user-ID2
OS/390-user-ID2

*
=
```

Figure 6-42 Mapping of smbimap file

Figure 6-43 shows the smbimap file. This file can be accessed in the HFS using the following command:

/etc/dfs/home/dfskern/smbimap

```
ROGERS01
ROGERS

RCONWAY1
RCONWAY

USER100
USER1

USER200
USER2

*
=
```

Figure 6-43 The smbimap file

Each entry has two elements: the SMB user ID and the OS/390 user ID. A blank line is required between entries. The first line of each entry is the SMB user ID to be mapped. It can be either a simple SMB user ID or it can be qualified by a domain name. (There may be two SMB users with the same ID in different domains.) If it is not qualified by Domain, that means you don't care which domain it came from (that is, it could have come from any domain or no domain).

The second line is the corresponding OS/390 user ID.

Several SMB user IDs can be mapped to the same OS/390 user ID.

A special entry allows you to specify that if the SMB user ID is not found in the smbimap file, or the login was unsuccessful, then the SMB user ID should be considered to be an OS/390 user ID.

This entry has * for the first line and = for the second line. This will only be done if the SMB user ID is eight characters or less.

Again, if all of the above fails to accomplish a successful login, then the _IOE_MVS_DFSDFLT environment variable is checked. If it exists and specifies a valid OS/390 user ID, then the SMB requestor is mapped to that user ID. Otherwise, the request is denied.

6.6 Batch job submission

There are many ways to submit batch jobs to Infoprint Server and its components NetSpool and IP PrintWay from z/OS, as follows:

- ▶ Printing data sets from batch jobs using the AOPPRINT JCL procedure
- ▶ Printing data sets from batch jobs by IP PrintWay using JCL
- ▶ Submitting batch jobs from VTAM applications to NetSpool

6.6.1 Batch jobs using the AOPPRINT procedure

The AOPPRINT JCL procedure is provided in SYS1.PROCLIB. This procedure lets a user submit data sets to be printed by a printer defined in the Printer Inventory. It allows the user to use the Infoprint Server Transforms.

Because the AOPPRINT JCL procedure provides the full spooling capabilities of Infoprint Server, you can specify job attributes. Print Interface validates jobs that are submitted with the AOPPRINT JCL procedure.

Note: For customization of the AOPPRINT JCL procedure, see “AOPPRINT JCL procedure customization” on page 51.

Submit data set to AFP printer

The following JCL submits a LIST3820 data set to an AFP printer that is defined in the Printer Inventory named POKEXFD.

```
//ROGAOPP JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//PRINT1 EXEC AOPPRINT,PRINTER='POKEXFD'
//SYSIN DD DSNAME=AXEL.R2.AOFRSA08.LIST3820,DISP=SHR
```

Figure 6-44 Batch job using AOPPRINT JCL procedure

When the job executes, the actual conversion of the JCL by the converter is shown in Figure 6-45 on page 165. The data set is passed to the Print Interface by the **lp** command.


```
//ROGAOPP JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//PRINT1 EXEC AOPPRINT,PRINTER='POKEXFD'
//SYSIN DD DSN=AXEL.R2.AOFRSA08.LIST3820,DISP=SHR
1 //ROGAOPP JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
2 //PRINT1 EXEC AOPPRINT,PRINTER='POKEXFD'
XX*-----
XX* AOPPRINT - OS/390 Print Server batch print procedure
XX*-----
3 XXAOPPRINT PROC PRINTER='p1',OPTIONS=,OUTCLASS='*',ERRCLASS='*'
4 XXLP EXEC PGM=AOPBATCH,
XX PARM='/lp -d &PRINTER -o "&OPTIONS" //DD:SYSIN'
IEFC653I SUBSTITUTION JCL - PGM=AOPBATCH,PARM='/lp -d POKEXFD -o "" //
5 XXSTDOUT DD SYSOUT=&OUTCLASS
IEFC653I SUBSTITUTION JCL - SYSOUT=*
6 XXSTDERR DD SYSOUT=&ERRCLASS
XX* STDENV may point to a dataset containing environment variables.
XX* Builtin values will work for the default installation.
XX*STDENV DD DSN=USERID.JCL(ENVVARS),DISP=SHR
IEFC653I SUBSTITUTION JCL - SYSOUT=*
7 //SYSIN DD DSN=AXEL.R2.AOFRSA08.LIST3820,DISP=SHR
```

Figure 6-45 AOPPRINT execution JCL from the batch job

JES3 (E)JES job display

Using (E)JES with JES3, the output data sets that are created are shown in Figure 6-46.

Navigate Filter View Options Help									
DSSTAT ROGAOPP JOB15426 49 Records								Row 1 of 5	
Command ==>								Scroll ==> HALF	
Cmd	DDName	/StepName	ProcName	Que*	C	Pri	Cop	Hld	Destination
----->									
JESMSGLG				WTR	A	1	1		ANYLOCAL
JESJCL				WTR	A	1	1		ANYLOCAL
JESYSMSG				WTR	A	1	1		ANYLOCAL
STDOUT	PRINT1	LP		WTR*	A				
STDERR	PRINT1	LP		WTR	A	1	1		ANYLOCAL
***** Bottom of Data *****									

Figure 6-46 Output data set created by AOPPRINT batch job

JES2 SDSF job display

Using SDSF with JES2, the output data sets that are created are shown in Figure 6-47 on page 166

Display Filter View Print Options Help									

SDSF JOB DATA SET DISPLAY - JOB ROGAOPP (JOB14281)						DATA SET DISPLAYED			
COMMAND INPUT ==>						SCROLL ==> PAGE			
PREFIX=ROG* DEST=LOCAL OWNER=*									
ACTION=//-Block,=-Repeat,+-Extend,Q-Outdesc,S-Browse,V-View,X-Print									
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	PAGE
	JESMSG LG	JES2		2	ROGERS	A	LOCAL	22	
	JESJCL	JES2		3	ROGERS	A	LOCAL	17	
	JESYSMSG	JES2		4	ROGERS	A	LOCAL	18	
	STDOUT	PRINT1	LP	101	ROGERS	A	LOCAL	1	

Figure 6-47 Output data set created by AOPPRINT batch job

STDOUT data set

The STDOUT data set contains the messages returned from the Print Interface for both JES2 and JES3, as follows:

AOP007I Job 67287 successfully spooled to POKEXFD.

Job 67287 is the PS number (PS067287) of the job ID assigned to the submitted data set.

STDERR data set

If the printer, POKEXFD, has not been defined or if any other error occurred, the STDERR data set contains the error messages. Example:

lp: AOP001E Printer POKEXFD is not defined.

6.6.2 Batch job submission to IP PrintWay

To transmit your output from a batch job to an IP PrintWay printer, do one of the following in your JCL on the DD SYSOUT statement:

1. Use DEST= printrname, CLASS= IP PrintWay selection class, or FORMS=.
2. Use the OUTPUT JCL statement.

IP PrintWay FSA printers select output from the JES spool by using the work-selection criteria that your system administrator has defined for the IP PrintWay FSA writer; for example, selection by output class.

Using DEST=

The submitting user needs to know the following, which the administrator has defined:

- ▶ IP PrintWay output selection class
- ▶ IP PrintWay printer definition names in the Printer Inventory

The user can submit the JCL shown in Figure 6-48 on page 167, where SYSOUT class J is selected by IP PrintWay printers and **POKEPS** is defined in the Printer Inventory. The IP address and queue name of the printer is in the protocol entry of this printer definition.

```
//ROGERSY JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD SYSOUT=J,DEST=POKEPS
//SYSUT1 DD DISP=SHR,DSN=ROGERS.$$$$PAUL.PSBIN
//SYSIN DD DUMMY
//
```

Figure 6-48 JCL to print to an IP PrintWay printer

Using the OUTPUT JCL statement

To submit a print data set to an IP PrintWay printer for which your system administrator has not created a printer definition, specify a value for the DEST=IP parameter on the OUTPUT JCL statement, as shown in Figure 6-49. This value is the IP address or host name of the system to which the printer is attached.

When you specify DEST=IP, you *must* also specify a value for either the PRTQUEUE parameter or the PORTNO parameter.

```
//ROGERSZ JOB (POK,999),MSGCLASS=A,NOTIFY=ROGERS
//R1 OUTPUT DEST='IP:9.12.2.4',PRTQUEUE='afccu2',
// RETAINS=FOREVER
//PRINT EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD SYSOUT=J,OUTPUT=(*.R1)
//SYSUT1 DD DISP=SHR,DSN=ROGERS.PRAGINFO.PS
//SYSIN DD DUMMY
//
```

Figure 6-49 Printing to an IP PrintWay printer with OUTPUT JCL statement

Resubmit for filtering

When you submit a batch job that creates an output data set to print on an IP PrintWay printer, the data resides on the JES spool to be selected by an FSA writer. When the data set is selected by IP PrintWay for processing and the Resubmit for filtering option is selected in the Printer Inventory entry, as shown in Figure 6-50 on page 168, IP PrintWay sends the data set to Print Interface. Print Interface can transform the data format to a format supported by the printer specified by the user that resides in the Printer Inventory.

Edit		Processing	
Command ==> _____			
Printer definition name . <u>NP24</u>			
		More: *	
Document code page . . _____			
Printer code page. . . <u>ISO8859-1</u>			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
/ Line data	<u>afp2pcl.dll %filter-options</u>		(extend)
/ M0:DCA-P	<u>afp2pcl.dll %filter-options</u>		(extend)
/ PostScript	_____		(extend)
/ Text	<u>aopfiltr.so</u>		(extend)
/ PCL	_____		(extend)
/ PDF	_____		(extend)
- SAP	_____		(extend)
/ Other	<u>afp2pcl.dll %filter-options</u>		(extend)
/ Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top . . .	Bottom . .	Left . .	Right . .
Line length . . .		Page length . .	
Tabs: Vertical . .			(extend)
Horizontal . .			(extend)
IP PrintWay Line-to-Text Conversion:			
/ Pagination			
Margins: Top . .		Bottom . .	
Page height . . .	<u>58</u>		

Figure 6-50 Resubmit for filtering selected for this printer

In Figure 6-51 on page 169, a batch job creates an AFP data set on spool. The AFP data set is selected by an IP PrintWay FSA writer and checks the Printer Inventory for the IP destination. The Resubmit for filtering option is set and IP PrintWay sends the AFP data set to Print Interface. Print Interface checks the data set format and checks the format type in the Printer Inventory entry shown in Figure 6-50. Since a filter indicates to transform the AFP data set to PCL, the data is passed to the transform manager to select the AFP2PCL transform to convert the data. The PCL data set is now dynamically allocated to the spool. IP PrintWay selects the PCL data set and sends it via the IP address to the PCL printer.

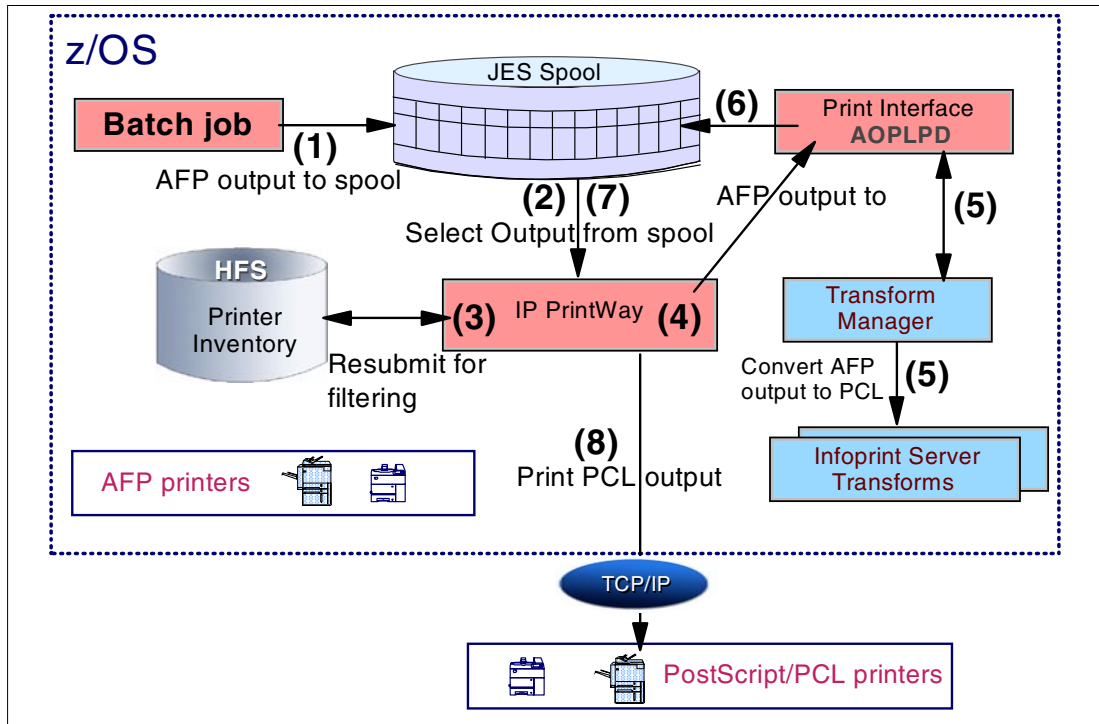


Figure 6-51 Output processed with the Resubmit for filtering option

The batch job JOBATCH, submitted in Figure 6-51, can be seen in Figure 6-52 on page 170, which is the IP PrintWay message file. Once the AFP output is on the spool, FSA writer PRT1 selects the AFP output. The ANFM601I message shows that the AFP output is sent to the Print Interface where the host IP address is 9.12.6.121, for Print Inventory printer NP24.

Once the AFP output is transformed to PCL, IP PrintWay selects the PCL data set in the next message.

Note: In message ANFM700I, notice that the jobname JOBATCH has been lost and is replaced by the user ID ROGERS.

In the second message, ANFM601I, the IP address 9.12.2.8 is the PCL printer and the queue name is PASS to which IP PrintWay sends the output. The last two messages are releases to the two output data sets that were on the JES spool.

```

2001062713375166-ANFM700I Data set:
      WTSCPLX9.ROGERS.JOBATCH.JOB15443.D000000A.?
      The data set has been acquired by PRTWAY
2001062713375272-ANFM601I Data set:
      WTSCPLX9.ROGERS.JOBATCH.JOB15443.D000000A.?
      The data set was successfully transmitted to host and queue:
      9.12.6.121
      NP24
2001062713375282-ANFM700I Data set:
      WTSCPLX9.ROGERS.ROGERS.JOB15330.D000000E.#0000A##
      The data set has been acquired by PRTWAY
2001062713375952-ANFM601I Data set:
      WTSCPLX9.ROGERS.ROGERS.JOB15330.D000000E.#0000A##
      The data set was successfully transmitted to host and queue:
      9.12.2.8
      PASS
2001062713385811-ANFM604I Data set:
      WTSCPLX9.ROGERS.JOBATCH.JOB15443.D000000A.?
      The data set is being released to JES
2001062713395826-ANFM604I Data set:
      WTSCPLX9.ROGERS.ROGERS.JOB15330.D000000E.#0000A##
      The data set is being released to JES

```

Figure 6-52 IP PrintWay message file for batch job JOBATCH

Thus, IP PrintWay can transform your data to a format that the printer can accept before transmitting the data to the printer. This function lets you, for example, print AFP and line data to printers that can print PCL or PostScript data.

Note: To use this function, you must have the Infoprint Server Transforms (Program Number 5697-F1) installed. Your administrator must also configure the appropriate data transforms in the printer definition in the Infoprint Server Printer Inventory.

You can transform AFP and line data to any of the following data streams:

- ▶ Printer Control Language (PCL)
- ▶ Portable Data Format (PDF)
- ▶ PostScript

6.7 z/OS UNIX users using the lp command

The **lp** command sends data to the Print Interface containing one or more files to print on a specified or default printer. If you do not specify any files on the command line, or if you specify a filename of -, **lp** prints from standard input. The files can be:

- ▶ MVS data sets, such as partitioned data sets or sequential data sets
- ▶ Hierarchical file system (HFS) files
- ▶ Lists of printable files

6.7.1 z/OS UNIX print commands

The z/OS UNIX commands **lp**, **lpstat**, and **cancel** have been modified by the Infoprint Server product to communicate with the Print Interface.

Where:

- lp** Sends a job to a printer in the Printer Inventory
- lpstat** Queries printers, locations, and status of jobs in the Printer Inventory
- cancel** Cancels a print job sent via the Print Interface

They are in the HFS in the following path:

/usr/lpp/Printsrv/bin

See “Customization of UNIX shell print commands” on page 37 for more information. When you are in an OMVS shell session, to make sure you have access to the modified commands, issue the command **echo \$PATH**, shown in Figure 6-53. The modified commands’ directory path must be before the ***/bin*** directory where the unmodified commands exist.

```
ROGERS @ SC43: />echo $PATH
/usr/lpp/java18/J1.1/bin:/usr/lpp/Printsrv/bin:/bin:.
ROGERS @ SC43: />
```

===>

INPUT

Figure 6-53 Use of echo \$PATH to determine command path

Query printers in the Printer Inventory

A z/OS UNIX user can query the status of all printers defined in the Printer Inventory as follows:

lpstat -a

Returned to the user’s screen, shown in Figure 6-54 on page 172, are printers in our Inventory (not all printers are shown due to the total number).

The other options for the **lpstat** command are:

- d** Query default printer Example: **lpstat -d**
- o** Query specified printer and jobs Example: **lpstat -o poke**
- p** Query specified printer Example: **lpstat -p poke**
- t** Query all printers and jobs Example: **lpstat -t**
- u** Query all printers and jobs by user ID Example: **lpstat -u ROGERS**
- a** Query names and locations of all printers Example: **lpstat -a**

color	0	2C-16	Infoprint color 8			
colord	0	2C-16	Infoprint color 8			
colort	0	2C-16	Infoprint color 8			
colorx	0	2C-16	EBCDIC PassThru to IAZFSS			
colory	0	2C-16	ASCII PassThru to IAZFSS			
ippt	0	N/A	N/A			
poke	0	2c-16	3130			
pokeb	0	2c-16	3130			
pokep	0					
pokepcl	0	2c-16	3130			
pokepdf	0	2c-16	POKNP24			
pokeps	0	Room 2c-16 B ...	3130 Postscript printer			
pokeret	0	2c-16	3130			
poketest	0	Room 2c-16 B ...	3130 Postscript printer			
pokexpdf	0		PDF to AFP			
poknp24	0	2c-16	POKNP24			
BOBS	0	Raleigh	NP24 - Has an IPDS card			
DFLTNTRY	0	Sysplex	IP: Defaults			
FIJVBIN	0	Finland	Proprinter III			
FIJVNBIN	0	VAINI in Finland	Proprinter III			
FINPRT01	0	Finland	A Printer			
FIP17D2	0	IBM Finland ...	HENP17D2 (very slow)			
IAZFSS	0	ITS0 Sysplex	Shredder			
IAZFSSB	0	ITS0 Sysplex	Shredder			
IAZFSSL	0	ITS0 Sysplex	Shredder			
IPDP0K	0	2C16 Poughke ...	AFP printer			
IPDSB0B	0	Raleigh	Bob's AFP printer			
IPDSP0K	0	2C?? Poughke ...	AFP printer			
==>						
MORE...						
ESC=	1=Help	2=SubCmd	3=HlpRetrn	4=Top	5=Bottom	6=TS0
	7=BackScr	8=Scroll	9=NextSess	10=Refresh	11=FwdRetr	12=Retrieve

Figure 6-54 A display of names and locations of printers in the Printer Inventory

Send a print request

To send a print request to print an MVS data set owned by the submitting user ID to the Print Interface, issue the following:

```
ROGERS @ SC43: />lp -d pokeps //test.jcl
AOP007I Job 67289 successfully spooled to pokeps.
ROGERS @ SC43: />

==>
```

INPUT

Figure 6-55 z/OS UNIX user submitting a print request

The job ID returned in the message is PS067289.

Cancel a print request

To cancel a print request that has not yet been printed, issue:

```
cancel 67289
```

6.8 TSO/E and LPR protocol

You will find HELP text for the z/OS TCP/IP TSO **lp** and related commands in the hlq.SEZAHLP data set. The commands are as follows:

- LPQ** The LPQ command allows you to request a list of the printer queue on a remote printer from the LPD server controlling that printer.
- LPR** The LPR command allows you to print on a remote printer.

- LPRM** The LPRM command allows you to remove a job from the printer queue on a remote host.
- LPRSET** The LPRSET command sets the default printer and host name to be used when this information is not included in the line printer commands (LPR, LPQ, and LPRM).

6.8.1 TSO/E ISPF command line

From the ISPF command line, you may enter the following command to print data set **rogers.test.jcl**:

```
lpr test.jcl (HOST 9.12.6.121 PRINTER pokeps
```




Infoprint Server Transforms

This chapter describes Infoprint Server Transforms, which is designed to work with Infoprint Server. Infoprint Server Transforms gives you the flexibility to print an increasing variety of outputs on a wide array of printers by providing filters that transform files into the format required for printing on the desired printer.

This chapter contains the following:

- ▶ What are Infoprint Server Transforms
- ▶ Transform Manager customization
- ▶ Data stream transform customization
- ▶ Transform classes
- ▶ PCL to AFP transform customization
- ▶ PostScript and PDF to AFP customization
- ▶ AFP to PCL, PDF, and PostScript customization

7.1 Infoprint Server Transforms

Infoprint Server Transforms is an optional product (Program Number 5697-F51) designed to work with Infoprint Server. Infoprint Server Transforms gives you the flexibility to print an increasing variety of outputs on a wide array of printers. Infoprint Server Transforms provides data transforms that convert the following data streams to Advanced Function Presentation (AFP) format for printing on IBM AFP printers:

- ▶ Printer Control Language (PCL) 5e to AFP
- ▶ PostScript Language Level 3 to AFP
- ▶ Portable Document Format (PDF) 1.2 to AFP
- ▶ SAP Output Text Format (OTF) Versions 1 and 2; and SAP Advanced Business Application Programming (ABAP) Versions 1 and 2 to AFP

The AFP to PCL, AFP to PDF, and AFP to PostScript Transforms are priced features of Infoprint Server Transforms. These features can transform files in AFP or line data format to any of the following formats for printing on PCL or PostScript printers, or posting on the Web:

- ▶ AFP to Printer Control Language (PCL) 5, 5e, or 5c (color)
- ▶ AFP to PostScript Language Level 2 (monochrome or color)
- ▶ AFP to Portable Document Format (PDF) 1.2 (monochrome or color)

Infoprint Server Transforms also provides the following features:

- ▶ A Kanji AFP Print feature provides the Heisei Kaku Gothic W5 and Heisei Mincho W3 fonts. Once these fonts are installed on the z/OS system, the PDF to AFP and the PostScript to AFP transforms can use these fonts and map some other commonly used Japanese fonts, including Ryumin-Light and Gothic BBB-Medium, to these two Heisei fonts.
- ▶ A Coax Printer Support feature lets IP PrintWay print to VTAM-controlled printers. The Coax Printer Support feature is a priced feature of Infoprint Server Transforms. This feature lets you print to VTAM-controlled printers that are defined to VTAM as LU0, LU1, or LU3 printers. The supported output data streams are Data Stream Compatibility/Data Stream Extended (DSC/DSE) and SNA Character String (SCS). The IP PrintWay component of Infoprint Server uses the Coax Printer Support feature to print to VTAM-controlled printers.

7.1.1 Print Interface and transforms

When a print document has one of the data formats for which a transform is provided, Print Interface can automatically call the appropriate transform before it writes data to the JES spool. If you want Print Interface to automatically call a transform, the administrator must name the transform DLL, which is called a *filter*, in the printer definition for the target printer.

Users can also invoke the transform DLLs directly from the z/OS UNIX command line without printing the output using a command, as shown in Figure 7-1 on page 177 in Step (1a). The transform DLL can also be invoked when users use the **lpr** command to pass print files to the Print Interface LPD, as shown in Step (1).

The Transform Manager component of Infoprint Server manages the transforms that are implemented as *daemons*, shown at Step (2). The Transform Manager starts and stops the transform daemons using configuration information specified by the administrator. The administrator can limit the number of transform daemons that are active at a time. The Transform Manager component manages the PCL to AFP, PostScript to AFP, PDF to AFP, AFP to PCL, AFP to PDF, and AFP to PostScript transforms.

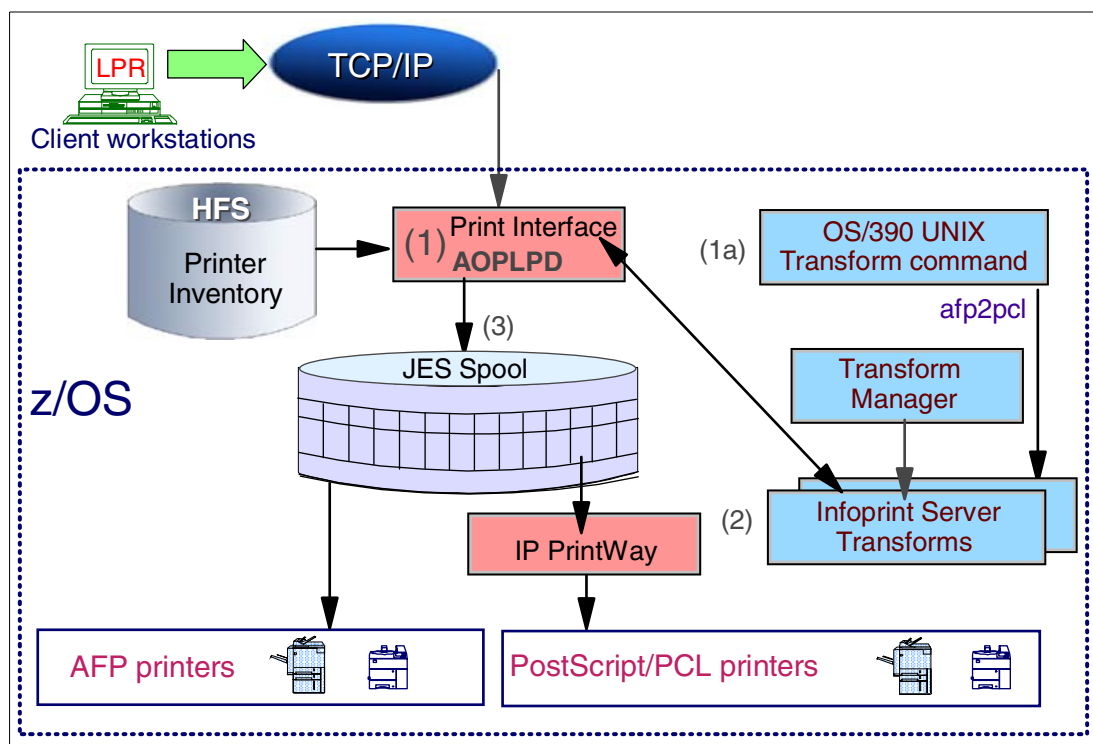


Figure 7-1 Users invoking Infoprint Transforms for printing passed data sets

Remote transform support

You can also request remote transforms for the PCL, PostScript, or PDF data to AFP format on an AIX or Windows NT system. To transform data remotely, you do not need to customize Infoprint Server Transforms or the Transform Manager. However, Infoprint Manager for Windows NT or Infoprint Manager for AIX (or PSF for AIX) must be installed on a Windows NT or AIX system, and the administrator must specify the remote transform filter in printer definitions in the Printer Inventory.

7.1.2 Enable the Infoprint Server Transforms

Before you can use the transforms, all installations must dynamically enable Infoprint Server Transforms. To do this, add the following entry to your active IFAPRDxx member of SYS1.PARMLIB if it has not already been added:

```
PRODUCT OWNER('IBM CORP')
NAME('INFOPRINT XFORMS')
ID(5697-F51)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('TRANSFORM TO AFP')
STATE(ENABLED)
```

You should use the **DISPLAY PROD** command to display information about products that have been registered or display the product enablement policy. You can also use the command to determine the state (enabled, disabled, not defined, or not found) that, according to the current policy, exists for a specific product or set of products. Figure 7-2 shows the command; only the Infoprint Server products are shown.

D PROD,STATE,ALL					
IFA111I	14.38.42	PROD	DISPLAY	956	
S	OWNER	NAME	FEATURE	VERSION	ID
.....					
E	IBM CORP	INFOPRINT XFORMS	TRANSFORM TO AFP	*.*.*	5697-F51
.....					
E	IBM CORP	OS/390	INFOPRINT SERVER	*.*.*	5647-A01
E	IBM CORP	OS/390	JES3	*.*.*	5647-A01

Figure 7-2 Display of Infoprint Server products from IFAPRDxx PARMLIB member

For more information on how to use z/OS dynamic enablement, refer to *z/OS MVS Product Management*, SA22-7603.

7.2 Transform Manager customization

The Transform Manager component of Infoprint Server manages transforms that are implemented as daemons. The configuration file **aopxfd.conf** must be customized, and the **aopd.conf** configuration file should include the specification of the xfd daemon.

7.2.1 Edit the aopd.conf file

To start the Transform Manager, you must add or edit the start-daemons attribute in the Infoprint Server configuration file **aopd.conf** before issuing the **aopstart** command. Enclose the values in braces. By default, only the Printer Inventory Manager and the LPD start; therefore, this attribute is required to start the Transform Manager daemon.

```
start-daemons = { xfd }
```

7.2.2 Create the transform configuration file (aopxfd.conf)

The transform configuration file contains information that the Transform Manager uses to manage transform daemons. To create the transform configuration file, copy the sample configuration file provided in **/usr/lpp/Printsrv/samples/aopxfd.conf** to **/etc/Printsrv/aopxfd.conf**, as shown in Figure 2-8 on page 32. This is the default location. You can choose to copy the configuration file into another location; if you do so, specify the full path name of the configuration file in the **AOPXFD_CONF** environment variable.

For example, use the following z/OS UNIX command to copy the sample configuration file:

```
cp /usr/lpp/Printsrv/samples/aopxfd.conf /etc/Printsrv/aopxfd.conf
```

In the transform configuration file, you can specify options and environment variables that affect the transforms. These options and environment variables apply whether Print Interface invokes the transform automatically or whether the user invokes the transform with a transform command from z/OS UNIX.

To edit the file, you can use the TSO/E **0EDIT** command, the ISHELL, or the z/OS UNIX **oedit** command. For example, enter the following z/OS UNIX command:

```
oedit /etc/Printsrv/aopxfd.conf
```

Note: If you change the transform configuration file while the Transform Manager is running, you must restart the Transform Manager to pick up the changes.

Restart of Transform Manager

If you make changes to the **aopxfd.conf** file, issue the following commands to pick up the changes:

Use the **aopstop -d xfd** command to stop the Transform Manager

Use the **aopstart** command to restart it

7.3 Data stream transforms customization

This Transform Manager daemon manages other transform daemons, which transform data from one format to another. The Transform Manager daemon starts and stops transform daemons that are configured in the **aopxfd.conf** configuration file, as shown in Figure 7-3.

The parent process of the Transform Manager daemon is the Printer Inventory Manager daemon. The Transform Manager daemon is the parent of all the individual transform daemons that are started.

The number of data stream transform daemons that are started is controlled by the min-active parameter, shown in Figure 7-4 on page 182, in the **aopxfd.conf** file for each transform type.

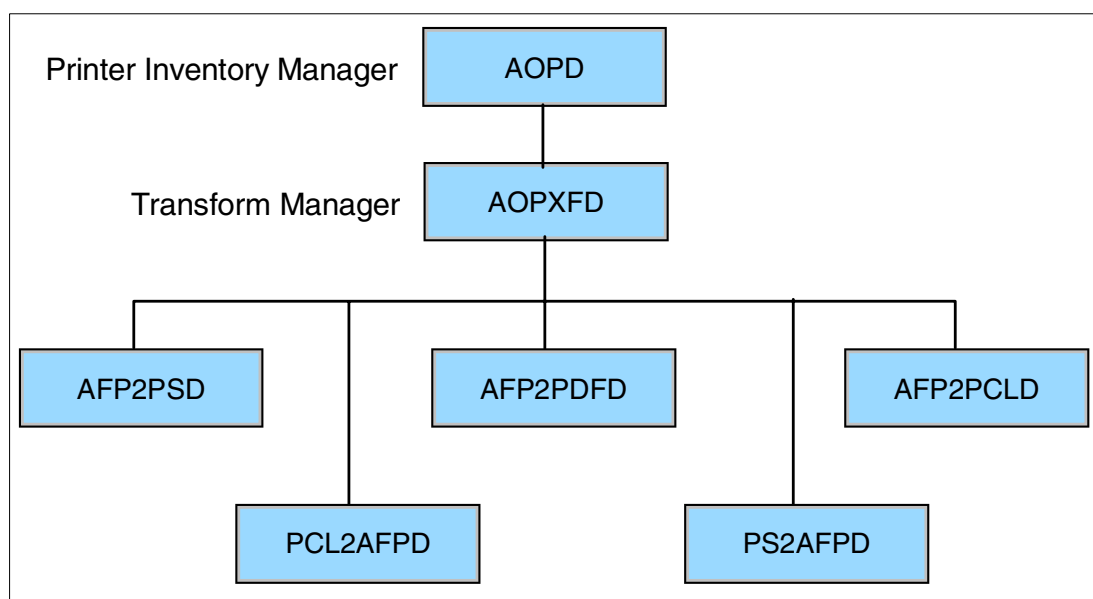


Figure 7-3 z/OS UNIX System Services processes for data stream transforms

7.3.1 Transform daemon messages

The transform daemons issue messages to message logs located in the xfd subdirectory in the base directory defined in the Infoprint Server configuration file (**aopd.conf**); the default base directory is **/var/Printsrv/**.

The Transform Manager, a component of Infoprint Server, controls the transform daemons provided with Infoprint Server Transforms for OS/390. The Transform Manager starts and stops the transform daemons using configuration information specified by the administrator. For example, an administrator can limit the number of transform daemons that are active at a time. Each instance of a transform daemon writes messages to its own message log. The message log is cleared when you restart the Transform Manager.

The file names of the message logs have the format:

transform.n.stderr

Where:

transform Is the name of the transform with the error. The transform name includes the name of the transform class if one is defined; for example, pcl2afp or pcl2afp_letter_300.

n Is the instance of the transform daemon.

7.4 Establish security for the ps2afpd daemon

Security checking performed in the transform requires that the user identifier (UID) of the executable file for the PostScript and PDF to AFP transform, **ps2afpd**, *not be 0* (zero). When it is installed, ps2afpd has a UID of 0; therefore, you must change the owner of the file. The new owner must have a UID that is not 0 and not the default UID; also, the set-user-ID flag for the file must be turned on.

First use the Resource Access Facility (RACF), or another program that follows system authorization facility (SAF) protocol, to create a user and group profile for the owner of ps2afpd. Then change the owner of ps2afpd and turn on the set-user-ID flag. Follow these steps:

1. Define a group

The group profile must have an OMVS segment and a group identifier (GID). You can use any group name. IBM recommends that you do not give this group any authority to the z/OS file system.

For example, the following RACF command defines group NOGROUP; for nogroup-gid, specify an integer that is different from other GIDs in your installation:

```
ADDGROUP (NOGROUP) OMVS(GID(nogroup-gid))
```

2. Define a user as an z/OS UNIX user

This user will be the owner of ps2afpd. The transform, as well as PostScript jobs being transformed, run with the UID of this user.

The user profile must have an OMVS segment. Its UID must *not be 0* and *not be the default UID*, which is defined in RACF in the BPX.DEFAULT.USER profile in the FACILITY class. You can use any user name; for example, you can use NOBODY.

IBM recommends that you do not give this user authority to the z/OS file system; however, if any PostScript jobs to be transformed require access to certain files, you can give this user access to the required files or connect this user to another group (or groups) that has access to the required files.

For example, the following RACF command defines user NOBODY; for nobody-uid, specify an integer that is different from other UIDs in your installation:

```
ADDUSER (NOBODY) OMVS(UID(nobody-uid))
```

3. Connect the user to the group

Connect the user defined in step 2 to the group defined in step 1. For example, the following RACF command connects user NOBODY to the NOGROUP group:

```
CONNECT (NOBODY) GROUP(NOGROUP)
```

4. Assign the user as the owner of file **ps2afpd**

Use the **chown** command to assign the user defined in step 1 as the owner of **ps2afpd**. For example, type the following command on the z/OS UNIX command line to assign user NOBODY as the owner:

```
chown NOBODY /usr/lpp/Printsrv/bin/ps2afpd
```

5. Turn the set-user-ID flag on for file **ps2afpd**

The **chown** command turns *off* the set-user-ID flag; therefore, you must use the **chmod** command to turn this flag *on* again. For example, enter the following command on the z/OS UNIX command line:

```
chmod u+s /usr/lpp/Printsrv/bin/ps2afpd
```

After performing these steps, you can use the **ls** command to list the owner of **ps2afpd** and to verify that the set-user-ID flag is on. For example, on the z/OS UNIX command line, enter:

```
ls -l /usr/lpp/Printsrv/bin/ps2afpd
```

Assuming that you used the user and group names shown in the previous examples, output from the **ls** command should look like:

```
-rwsr-xr-x 1 NOBODY NOGROUP ...
```

The lowercase letter **s** in the owner permissions section indicates that the set-user-ID flag is on and that the owner has permission to execute the file.

For more information about RACF, refer to *z/OS SecureWay Security Server RACF Security Administrator's Guide*, SA22-7683.

For more information about the **chown**, **chmod**, and **ls** commands, refer to *z/OS UNIX System Services Command Reference*, SA22-7802.

7.5 Transform classes

For each transform, you can optionally define one or more transform classes in the transform configuration file. Transform classes let you transform data with different:

- ▶ Transform options
- ▶ Environment variables

For example, if you want to transform PCL data to AFP format for printers that have different resolutions or paper sizes, you would define a transform class for each combination of printer resolution and paper size and specify the resolution and paper size in environment variables that are specific for the transform class. The sample transform configuration file, */usr/lpp/Printsrv/samples/aopxfd.conf*, defines several sample transform classes, as shown in Figure 7-4 on page 182. The transform classes shown are:

- ▶ legal_240
- ▶ legal_300

```

#-----
# PCL -> AFP: legal size output
#-----
transform pcl2afp_legal_240
  min-active = 1
  max-active = 2
  start-command = pcl2afpd
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_PAGE_HEIGHT -> 14i
    _BPX_JOBNAME -> PCL2AFPD
  }
;

transform pcl2afp_legal_300
  start-command = pcl2afpd
  min-active = 1
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_PAGE_HEIGHT -> 14i
    AOP_RESOLUTION -> 300
    _BPX_JOBNAME -> PCL2AFPD
  }
;

```

Figure 7-4 Sample transform classes from /usr/lpp/Printsrv/samples/aopxfd.conf

7.5.1 Using transform classes

To use a transform class, the administrator, the z/OS UNIX user, or the job submitter must specify the transform class. The administrator specifies the transform class in the printer definition in the **-c** filter option. For example, in the Filter field for the PCL data format, specify as shown in Figure 7-5 on page 183:

```
pcl2afp.dll -c letter_300 %filter-options
```

AOPIPDRR		Processing	
Command ==> _____			
Printer definition name . <u>IPDSP0K</u>			
			More: +
Document code page . . _____			
Printer code page. . . <u>IBM-1047</u>			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
/ Line data			(extend)
/ M0:DCA-P			(extend)
/ PostScript	<u>ps2afp.dll -l 10i -w 7.5i %filter-options</u>		(extend)
/ Text	<u>aopfiltr.so</u>		(extend)
/ PCL	<u>pcl2afp.dll -c letter 300 %filter-options</u>		(extend)
/ PDF	<u>ps2afp.dll -l 10i -w 7.5i %filter-options</u>		(extend)
- SAP			(extend)
/ Other			(extend)
/ Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top . . .	Bottom . . .	Left . . .	Right . . .
Line length . . .		Page length . . .	
Tabs: Vertical . . .			(extend)
Horizontal . . .			(extend)
IP PrintWay Line-to-Text Conversion:			
/ Pagination			
Margins: Top . . .	Bottom . . .		
Page height . . .	<u>58</u>		

Figure 7-5 Processing section panel showing filters

Administrator specification

Figure 7-5 shows the transform command **pcl2afp** with the transform class (**-c**) specification as follows:

```
pcl2afp.dll -c letter_300 %filter-options
```

Z/OS UNIX user

z/OS UNIX user issues this command from the OMVS command line:

```
pcl2afp -o one.afp -c letter_300 one.pcl
```

Job submitter specification

The z/OS UNIX user can specify the transform class in the filter-options job attribute on the **lp** command or in an AOPPRINT JCL procedure example:

```
lp -d IBMNP24 -o "filter-options='-c letter_300'" one.pcl
```

```
//AOPPRINT PROC PRINTER='p1',OPTIONS=,OUTCLASS='*',ERRCLASS='*'
//LP      EXEC PGM=AOPBATCH,
// PARM='/lp -d IBMNP24 -o "filter-options='-c letter_300'" one.pcl'
//STDOUT  DD  SYSOUT=&OUTCLASS
//STDERR  DD  SYSOUT=&ERRCLASS
```

Figure 7-6 AOPPRINT JCL procedure with transform class in filter options

7.5.2 Format of a transform entry

The **aopxfd.conf** file must contain one entry for each transform that will be used. If you want to define transform classes, create a separate entry for each transform class. Each entry consists of a set of attributes, starting with the transform attribute and ending with a semicolon. The general format of a transform entry is shown in Figure 7-7:

```
# comment
transform transformname[_transformclass]
    start-command = "daemon [ option ]..."
    [ environment = {name -> value [ name -> value ]...} ]
    [ maximum-idle-time = seconds ]
    [ minimum-active = number ]
    [ maximum-active = number ]
;
```

Figure 7-7 Format of a transform entry

Where:

transform transformname[_transformclass]

This attribute names the transform and, optionally, the transform class. This attribute is required and must be the first attribute in the entry.

transformname

The name of the transform, for example, pcl2afp or ps2afp

transformclass

The name of a transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters. To use the transform class, the transform class must be specified when the transform is invoked. This parameter is optional.

Default: If this parameter is omitted, this entry is used only if no class is specified when the transform is invoked.

Example: transform pcl2afp_letter_300.

start-command

"daemon [option]" -- The name of the transform daemon and options. Enclose the value in single or double quotation marks if you specify an option. This attribute is required.

daemon

The name of the transform daemon, for example, pcl2afpd or ps2afpd. If the transform daemon is not in a directory identified in the PATH environment variable, then specify the full directory path name of the daemon.

option

One or more options supported by the transform daemon.

[m nnn{KIM}] - The number of bytes of memory the transform daemon can use to perform transforms. Specify the number of bytes in either kilobytes or megabytes. The amount required depends on the compression type, the image being compressed, fonts, and so forth.

environment

{name -> value [name -> value]...} -- Environment variables that define the transform environment. Enclose the environment variables in braces. The values in these environment variables override values with the same names that were set when the **aopstart** command was issued.

Example: environment = {AOP_RESOURCE_PATH -> /usr/lpp/Printsrv/ps2afp}

You can define the **_BPX_JOBNAME** environment variable in each transform entry to assign a different job name to each class of

transform daemon. By default, the job name is the user ID of the user who starts Infoprint Server. Assigning a different job name to each class of transform daemon lets the operator manage the transform daemons more effectively. The job name can be 1 to 8 alphanumeric characters. Incorrect job names are ignored.

The job name is changed only if the user who starts Infoprint Server has either superuser authority or READ permission to the BPX.JOBNAME FACILITY class profile. If these conditions are not met, this environment variable is ignored. For more information about the `_BPX_JOBNAME` variable, refer to *z/OS UNIX System Services Planning*, GA22-7800.

maximum-idle-time The number of seconds before the Transform Manager shuts down an idle transform daemon and system resources are freed. This attribute is optional; however, to avoid having many transform daemons active, consider specifying either this attribute or the maximum-active attribute (or both). The number you specify must be greater than 0.

Default: An idle transform daemon is not shut down.

minimum-active The number of transform daemons that the Transform Manager starts. Also the minimum number of transform daemons that the Transform Manager keeps active (that is, not shut down) even when the maximum-idle-time expires for an idle transform daemon. This attribute is optional.

Default: minimum-active = 0

maximum-active The maximum number of transform daemons that the Transform Manager keeps active. When this number is reached, the Transform Manager does not start a new transform daemon to perform the transform; therefore, the transform request waits until a transform daemon is available. This attribute is optional; however, to avoid having many transform daemons active, consider specifying either maximum-active or maximum-idle-time, or both. The number you specify must be greater than 0 and greater than or equal to the number specified in minimum-active.

Default: No maximum number; transform daemons are started when needed.

Transform configuration file syntax rules

When editing the transform configuration file, follow these syntax rules:

- ▶ End the last attribute in a transform entry with a semicolon.
- ▶ Use lowercase characters for the attribute names and uppercase or lowercase characters for the attribute values. Code blank characters before or after the equal sign if desired.
- ▶ Use uppercase characters for the environment variables.
- ▶ If a value contains blank characters or special characters (such as `{ } -> = ,`), enclose the value in single or double quotation marks.
- ▶ Start comments with: `#`.
- ▶ Include blank lines if desired.

Set environment variable

The `AOPXFD_CONF` environment variable affects the Transform Manager. You can set this environment variable in the */etc/profile* file.

The **AOPXFD_CONF** variable sets the full name of the transform configuration file. If the transform configuration file is in */etc/Printsrv/aopxfd.conf*, you do not need to set this environment variable.

7.6 Customize the PCL to AFP transform

If you want Print Interface to automatically call the PCL to AFP transform before writing data to the JES spool, the administrator must specify the transform DLL in the Filter field of the printer definitions in the Printer Inventory, as shown in Figure 7-5 on page 183. In the printer definition, the administrator can also specify transform options, such as the type of AFP images to create and the name of the transform class to use. The user can specify the same transform options on the **pc12afp** command and in the filter-options when submitting a print request.

Refer to the following publications for more information on Infoprint Server Transforms:

- ▶ *z/OS Infoprint Server Operation and Administration*, S544-5745 for information about how to specify the transform DLL and transform options in a printer definition.
- ▶ *z/OS Infoprint Server User's Guide*, S544-5746 for information about the **pc12afp** command and the transform options.

The transform configuration file */etc/Printsrv/aopxfd.conf* contains information that the Transform Manager uses to manage the PCL to AFP transform, and other transforms as well.

7.6.1 Specify transform classes for the PCL to AFP transform

You can define transform classes in the configuration file if you want to transform data for printers with different resolutions and different paper sizes. For example, you can define one transform class to transform data for 240-pel printers and another to transform data for 300-pel printers. For each transform class, you create a separate entry in the transform configuration file. In each entry, you can specify different resolutions and paper sizes in environment variables.

The sample transform configuration file, */usr/lpp/Printsrv/samples/aopxfd.conf*, shows how to define several different PCL to AFP transform classes.

Create a PCL to AFP transform from a default entry from the */samples/aopxfd.conf* file and create separate entries for different PCL to AFP transform classes, as shown in Figure 7-8 on page 187.

Calculate storage requirements

Use the following algorithm or refer to the following table to determine the amount to specify initially; if the transform fails due to lack of memory, then specify a larger amount of storage.

```
((width in pixels / 8) * height in pixels) * 3  
Width in pixels = (width in inches * resolution)  
Height in pixels = (height in inches * resolution)
```

Table 7-1 on page 187 shows the amount of storage for different paper sizes and resolutions (240, 300, and 600 pixels per inch), using the previous algorithm.

Table 7-1 *pcl2afp storage requirements for different paper sizes and resolutions*

Paper Size	Resolution 240	Resolution 300	Resolution 600
Letter	2.0M	3.2M	12.6M
Legal	2.6M	4.0M	16.1M
A4	2.1M	3.3M	13.1M

The example in Figure 7-8 shows two sample entries in the **aopxfd.conf** configuration file as follows:

- The first entry is for the PCL to AFP transform. Because the transform attribute does not specify a transform class, the Transform Manager uses this entry when no transform class is specified when the transform is invoked.

This entry sets environment variables for letter size paper (8.5 inches wide, 11 inches high) and printers with a resolution of 240 pixels per inch.

- The second entry is also for the PCL to AFP transform. The Transform Manager uses this entry when transform class `letter_300` is specified when the transform is invoked.

This entry sets environment variables for letter size paper (8.5 inches wide, 11 inches high) and printers with a resolution of 300 pixels per inch.

```
transform pcl2afp
    start-command = "pcl2afpd -m 4M"
    environment = {
        AOP_PAGE_HEIGHT -> 11i
        AOP_PAGE_WIDTH -> 8.5i
        AOP_RESOLUTION -> 240
        AOP_HORIZONTAL_MARGINS -> 0.167i
        AOP_VERTICAL_MARGINS -> 0.167i
    }
    minimum-active = 1
    maximum-active = 2
    maximum-idle-time = 300 # 5 minutes
;
transform pcl2afp_letter_300
    start-command = pcl2afpd
    environment = {
        AOP_PAGE_HEIGHT -> 11i
        AOP_PAGE_WIDTH -> 8.5i
        AOP_RESOLUTION -> 300
        AOP_HORIZONTAL_MARGINS -> 0.167i
        AOP_VERTICAL_MARGINS -> 0.167i
    }
    minimum-active = 1
    maximum-active = 2
    maximum-idle-time = 300 # 5 minutes
;
```

Figure 7-8 *Two entries created in the aopxfd.conf file for PCL to AFP transforms*

Environment variables

The environment variables define the environment for the PCL to AFP transform. Enclose the environment variables in braces. The values in these environment variables override values with the same names that were set when the **aopstart** command was issued. The following environment variables used are optional and are specified as follows:

AOP_PAGE_HEIGHT Page height in inches (i) or millimeters (m).

Default: AOP_PAGE_HEIGHT -> 11i

AOP_PAGE_WIDTH Page width in inches (i) or millimeters (m).

Default: AOP_PAGE_WIDTH -> 8.5i

AOP_RESOLUTION Output resolution in pixels per inch.

Default: AOP_RESOLUTION -> 240

AOP_HORIZONTAL_MARGINS Left and right margins in inches (i) or millimeters (m).

Default: AOP_HORIZONTAL_MARGINS -> 0.167i

AOP_VERTICAL_MARGINS Top and bottom margins in inches (i) or millimeters (m).

Default: AOP_VERTICAL_MARGINS -> 0.167i

See “Format of a transform entry” on page 184” for an explanation of the following entries in **aopxfd.conf**:

- ▶ maximum-idle-time = seconds
- ▶ minimum-active = number
- ▶ maximum-active = number

Starting the Transform Manager

When the Transform Manager starts, it starts two types of transform daemons, one for PCL to AFP transforms that do not specify a transform class and another for PCL to AFP transforms that specify transform class letter_300. The Transform Manager keeps no more than two of each type of transform daemon active at any time. The Transform Manager shuts down any transform daemon that is idle and frees system resources after 5 minutes; however, the Transform Manager keeps at least one transform daemon of each type active.

7.7 Customize the PostScript to AFP and PDF to AFP transform

If you want Print Interface to automatically call the PostScript to AFP transform or the PDF to AFP transform before writing data to the JES spool, the administrator must specify the transform DLL in the Filter field of printer definitions in the Printer Inventory. In the printer definition, the administrator can also specify transform options, such as the type of AFP images to create; the resolution, length, and width of the output images; and the transform class to use. The user can specify the same transform options on the ps2afp and pdf2afp commands and in the filter-options job attribute when submitting a print request.

Note: PostScript data is usually in ASCII representation; however, the PostScript to AFP transform can also process PostScript data in EBCDIC representation. Print Interface converts an EBCDIC data stream that starts with %! (X'6C5A') from EBCDIC code page IBM-1047 to ASCII code page ISO8859-1. You cannot customize these code pages.

7.7.1 Storage requirements

The transform requires at least 42 MB of memory. For improved performance, IBM recommends a region size of 64 MB or more. To ensure that sufficient memory is available when you start the transform, do one of the following, depending on how you plan to issue the **aopstart** command to start Infoprint Server:

- ▶ The MAXASSIZE parameter in the BPXPRMxx member of SYS1.PARMLIB determines the system-wide maximum region size for an address space. For example, specify one of the following statements in the BPXPRMxx member:

```
MAXASSIZE(44040192)    /* 42*1024*1024 = 42MB (minimum)    */
MAXASSIZE(67108864)    /* 64*1024*1024 = 64MB (recommended) */
```

- ▶ You can also set the MAXASSIZE parameter dynamically with the SETOMVS command from the z/OS UNIX command line. For example:

```
SETOMVS MAXASSIZE=44040192
```

- ▶ The ASSIZEMAX parameter of the RACF ADDUSER and ALTUSER commands lets you define the maximum region size for a particular user in the OMVS segment of the user's RACF profile. If the ASSIZEMAX parameter is not specified, the MAXASSIZE value in BPXPRMxx is used. If you specify the ASSIZEMAX parameter for the userid that starts Infoprint Server, specify at least 44040192.
- ▶ From the TSO/E OMVS command line
If you issue the **aopstart** command from TSO, the SIZE parameter on the TSO/E LOGON panel determines the maximum region size for an address space. Set the SIZE parameter to 42MB or larger.
- ▶ Using the AOPSTART procedure: If you use the AOPSTART procedure to issue the command, the REGION parameter on the EXEC statement in the procedure determines the maximum region size. Specify REGION=42M or *higher*. If the REGION parameter is not specified, the default region size defined for your installation is used.

Note: The region size must be at least 10M larger than the number of bytes specified in the -M option in the transform configuration file; the default for the -M option is 32M.

7.7.2 Edit the transform configuration file (aopxfd.conf)

The transform configuration file contains information that the Transform Manager uses to manage the PostScript to AFP and PDF to AFP transform and also other transforms. See “Create the transform configuration file (aopxfd.conf)” on page 178 for more information about this file.

The **aopxfd.conf** configuration file must exist before you can start the Transform Manager successfully. If you change the configuration file after the Transform Manager has started, stop and restart the Transform Manager to pick up the changes, as shown in “Restart of Transform Manager” on page 179.

Specify transform classes

You can define transform classes in the configuration file if you want to transform data using different sets of transform resource files. For each transform class, you must create a separate entry in the transform configuration file. In each entry, you can specify a different resource directory in the AOP_RESOURCE_PATH environment variable.

7.7.3 Format of a PostScript and PDF to AFP transform entry

In the **aopxfd.conf** file, specify one transform entry for the PostScript and PDF to AFP transform. If you want to use transform classes, specify a separate transform entry for each transform class:

```
transform ps2afp[_transformclass]
    start-command = "ps2apfd -i initializationfile [-M nnnnn{K|M}]"
    [ environment = {name -> value[ name -> value]... } ]
    [ maximum-idle-time = seconds ]
    [ minimum-active = number ]
    [ maximum-active = number ]
;
```

Figure 7-9 Transform for PS and PDF to AFP

See “Format of a transform entry” on page 184 for the details of the transform entries.

Specific PS2AFP transform entries

For a description of all the transform entries, see “Format of a transform entry” on page 184.

transformclass Specify a transform class if you want to transform PostScript or PDF data using different transform resource files.

Example: transform ps2afp_no_font_substitution

start-command *-i initializationfile* - The initialization file name. This initialization file should *not* be customized. To use the default installation file provided with Infoprint Server Transforms, specify *ps2afpe.ps.bin*.

The *-i* option is required.

Default: None

-M nnnnn{K|M} -- The number of bytes of memory the transform daemon uses to perform transforms. Specify the number of bytes in either kilobytes (K) or megabytes (M). Valid values are:

15M to 256M

15360K to 262144K

The amount of memory required depends on the complexity of the documents to be transformed. If you specify too low a value, the transform of complex PostScript or PDF documents might fail with undefined PostScript errors. If this occurs, try increasing the value in this option.

Also, specify a maximum region size that is at least 10M greater than the value you specify in this option. For example, if you specify *-M 32M*, set the region size to 42M (32M + 10M). If the region size is too low, the transform will fail immediately due to lack of memory. See “Storage requirements” on page 189 for information about how to specify the region size.

Default: 32M

Example: start-command = "ps2afpd -i ps2afpe.ps.bin -M 32M"

environment Environment variables that define the transform environment for the PostScript and PDF to AFP transform. The following environment variable is optional:

AOP_RESOURCE_PATH -- The directory that contains PostScript and PDF to AFP transform resources. If the transform resources are in more than one directory, list all directories separated by a colon. The directories are searched in the order that they are listed. You do not need to list the default location for transform files, */usr/lpp/Printsrv/ps2afp*, because the Transform Manager always searches the default directory last.

Default: AOP_RESOURCE_PATH -> /usr/lpp/Printsrv/ps2afp

PS2AFP transform examples

Figure 7-10 shows two sample entries in the **aopxfd.conf** configuration file; they are defined as follows:

- ▶ The first entry is for the PostScript to AFP and the PDF to AFP transform. Because the transform attribute does not specify a transform class, the Transform Manager uses this entry when no transform class is specified when the transform is invoked.
- ▶ The second entry is also for the PostScript to AFP and the PDF to AFP transform. The Transform Manager uses this entry when transform class **no_font_substitution** is specified when the transform is invoked.

This entry sets environment variable **AOP_RESOURCE_PATH** to a directory that contains transform resource files that have been customized. Transform resource files that are not customized can reside in the default directory, *usr/lpp/Printsrv/ps2afp*.

When the Transform Manager starts, it starts two types of transform daemons, one for PostScript and PDF to AFP transforms that does not specify any transform class, and another for PostScript and PDF to AFP transforms that specify transform class **no_font_substitution**. The Transform Manager keeps no more than two transform daemons of each type active at any time. It shuts down any transform daemon that is idle and frees system resources after 5 minutes; however, the Transform Manager keeps at least one transform daemon of each type active.

```
transform ps2afp
  start-command = "ps2afpd -i ps2afpe.ps.bin"
  minimum-active = 1
  maximum-active = 2
  maximum-idle-time = 300    # 5 minutes
;

transform ps2afp_no_font_substitution
  start-command = "ps2afpd -i ps2afpe.ps.bin"
  environment = {
    AOP_RESOURCE_PATH -> /etc/Printsrv/ps2afp
  }
  minimum-active = 1
  maximum-active = 2
  maximum-idle-time = 300    # 5 minutes
;
```

Figure 7-10 Example of PS2AFP transforms in AOPXFD.CONF

7.7.4 Customize PS2AFP transform resource files

Infoprint Server Transforms provides default transform resource files in the */usr/lpp/Printsrv/ps2afp* directory. You can customize the following transform resource files for your installation:

- ▶ UserInit
- ▶ preload.ps

To customize transform resource files, do the following:

1. Create a new directory for the transform resource files.

IBM recommends that you do *not* change files that reside in the `/usr/lpp/Printsrv` directory. Therefore, create a new directory, for example `/etc/Printsrv/ps2afp`.

2. Copy the transform resource files that you want to customize to the new directory and edit the files.

All files must be in ASCII representation. Therefore, create and edit the resource files on an ASCII platform (such as a Windows system). If you create or edit the file on the z/OS system, you can use the following sample shell scripts to convert your resource files between EBCDIC and ASCII code pages:

To convert ASCII to EBCDIC:

```
#$/bin/sh
cat $1 | iconv -f iso8859-1 -t ibm-1047
```

To convert EBCDIC to ASCII:

```
#$/bin/sh
cat $1 | iconv -f ibm-1047 -t iso8859-1
```

Follow PostScript rules for including comments.

3. Name the new directory that contains the customized files in environment variable `AOP_RESOURCE_PATH` in file **aopxfd.conf**. For example, set `AOP_RESOURCE_PATH` to `/etc/Printsrv/ps2afp`.

UserInit resource file

The UserInit file can contain any valid PostScript commands. PostScript commands in this file are processed during transform initialization. Refer to the PostScript language documentation provided by Adobe for information about PostScript commands. The first two characters in the file must be `%!`.

For example, you can specify a PostScript command to control whether the printer will substitute fonts for any fonts defined in the job that are neither inline with the job nor resident in the printer. Specify one of the following PostScript commands to turn font substitution on and off:

- ▶ To ensure that the printer substitutes Courier fonts, specify:

```
turnFontSubstitutionOn
```

- ▶ To ensure that the printer not print fonts that are not inline or resident in the printer, specify:

```
turnFontSubstitutionOff
```

The default UserInit file provided by Infoprint Server Transforms shown in Figure 7-11 turns font substitution on.

```
%!PS-Adobe-3.0 Resource-ProcSet
turnFontSubstitutionOn
```

Figure 7-11 Default UserInit resource file

preload.ps resource file

If certain fonts occur frequently in your PostScript or PDF data streams, consider listing these font names in the **preload.ps** file. The transform loads the fonts listed in this file into memory during initialization, thereby improving performance of the transform. The default **preload.ps** file provided by Infoprint Server Transforms is shown in Figure 7-12 on page 194.

```

% font    font    font    font    font    font    font    font    font
% preload preload preload preload preload preload preload preload preload

% Comment out any fonts that you do not want preloaded by inserting
% a % in the front of a line.

% Add any fonts that you do want preloaded by inserting
% lines of the format:
% /font_name findfont pop

/Courier          findfont pop
/Courier-Oblique  findfont pop
/Courier-Bold     findfont pop
/Courier-BoldOblique findfont pop

/Helvetica        findfont pop
/Helvetica-Oblique findfont pop
/Helvetica-Bold   findfont pop
/Helvetica-BoldOblique findfont pop

/Times-Roman      findfont pop
/Times-Italic     findfont pop
/Times-Bold       findfont pop
/Times-BoldItalic findfont pop

/Symbol           findfont pop

/AvantGarde-Book   findfont pop
/AvantGarde-BookOblique findfont pop
/AvantGarde-Demi   findfont pop
/AvantGarde-DemiOblique findfont pop

/Bookman-Demi      findfont pop
/Bookman-DemiItalic findfont pop
/Bookman-Light     findfont pop
/Bookman-LightItalic findfont pop

/Helvetica-Narrow   findfont pop
/Helvetica-Narrow-Oblique findfont pop
/Helvetica-Narrow-Bold findfont pop
/Helvetica-Narrow-BoldOblique findfont pop

/NewCenturySchlbk-Roman findfont pop
/NewCenturySchlbk-Italic findfont pop
/NewCenturySchlbk-Bold findfont pop
/NewCenturySchlbk-BoldItalic findfont pop

/Palatino-Roman    findfont pop
/Palatino-Italic   findfont pop
/Palatino-Bold     findfont pop
/Palatino-BoldItalic findfont pop

/ZapfChancery-MediumItalic findfont pop
/ZapfDingbats      findfont pop

```

Figure 7-12 Default *preload.ps* resource file

7.8 Customize the SAP to AFP Transform

If you want Print Interface to automatically call the SAP to AFP transform before writing data to the JES spool, the administrator must specify the transform DLL in the Filter field of printer definitions in the Printer Inventory. In the printer definition, the administrator can also specify transform options, such as the resolution of the image data created and the transform class to use. The user can specify the same transform options on the **sap2afp** command and in the filter-options job attribute when submitting a print request.

For a complete description on customizing the SAP to AFP transform, see *z/OS Infoprint Server Customization*, S544-5744.

7.9 Customize the AFP to PCL, PDF, and PostScript transforms

If you want Print Interface to automatically call a transform before writing data to the JES spool, the administrator must specify the name of the transform DLL in the Filter field of printer definitions in the Printer Inventory. In the printer definition, the administrator can also specify transform options, such as the transform class to use. The user can specify the same transform options on the **afp2pcl**, **afp2pdf**, and **afp2ps** commands and in the filter-options job attribute when submitting a print request.

7.9.1 Edit the transform configuration file (aopxfd.conf)

The transform configuration file contains information that the Infoprint Server Transform Manager uses to manage the AFP transforms and also other transforms. See “Create the transform configuration file (aopxfd.conf)” on page 178” for more information about this file.

This configuration file must exist before you can start the Transform Manager successfully. If you change the configuration file while the Transform Manager is started, stop and restart the Transform Manager to pick up the changes.

In the **aopxfd.conf** file, create a separate transform entry for each of the following transforms that you plan to use:

- ▶ AFP to PCL transform
- ▶ AFP to PDF transform
- ▶ AFP to PostScript transform

The general format of an AFP to PCL, PostScript, or PDF transform entry is shown in Figure 7-13.

```
transform afp2ccc[_transformclass]
    start-command = afp2cccd
    [ environment = {name -> value [ name -> value]... } ]
    [ maximum-idle-time = seconds ]
    [ minimum-active = number ]
    [ maximum-active = number ]
```

Figure 7-13 Transform entry format for AFP to PCL, PostScript, and PDF

Where:

afp2ccc is the name of the transform provided by Infoprint Server Transforms. Valid values are:

- afp2pcl** The AFP to PCL transform
- afp2pdf** The AFP to PDF transform
- afp2ps** The AFP to PostScript transform

start-command

The transform daemons are started by the specification in the **aopxfd.conf** file. Specify the name of the transform daemon. If the transform daemon is not in a directory identified in the PATH environment variable, specify the full directory path name of the daemon.

Valid values are:

- afp2pcld** The AFP to PCL transform daemon
- afp2pdfd** The AFP to PDF transform daemon
- afp2psd** The AFP to PostScript transform daemon

transformclass

The name of an optional transform class, for example **transform afp2pcl_us**

Note: For a z/OS UNIX user to use this transform class on the **afp2pcl** command, enter **afp2pcl -c us**

environment

The following environment variables are optional:

AOP_CHARS

The default coded font. You might want to specify the same font as the hardware font in an AFP printer. The transforms use this font to format error messages unless the page definition specified in the AOP_MSGPAGEDEF variable names a font. The transforms also use this font for (1) line data when no other font is specified in the page definition used to print the document and (2) AFP data when no other font is specified in the AFP data stream.

Specify only one default font. Specify the 1- to 4-character coded font name. You can specify the X0 or XZ prefix of the coded font name; if you do not specify a prefix, the transform adds an X0 prefix. Some coded fonts have 6-character names, not counting the X0 or XZ prefix. For these fonts, use the 4-character alternate coded font name. For font names and alternate font names, refer to IBM AFP Fonts: Font Summary for AFP Font Collection.

If the default font is a raster font (indicated by the X0 prefix), the AFP to PDF and AFP to PostScript transforms map it to an outline font if AOP_FONTMAP -> yes. The coded font member for the raster font, for example X060D9, must exist in an AFP font library specified to the transform, even if the transform maps raster-to-outline fonts.

The code page associated with this coded font must be an EBCDIC code page.

Default: AOP_CHARS -> 60d9. The transforms prefix X0 to this font.

Example: environment = {AOP_CHARS -> 60d8}

AOP_COLOR

An indication of whether the transform is to produce color output. Specify yes if the printer supports color or if you want to view PDF output in color. Valid values are:

yes -The transform produces color output.

no - The transform performs color simulation.

For the AFP to PDF transform, specify *AOP_COLOR* -> *yes* to view the output in color with Adobe Acrobat.

Default: AOP_COLOR -> no

Example: environment = {AOP_COLOR -> yes}

AOP_FONTLIB

AFP system resource libraries that contain fonts. Specify one to eight data set names; separate each name with a space. Data sets are searched in the order listed.

Specify the AFP 300-pel raster and outline font libraries used by your installation. The font libraries you need to specify depend on the type of transform and how you set the AOP_FONTMAP variable, as follows:

AFP to PCL transform -- Specify AFP 300-pel raster font libraries because the AFP to PCL transform requires raster fonts.

You do not need to specify AFP outline font libraries.

AFP to PDF transform -- Specify AFP 300-pel raster font libraries if either you specify a raster font in the AOP_CHARS environment variable or documents to be transformed reference raster fonts and you specify AOP_FONTMAP-> NO to disable mapping of raster-to-outline fonts.

Specify AFP outline font libraries if either data to be transformed references outline fonts or you specify AOP_FONTMAP -> YES to map raster-to-outline fonts.

AFP to PostScript transform -- Specify AFP 300-pel raster font libraries if either you specify a raster font in the AOP_CHARS environment variable or documents to be transformed reference raster fonts and you specify AOP_FONTMAP-> NO to disable mapping of raster-to-outline fonts.

Specify AFP outline font libraries if either data to be transformed references outline fonts or you specify AOP_FONTMAP -> YES to map raster-to-outline fonts.

Default: AOP_FONTLIB -> "sys1.font300"

Example: environment = {AOP_FONTLIB -> "sys1.font300
inst.font300"}

AOP_FONTMAP

An indication of whether or not the transform maps fonts. Font-mapping is performed using a font-mapping table provided with the transform. Each transform maps between outline and raster fonts; however, each transform maps fonts in only one direction, as follows:

AFP to PCL transform: This transform maps outline fonts to AFP raster fonts. Because the AFP to PCL transform requires raster fonts, *always* specify AOP_FONTMAP -> *yes* (default).

AFP to PDF transform: This transform maps AFP raster fonts to outline fonts. The AFP to PDF transform can use either raster or outline fonts; however, outline fonts provide higher quality output for viewing and printing.

AFP to PostScript transform: This transform maps AFP raster fonts to outline fonts. The AFP to PostScript transform can use either raster or outline fonts; however, outline fonts provide higher quality output for printing.

Valid values are:

yes -- The transform maps fonts.

no -- The transform does not map fonts.

Default: AOP_FONTMAP -> yes

Example: environment = {AOP_FONTMAP -> no}

AOP_FORMDEF

Default form definition used to format the input data stream and create PCL, PDF, or PostScript output. Specify the one to eight character form definition name, with or without the f1 prefix. If you omit the f1 prefix, the transform adds it.

The transform uses this form definition only if no other form definition is specified.

Default: AOP_FORMDEF -> f1cp0110

Example: environment = {AOP_FORMDEF -> f1cp0111}

AOP_FORMDEFLIB

AFP system resource libraries that contain form definitions. Specify from one to eight data set names; separate each name with a space. Data sets are searched in the order listed.

Default: AOP_FORMDEFLIB -> "sys1.fdeflib"

Example: environment = {AOP_FORMDEFLIB -> "sys1.fdeflib
inst.fdeflib"}

AOP_MSGFORMDEF

Form definition used to format transform error messages. Specify the one to eight character form definition name, with or without the f1 prefix. If you omit the f1 prefix, the transform adds it. This form definition must be located in one of the libraries specified in the AOP_FORMDEFLIB variable.

Default: AOP_MSGFORMDEF -> f1cp0110

Example: environment = {AOP_MSGFORMDEF -> f1cp0111}

AOP_MSGPAGEDEF

Page definition used to format transform error messages. Specify the one to eight character page definition name, with or without the p1 prefix. If you omit the p1 prefix, the transform adds it. This page definition must be located in one of the libraries specified in the AOP_PAGEDFLIB variable. The transform formats messages for the first paper size defined in the AOP_PAPER variable. IBM recommends page definition P1P08682 for letter size paper, and page definition P1Q09182 for A4 paper.

Default: AOP_MSGPAGEDEF -> p1p08682

Example: environment = {AOP_MSGPAGEDEF -> p1p06362}

AOP_OUTLINES

Type of processing the AFP to PDF transform performs for outline fonts. This variable applies only to the AFP to PDF transform; the AFP to PCL and AFP to PostScript transforms ignore this environment variable. Valid values are:

builtin -- The AFP to PDF transform includes only the names of outline fonts in the PDF output. Outline fonts themselves are not included in the output. When the PDF output is viewed or printed, the

PDF driver (for example, Adobe Acrobat or a PDF printer) maps the requested typeface to an equivalent typeface. This option produces smaller output files than when fonts are included in the output; however, differences in output appearance can result due to the mapping performed by the PDF driver. For example, font spacing might be different, and if you use special characters or if you use characters or glyphs that are unknown to the PDF driver, these characters will not be visible.

yes -- The AFP to PDF transform includes outline fonts within the PDF output. This option provides better output fidelity than the builtin option; however, each typeface increases the size of the PDF output file by approximately 110K.

Default: AOP_OUTLINES -> yes

Example: environment = {AOP_OUTLINES -> builtin}

AOP_OVERLAYLIB AFP system resource libraries that contain overlays. Specify from one to eight data set names; separate each name with a space. Data sets are searched in the order listed.

Default: AOP_OVERLAYLIB -> "sys1.overlib"

Example: environment = {AOP_OVERLAYLIB -> "sys1.overlib
inst.overlib"}

AOP_PAGEDDEF Default page definition used to format line data and create PCL, PDF, or PostScript output. Specify the one to eight character page definition name, with or without the p1 prefix. If you omit the p1 prefix, the transform adds it. The transform uses this page definition only if no other page definition is specified. IBM recommends page definition P1P08682 for letter size paper, and page definition P1Q09182 for A4 paper.

Default: AOP_PAGEDDEF -> p1p08682

Example: environment = {AOP_PAGEDDEF -> p1q09182}

AOP_PAGEDFLIB AFP system resource libraries that contain page definitions. Specify from one to eight data set names; separate each name with a space. Data sets are searched in the order listed.

Default: AOP_PAGEDFLIB -> "sys1.pdeflib"

Example: environment = {AOP_PAGEDFLIB -> "sys1.pdeflib
inst.pdeflib"}

AOP_PAGESEGLIB AFP system resource libraries that contain page segments. Specify from one to eight data set names; separate each name with a space.

Default: AOP_PAGESEGLIB -> "sys1.pseglib"

Example: environment = {AOP_PAGESEGLIB -> "sys1.pseglib
inst.pseglib"}

AOP_PAPER Paper names that identify the size of paper for each AFP input tray number. You can specify from one to ten paper names; separate each name with a space. Default values are used if you specify less than ten paper sizes.

Default: AOP_PAPER -> "letter letter letter letter letter letter letter letter letter letter"

Example: environment = {AOP_PAPER -> "letter legal letteree letter letter letter letter letter letter"}

AOP_TRAYID

The input-tray number of the target printer that maps to AFP input-tray numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, and any number greater than 9. Specify from one to ten tray numbers; separate each number with a space. Default values are used if you specify less than ten tray numbers.

Number 0 (zero) indicates that an input tray is not installed in the target printer; if a job submitter selects an input tray that is not installed, the transform issues an error message in the output file and uses printer tray 1. To avoid an error message, specify tray 1 instead of tray 0 for input trays that are not installed.

If a tray is installed, the value you specify depends on the type of transform:

AFP to PCL transform: Specify the ID used by the PCL printer to select each tray. Tray IDs are usually specified by the printer manufacturer.

AFP to PDF transform: Specify 1 if you want the AFP tray to be available for selection.

AFP to PostScript transform: Specify the ID used by the PostScript printer to select each tray. This value, minus 1, corresponds to an entry in the Priority array in the InputAttributes dictionary for the PostScript printer. Printer-specific values are described in the PostScript PPD file for the printer.

Defaults: The default values for each transform are:

AFP to PCL: AOP_TRAYID -> "1 4 0 0 0 0 0 0 2"

AFP to PDF: AOP_TRAYID -> "1 1 1 1 1 1 1 1 1 1"

AFP to PostScript: AOP_TRAYID -> "1 2 0 0 0 0 0 0 2"

Example: environment = {AOP_TRAYID -> "1 4 1 1 1 1 1 1 1 2"}

7.9.2 Transform classes

If you also want to define transform classes, create a separate transform entry for each transform class. You can define transform classes in the configuration file if you want to transform data for printers with different:

- ▶ Color capabilities
- ▶ Input tray numbers
- ▶ Paper sizes
- ▶ Default AFP resources
- ▶ AFP resource libraries

These definitions are accomplished by using environment variables.

Color capabilities

For example, you can define one transform class for color printers, and another for monochrome printers by using the environment variable **AOP_COLOR**. For each transform class, you must create a separate entry in the transform configuration file. In each entry, you can specify different options such as color capability in environment variables.

The sample transform configuration file */usr/lpp/Printsrv/samples/aopxfg.conf* includes definitions for different transform classes. A typical transform is shown in Figure 7-14.

```

#-----
# AFP to PCL transform with typical United States paper sizes
#
# NOTE: An outline font library is needed for font mapping. If font mapping
#       is wanted, add your outline font library (e.g. SYS1.FONTOLN) to the
#       AOP_FONTLIB concatenation. If font mapping is not wanted, AOP_FONTMAP
#       should be set to "no".
#-----
transform afp2pcl_us
  start-command = afp2pcl
  min-active = 0
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_CHARS -> 60D9
    AOP_COLOR -> no
    AOP_FONTLIB -> "sys1.font300"
    AOP_FONTMAP -> yes
    AOP_FORMDEF -> F1CP0110
    AOP_FORMDEFLIB -> "sys1.fdeflib"
    AOP_MSGFORMDEF -> F1CP0110
    AOP_MSGPAGEDEF -> P1P08682
    AOP_OVERLAYLIB -> "sys1.overlib"
    AOP_PAGEDEF -> P1P08682
    AOP_PAGEDEFLIB -> "sys1.pdeflib"
    AOP_PAGESEGLIB -> "sys1.pseglib"
    AOP_PAPER -> "letter legal letter letter letter letter letter letter letter"
    AOP_TRAYID -> "1 4 0 0 0 0 0 0 2"
    _BPX_JOBNAME -> AFP2PCLD
  };

```

Figure 7-14 Sample AFP to PCL transform class

Specify paper trays and paper sizes

In the transform entries, you can specify which paper trays are installed in your printers and you can control how the transform maps AFP tray IDs to those trays. You can also specify the size of paper installed in each paper tray. You specify this information in two environment variables:

AOP_TRAYID This variable indicates which paper trays are installed in your printer and specifies a printer tray ID for those trays. You can specify from one to ten positional tray IDs. The position (1 to 10) of the value indicates the AFP tray ID that corresponds to the printer tray ID; the tenth position represents any AFP tray ID greater than 9. For example, assume that you specify these printer tray IDs in AOP_TRAYID:

```
AOP_TRAYID -> "1 4 1 1 1 1 1 1 1 2"
```

If you specify AFP tray ID 2, the transform will use tray ID 4. If the user specifies AFP tray ID M, then the transform uses tray ID 2.

Value 0 indicates that the paper tray is not installed in the printer.

If a tray is installed, the value you specify depends on the type of transform:

AFP to PCL transform: Tray IDs are usually specified by the printer manufacturer.

AFP to PDF transform: Specify 1 if you want the AFP tray number to be available for selection. Specify 0 if you do not want it to be available.

AFP to PostScript transform: Specify the ID used by the PostScript printer to select each tray. This value, minus 1, corresponds to an entry in the Priority array in the InputAttributes dictionary for the PostScript printer.

Printer-specific values are described in the PostScript PPD file for the printer.

AOP_PAPER This variable specifies the paper that is installed in each input paper tray. You can specify a fixed set of paper names. Table 7-2 on page 203 summarizes the valid paper names and their dimensions.

You can specify from one to ten positional paper names. The position (1 through 10) of the paper name indicates the AFP tray ID that corresponds to the paper name; the tenth position is used for any AFP tray ID greater than 9. For example, assume that you specify these paper names in AOP_PAPER:

AOP_PAPER -> "letter legal letteree"

If you select AFP input tray 2, then the transform formats the output for legal size paper. If you select input tray 3, then the transform formats the output for letteree size paper. If you select any AFP input tray greater than 3, the transform formats the output for the default paper size, which is letter.

Note: The AFP to PDF transform formats all output for the same paper size, using the paper size for the first AFP input tray ID selected.

You can specify an AFP tray ID in one of the following:

- ▶ The input-tray-number job attribute
- ▶ The INTRAY JCL parameter
- ▶ The AFP form definition.

You can also specify an AFP tray name in the input-tray job attribute; Print Interface maps this tray name to a tray number using information specified in the printer definition.

All paper size values are expressed in 300 dots per inch (dpi) in both the X (width) and Y (height) directions; the origin of the X and Y coordinates is the top left corner of the page.

The columns have these meanings:

Paper Name	The name you specify in the AOP_PAPER environment variable.
Paper Size	The size in 300 dots per inch (dpi) of the entire media. The AFP to PDF and the AFP to Postscript transforms use this value as the logical page to determine the area of the page that can be toned or printed.
Printable Area Origin	The X and Y coordinates of the top left corner of the printable area in relation to the physical media.
Printable Area Size	The area of the page that can be toned or printed.
Logical Page Origin	The X and Y coordinates of the top left corner of the logical page in relation to the physical media.
Logical Page Size	The area of the page that can be addressed (positioned by the printer). The AFP to PCL transform uses this value.
PCL Paper Type ID	The paper type identifier assigned by PCL and used to select this paper name.

Table 7-2 summarizes the names and dimensions of the paper that the transforms support.

Table 7-2 Paper names and dimensions

Paper name	Paper size	Printable area origin	Printable area size	Logical page origin	Logical paper size (PCL only)	PCL paper type ID
a3	3507,4960	50,50	3407,4860	71,0	3365,4818	27
a4	2480,3507	50,50	2480,3507	71,0	2338,3507	26
a4ee	2480,3507	0,0	2480,3507	0,0	2480,3507	26
c5	1913,2704	50,50	1813,2604	71,0	1771,2562	91
com10	1237,2850	50,50	1137,2750	75,0	1087,2700	81
dl	1299,2598	50,50	1199,2498	71,0	1157,2456	90
executive	2175,3150	50,50	2075,3050	75,0	2025,3000	1
legal	2550,4200	50,50	2450,4100	75,0	2400,4050	3
letter	2550,3300	50,50	2450,3200	75,0	2400,3150	2
letteree	2550,3300	0,0	2550,3300	0,0	2550,3300	2
monarch	1162,2250	50,50	1062,2250	75,0	1012,2100	80

Note: Use the a4ee and letteree paper names when the printer is configured for edge-to-edge printing.

If you specify a logical page that is greater than the paper dimensions, the transforms attempt to print outside of the paper dimensions; that is, the transforms use the user-specified logical-page dimensions. You can specify the size of the logical page in the AFP input and in a page definition (for line data input). When an attempt is made to print outside of the logical page size specified in the paper dimensions, loss of output data and unpredictable formatting results can occur.

Specify default AFP resources

AFP resources are collections of data and control information that the transforms use to create the PCL, PostScript, or PDF output. You can specify default AFP resources (font, form definition, and page definition) in each transform entry. You can also specify the same default AFP resources in the printer definitions for the target printers. Default resources specified in the printer definition override default resources specified in the transform entry.

In most situations, specify the same default AFP resources as you currently specify either in your PSF for OS/390 startup procedures (in the PRINTDEV statements) or in your PSF for OS/390 FSA definitions in the Printer Inventory. In the transform entry, specify the font, form definition, and page definition that apply to most of the target printers. If one or more printers require different defaults, specify those defaults in the printer definitions for the printers. In a printer definition, you can also specify other AFP resources for the transforms to use, such as an overlay for the front or back side of a page.

AFP resource environment variables

You specify default AFP resources in the following environment variables:

AOP_CHARS	This variable identifies the default font used for transform error messages and for line data and AFP data that does not specify another font. You can specify only one font in this variable. The default font can be either a raster or outline font.
AOP_FORMDEF	This variable identifies the default form definition used when no other form definition is specified.
AOP_PAGEDEF	This variable identifies the default page definition used when no other page definition is specified.

Locations to specify AFP resources

AFP resources can also be specified in the following locations:

- ▶ Job attributes and JCL parameters
- ▶ Input data set
- ▶ JES initialization statements for the IP PrintWay FSA (fonts and page definitions only)
- ▶ Printer definitions in the Printer Inventory

Search hierarchies for form definition

The transforms use search hierarchies to select resources when they are specified in more than one location.

The transforms use the following hierarchy to select the name of the *form definition*:

1. The form definition specified in one of the following:
 - form-definition job attribute
 - FORMDEF JCL parameter
2. The form definition in the Form definition field in the printer definition.
3. The first inline form definition.
4. The form definition in the AOP_FORMDEF environment variable.
5. Form definition F1CP0111. This default is coded in the transform.

Note: If the form definition name dummy is specified, the transform uses the first inline form definition. After the transform determines the name of the form definition, the transform searches for the form definition first inline in the data set, and then in the user and system resource libraries.

Search hierarchies for page definition

The transforms use the following hierarchy to select the name of the *page definition*:

1. The page definition specified in one of the following:
 - page-definition job attribute.
 - PAGEDEF or FCB JCL parameter; the value in the PAGEDEF parameter overrides the FCB parameter.
2. The default page definition supplied by JES. (See Note.)
3. The Page definition field in the printer definition.
4. The first inline page definition.
5. The AOP_PAGEDEF environment variable in the transform configuration file.

6. Page definition P1P08682; this default is coded in the transform.

Note: If page definition name dummy is specified, the transform uses the first inline page definition. After the transform determines the name of the page definition, the transform searches for the page definition first inline in the data set, and then in the user and system resource libraries.

JES page definitions

JES can supply a default page definition for jobs submitted directly to IP PrintWay and not processed first by NetSpool or Print Interface. To prevent JES from supplying a default page definition to the IP PrintWay FSA, do the following:

- ▶ In JES2, do not specify the FCB parameter in the JES PRTnnnnn statement, and do not specify the NIFCB parameter in the JES PRINTDEF statement.
- ▶ In JES3, specify PDEFAULT=FCB in the JES DEVICE statement.

Search hierarchies for fonts

The transforms use the following hierarchy to select a font for line data and AFP data that does not specify a *font*:

1. The font named in the page definition.
2. The font specified in one of the following locations:
 - chars job attribute
 - CHARS or UCS JCL parameter; the value in the CHARS JCL parameter overrides the UCS parameter.
3. The default font supplied by JES. (See Note.)
4. The font in the Character set field in the printer definition.
5. The font in the AOP_CHARS environment variable; the transform prefixes X0 to the font named in the variable if you do not specify a prefix.
6. Font X060D9; this default is coded in the transform.

JES font definitions

JES can supply a default font for jobs submitted directly to IP PrintWay and not processed first by NetSpool or Print Interface. To prevent JES from supplying a default font to the IP PrintWay FSA, do the following:

- ▶ In JES2, specify UCS=0 in the JES PRTnnnnn statement.
- ▶ In JES3, specify PDEFAULT=CHARS in the JES DEVICE statement.

Transform hierarchies for font selection

The transforms use the following hierarchy to select a *font* used for transform *error messages*:

1. The font named in the page definition specified in the AOP_MSGPAGEDEF environment variable.
2. The font in the AOP_CHARS environment variable; the transform prefixes X0 to the font named in the variable if you do not specify a prefix.
3. Font X060D9; this default is coded in the transform.

Specify AFP resource libraries

In the transform entries, you can specify up to eight AFP system resource libraries for fonts, page definitions, form definitions, page segments, and overlays. You can also specify the AFP resource libraries in the printer definitions for the target printers. Resource libraries specified in the printer definition are searched before libraries specified in the transform entry.

In most situations, specify the same default AFP resource libraries that you currently specify in your PSF for OS/390 startup procedures. In the transform entry, specify the libraries that apply to most of the target printers. If one or more printers require different resource libraries, specify those libraries in the printer definitions for the printers.

Resource library environment variables

You can specify system resource libraries in the following environment variables:

AOP_FONTLIB Identifies the system font libraries.

AOP_FORMDEFLIB Identifies the system form definition libraries.

AOP_OVERLAYLIB Identifies the system overlay libraries.

AOP_PAGEDEFLIB Identifies the system page definition libraries.

AOP_PAGESEGLIB Identifies the system page segment libraries.

AFP resource libraries can also be specified in the following locations:

- Job attributes and JCL parameters
- Printer definitions in the Printer Inventory

AFP resource libraries must be accessible by the user ID that starts Infoprint Server. When you use the AOPSTART JCL procedure (instead of the **aopstart** command) to start Infoprint Server, you can associate a specific user ID with the procedure and in this way ensure that the same user ID always starts Infoprint Server. (This user ID must be a member of the AOOPER group.)

Transform hierarchies for resource libraries

The transforms use the following hierarchy when searching *AFP resource libraries*:

1. Resource libraries specified by the job submitter in one of the following locations:
 - resource-library job attribute
 - USERLIB JCL parameter
2. Resource libraries specified in the Resource library field in the printer definition. These libraries are searched only when the job submitter does not specify any resource libraries (see step 1).
3. Resource libraries specified in environment variables in the transform configuration file (**aopxfd.conf**).
4. Default resource libraries that are hard-coded in the transform. These libraries are searched only if no system resource libraries are specified in the transform configuration file (see step 3).

Mapping AFP raster and AFP outline fonts

These transforms can optionally map single-byte AFP raster fonts to AFP outline fonts and single-byte AFP outline fonts to AFP raster fonts. Font mapping lets you migrate between type technologies while maintaining the same font appearance and document fidelity. By default, font-mapping is enabled; therefore, customization is required only if you want to disable font-mapping or modify the font-mapping table to, for example, add custom fonts used by your installation.

The AFP to PCL, PDF, and PostScript transforms use an internal font-mapping table to map fonts. This table supports the IBM AFP Font Collection Expanded Core Fonts, which are available in both raster and outline formats. It can be customized for your installation.

Transform font mapping

Each transform performs different font mapping, as follows:

- AFP to PCL** This transform can map outline fonts to equivalent 300-pel raster fonts. The AFP to PCL transform can use only 300-pel raster fonts; therefore, to print documents that use outline fonts, you must use the font-mapping function.
- If an input document uses an outline font and the transform cannot find an equivalent raster font in the mapping table, the transform generates an error and terminates printing.
- AFP to PostScript** This transform can map AFP raster fonts to equivalent AFP outline fonts. The AFP to PostScript transform can use either 300-pel raster fonts or outline fonts; however, outline fonts provide better print quality at higher printer resolutions.
- If an input document uses a raster font, and the transform cannot find an equivalent outline font in the mapping table, the transform uses the raster font.
- AFP to PDF** This transform can map AFP raster fonts to equivalent AFP outline fonts. The AFP to PDF transform can use either 300-pel raster fonts or outline fonts; however, outline fonts provide superior viewing quality.
- If an input document uses a raster font, and the transform cannot find an equivalent outline font in the mapping table, the transform uses the raster font.

Font mapping environment variables

In the transform configuration file, you can control whether or not font-mapping is enabled for each transform and how the AFP to PDF transform handles outline fonts. The following environment variables in the configuration file let you control font mapping:

- AOP_FONTMAP** This variable enables and disables font mapping. IBM recommends that you enable font mapping for all three transforms; by default, font mapping is enabled. Consider the following when selecting this value:
- AFP to PCL transform:** IBM recommends that you enable font mapping to map outline fonts to raster fonts because this transform requires raster fonts.
- AFP to PostScript transform:** IBM recommends that you enable font mapping to map raster fonts to outline fonts because outline fonts provide better print quality. For the transform to map raster-to-outline fonts, your installation must have outline font libraries installed.
- If your installation does not have outline font libraries installed, you do not need to disable font mapping for the AFP to PDF or AFP to PostScript transforms. If the transform does not find an outline font in the system font libraries, the transform automatically disables font-mapping for the raster font. The transform issues an error message in the transform error log once for each missing outline font. To avoid this error message, you can either remove the missing font from the font-mapping table or disable font mapping in this variable.
- AFP to PDF transform:** IBM recommends that you enable font mapping to map raster fonts to outline fonts because outline fonts

provide superior viewing quality. For the transform to map raster-to-outline fonts, your installation must have outline font libraries installed.

AOP_OUTLINES This variable controls, (*for AFP to PDF transform only*), whether or not the AFP to PDF transform includes outline fonts in the PDF output. The *yes* option (default), causes the transform to include outline fonts in the output data. The *builtin* option causes the transform to include only the names of outline fonts in the PDF output. Select the *builtin* if you want to reduce the size of your PDF files.

7.9.3 AFP to PCL transform examples

The examples in Figure 7-15 on page 209 show two sample entries in the **aopxfg.conf** configuration file. Both entries are for the AFP to PCL transform:

- The first entry can be used for printers that print on letter and legal size paper. Because the transform attribute does not specify a transform class, the Transform Manager uses this entry when no transform class is specified. For example, this entry is used when either of the following commands is entered:

```
afp2pcl -o mypclfile.out myfile.afp
lp -d myprinter myfile.afp
```

This entry sets environment variables suitable for a printer that has letter size paper in tray 1 and legal size paper in tray 2. The paper size defined for tray 3 lets users format output for edge-to-edge printing on letter size paper; to do this, users select input tray 3 in the input-tray-number job attribute or in the form definition.

- The second entry can be used for printers that print on A4, A3 and C5 paper. The Transform Manager uses this entry when transform class *eu* is specified. For example, this entry is used when either of the following commands is entered:

```
afp2pcl -c eu -o mypclfile.out myfile.afp
lp -o "filter-options='-c eu'" -d myprinter myfile.afp
```

This entry sets environment variables suitable for a printer that has A4 paper in tray 1, A3 paper in tray 2, and A4 paper in any tray with an AFP tray ID greater than 9. The paper size defined for tray 3 lets users format output for edge-to-edge printing on A4 paper; to do this, users select input tray 3 in the input-tray-number job attribute or in the form definition.

When the Transform Manager starts, it starts two types of transform daemons: One to transform jobs that do not specify a transform class, and another to transform jobs that specify transform class *eu*. The Transform Manager keeps no more than two of each type of transform daemon active at any time. It shuts down any transform daemon that is idle and frees system resources after 5 minutes. It keeps at least one transform daemon of the first type active.

```

transform afp2pcl
  start-command = afp2pcl
  min-active = 1
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_FONTLIB -> "sys1.font300"
    AOP_PAGDEFLIB -> "sys1.pdeflib inst.pdeflib"
    AOP_FORMDEFLIB -> "sys1.fdeflib inst.fdeflib"
    AOP_PAGESEGLIB -> "sys1.pseglib inst.pseglib"
    AOP_OVERLAYLIB -> "sys1.overlib inst.overlib"
    AOP_TRAYID -> "1 4 1 1 1 1 1 1 2"
    AOP_PAPER -> "letter legal letteree letter letter letter letter letter letter letter"
    AOP_COLOR -> no
    AOP_MSGPAGEDEF -> p1p08682
    AOP_MSGFORMDEF -> f1cp0110
    AOP_PAGEDEF -> p1p08682
    AOP_FORMDEF -> f1cp0110
    AOP_CHARS -> 60d9
    AOP_FONTMAP -> yes
  }
;

transform afp2pcl_eu
  start-command = afp2pcl
  min-active = 0 max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_FONTLIB -> "sys1.font300"
    AOP_PAGDEFLIB -> "sys1.pdeflib inst.pdeflib"
    AOP_FORMDEFLIB -> "sys1.fdeflib inst.fdeflib"
    AOP_PAGESEGLIB -> "sys1.pseglib inst.pseglib"
    AOP_OVERLAYLIB -> "sys1.overlib inst.overlib"
    AOP_TRAYID -> "1 4 1 1 1 1 1 1 2"
    AOP_PAPER -> "a4 a3 a4ee a4 a4 a4 a4 a4 c5"
    AOP_COLOR -> no
    AOP_MSGPAGEDEF -> p1q09182
    AOP_MSGFORMDEF -> f1cp0110
    AOP_PAGEDEF -> p1q09182
    AOP_FORMDEF -> f1cp0110
    AOP_CHARS -> 60d9
    AOP_FONTMAP -> yes
  }

```

Figure 7-15 AFP to PCL transform examples

7.9.4 Customizing the font-mapping table

All three transforms use the same default internal table to perform the font-mapping function. This font-mapping table supports all single-byte character sets included in the IBM AFP Font Collection V2 (program number 5648-B33), which are available in both raster and outline formats. The sample font-mapping table provided in SYS1.SAMPLIB(AOXFONTS) is the same as the default internal font-mapping table.

The default font-mapping table is suitable for most installations. You might want to modify the table for these conditions:

- ▶ You do not have AFP outline fonts for some character sets. In this case, you can delete the entry for the font from the table used by the AFP to PDF and AFP to PostScript transforms. However, it is not necessary to delete the font entry because the transforms use the raster font when an outline font in the table cannot be found in the font libraries. If an outline font cannot be found, a warning message (AOX1109W CZxxxx FONTLIB MEMBER NOT FOUND) is written to the transform stderr log; you can ignore this message.
- ▶ You have custom fonts in both raster and outline format, and you want the transform to substitute one font for another. In this case, add an entry in the table for the custom font.

Font-mapping table modifications

To modify the font-mapping table, do the following:

1. Edit the font-mapping entries in the default font-mapping table provided in SYS1.SAMPLIB(AOXFONTS). Follow these instructions when you add or delete an entry:
 - To add a font entry, specify positions 3 through 6 of the outline and raster character-set name. For example, the following table shows the character set names for the Latin1 Gothic Text font and the value to specify in the font-mapping table.

Table 7-3 Latin1 Gothic Text font character set names and font-mapping table value

Character Set Names	Outline Font	Table
C06200xx	CZ6200	C'6200'

The transforms use only a portion of the font character-set name to perform font mapping (that is, the type family, typeface, code page category, and complement), which are positions 3 through 6 of the character set name. You do not need to include a separate entry for each point size. The transforms select the appropriate point size based on the input font.

- Ensure that all entries in the table are in ascending order, with the smallest hexadecimal EBCDIC values first. For example, you would specify the following fonts in the order shown:

C'420P'
C'4200'

The hexadecimal EBCDIC value of C'420P' (X'F4F2F0D7') is smaller than C'4200' (X'F4F2F0F0').

- Do not change the CSECT name (PSSFONTS) of the font table.

See “Font-mapping restrictions” on page 211 for more information. Figure 7-16 on page 211 shows a portion of the sample font-mapping table.

2. Replace the font-mapping table in each transform to which the table applies. In most situations, the same font-mapping table would apply to each transform. The following sample usermods are provided to replace the font mapping table:

Table 7-4 Sample usermods

Transform	Sample Usermod
AFP to PCL	SYS1.SAMPLIB(AOX2PCLJ)
AFP to PDF	SYS1.SAMPLIB(AOX2PDFJ)
AFP to PostScript	SYS1.SAMPLIB(AOX2PSJ)

Refer to the comments in the sample job for instructions on installing the usermods.

3. Stop the Transform Manager if it is started. For example, use this z/OS UNIX command:

```
aopstop -d xfd
```

4. Restart the Transform Manager. For example, use this z/OS UNIX command:

aopstart

Note: If your installation modifies the font-mapping table, unexpected output results can occur if incorrect font mappings are performed, that is, when the font metrics of the source input font do not reflect the target font metrics.

Font-mapping restrictions

The following restrictions apply to the font-mapping function:

► Supported Fonts

The font-mapping table is tailored for the IBM AFP Font Collection Expanded Core Fonts. Customization of the font mapping table is required if additional font mappings are required. Font mappings can only be performed if positions 3 to 6 of the character set name for the outline and raster font are identical.

► Asymmetric fonts (for outline to raster mapping)

The font-mapping table does not support mapping of asymmetric AFP outline fonts. Asymmetric AFP outline fonts are fonts that are anamorphically scaled in the MCF (Map Coded Font) structured field.

► Point size restrictions (for outline to raster mapping)

When a request to map an AFP outline font results in the selection of a non-decimal point size, a font map to the equivalent raster is performed within one point size. If the corresponding raster font cannot be selected, the document is not transformed.

Sample font-mapping table - SYS1.SAMPLIB(AOXFONTS)

Figure 7-16 shows a portion of the default font-mapping table provided in SYS1.SAMPLIB(AOXFONTS). This table is identical to the default font-mapping table that the transforms use.

PSSFONTS CSECT			
*			
* TRANSLATE TABLE FROM AND TO RASTER/SCALE FONTS			
* THE SMALLEST VALUES MUST COME FIRST			
*			
	DC C'B20C'	SPECIALS	BOOKMASTER
	DC C'B200'	LATIN1	BOOKMASTER
	DC C'B30C'	SPECIALS	BOOKMASTER ITALIC
	DC C'B300'	LATIN1	BOOKMASTER ITALIC
.			
.			
.			
	DC C'6208'	KATAKANA	GOTHIC
	DC C'7200'	LATIN1	PRESTIGE
	DC C'7300'	LATIN1	PRESTIGE ITALIC
	DC C'7400'	LATIN1	PRESTIGE BOLD
	DC C'8400'	LATIN1	BOLDFACE
	DC C'920A'	OCR A	
	DC C'920B'	OCR B	
	DC AL4(-1)		TERMINATOR
*			
END			

Figure 7-16 Sample font-mapping table -- SYS1.SAMPLIB(AOXFONTS)

Scaling 240-pel to 300-pel fonts using the AOXCF30 program

The AOXCF30 font-conversion program is provided with Infoprint Server Transforms so that you can convert (scale) your currently installed single-byte, bounded-box, 240-pel fonts to 300-pel fonts. This program places the scaled fonts into a partitioned data set, which you can then use with the transform.

The AOXCF30 program can convert 240-pel single-byte and double-byte raster fonts to 300-pel raster fonts. The AOXCF30 program converts a 240-pel raster font to a close approximation of the font at 300-pel resolution. However, IBM does not warrant the quality of the resulting character pattern when using this program.

Specific fixed pitch fonts require the replacement of box characters. These are supplied in the AOX.SAOXCF30 library.

Use the AOXCF30 program only if you do not have a 300-pel font library. If you have previously scaled your 240-pel fonts using the PSF-supplied APSRCF30 program (required for the IBM 4028 printer), you do not need to use this program to scale them again.

Note: The IBM Core Interchange raster fonts are shipped in both 240-pel and 300-pel resolutions; therefore, you do not have to convert them. Outline fonts are resolution-independent and do not need to be converted.



Printer Inventory Definition Utility

This chapter describes how to use the Printer Inventory Definition Utility (PIDU) program to manage objects in the Printer Inventory. Inventory objects include printer definitions, printer pool definitions, components, FSS definitions, and FSA definitions.

This chapter describes:

- ▶ The **pidu** command
- ▶ Using the **pidu** command from the z/OS UNIX shell
- ▶ pidu commands and object classes
- ▶ Examples of pidu commands
- ▶ Using the **pidu** command from a batch job
- ▶ Backing up and restoring the Printer Inventory using the PIDU program

8.1 pidu command for Printer Inventory management

Instead of using the Infoprint Server ISPF panels to manage objects in the Printer Inventory, you can use the Printer Inventory Definition Utility program. The Printer Inventory Definition Utility Program is invoked by the **pidu** command that runs in the z/OS UNIX shell environment.

You might find the **pidu** command useful for creating and editing many objects at the same time. Also, the **pidu** command lets you perform some functions that you cannot perform with Infoprint Server ISPF panels. For example, you can:

- ▶ Export and dump Inventory objects to a file
- ▶ Perform more powerful searches of the Printer Inventory

Before running the **pidu** command, Infoprint Server must be started. The Printer Inventory Manager daemon (**AOPD**) must be active. Other Infoprint Server daemons, IP PrintWay, and NetSpool can also be active when you use the **pidu** command.

Note: To run the **pidu** command, you must have READ access to the AOPADMIN resource profile in the RACF FACILITY class and be a member of the AOPADMIN group. See “Define resource profile AOPADMIN in the FACILITY class” on page 25.

When the Printer Inventory Manager is started, it creates the Printer Inventory files if they do not already exist in directory */var/Printsrv* or in the directory specified in the base-directory statement of the Infoprint Server configuration file **aopd.conf**. To view the Printer Inventory files, use either the Infoprint Server ISPF panels or the **pidu** command.

You can run the **pidu** command in two ways:

- ▶ From the OS/390 UNIX shell using the **pidu** command; see “The pidu command” on page 214
- ▶ As a batch job; see “Running the pidu command as a batch job” on page 223.

The **pidu** command sets the following exit values (return codes):

- ▶ 0 -- The **pidu** commands were performed successfully.
- ▶ >0 -- An error occurred that prevented one or more **pidu** commands from being performed successfully.

8.2 The pidu command

The **pidu** command is used from the z/OS UNIX shell and the syntax is as follows:

```
pidu [-qv] [-c "command ... ;"] ... [filename]...
```

Where:

- c** If you do not specify the **-c** option or the name of a *file*, pidu reads the commands from standard input (stdin), which can be either keyboard data or the output from another command.
- “command ;”** Specifies one or more pidu commands. Enclose the statements in single or double quotes, and end each statement with a semicolon. You can repeat this option. For a list of commands, see “PIDU commands” on page 216.

filename	The name of an HFS file or MVS data set that contains the commands. If the data set is an MVS data set, specify // before the filename. You can repeat this option. When you specify pidu commands in a file by using the filename option, the following applies: Start any comments with a: # Include blank lines if desired
-q	Suppresses informational messages that the pidu command writes to stdout.
-v	Writes the name of the Printer Inventory to stderr. Also provides additional informational messages.

8.2.1 Using the pidu command

The **pidu** command lets you specify one or more of the PIDU commands, shown in Table 8-1 on page 216, to manage objects in the Printer Inventory. You can use the **pidu** command in one of the following ways:

- ▶ Using the **pidu** command with the **-c** option, as shown in Figure 8-1 on page 215
- ▶ Using the **pidu** command interactively, as shown in Figure 8-2 on page 216
- ▶ Using the **pidu** command and placing a PIDU command in a HFS file, as shown in Figure 8-7 on page 219, or an MVS data set, as shown in Figure 8-5 on page 219

The **pidu** command writes any errors to the standard error (stderr) file and writes informational messages and command output to standard output (stdout) file.

Example using the pidu command

From the OMVS shell command line, you can use the pidu command to display a Printer Inventory entry, as shown in Figure 8-1.

```

ROGERS @ SC43: />pidu "-c display printer pokeps ; "

# display printer pokeps
  printer-codepage = IS08859-1
  filters = {
    text -> aopfiltr.so
  }
===>
MORE...
```

Figure 8-1 pidu command issued from the OMVS shell

Example of interactive use of pidu command

You can enter one or more PIDU commands interactively from the keyboard of the z/OS UNIX shell:

- On the z/OS UNIX command line, type **pidu** and press Enter and your shell session screen looks as follows for user ROGERS:

```

ROGERS @ SC43: />pidu
PIDU (AOP1)
>
===>
INPUT

```

Figure 8-2 OMVS Shell after the *pidu* command is entered

- Now, you can enter commands interactively as follows:
 - Type a PIDU command, for example, to display the attributes of the printer definition already created:
display printer pokeps;
 - Type another PIDU command, **list**, to list all the defined printers, where **printer** is an object class, as follows:
list printer;
 - To end the **pidu** interactive session, enter:
exit

Note: For each command you enter interactively, always make sure you end the command with the **;**.

8.3 PIDU commands

As input to the **pidu** command, you must specify one or more PIDU commands, for example:

```
pidu -c "display printer pokeps ; "
```

Where:

display **display** is a PIDU command, as shown in Table 8-1
printer printer is an object class, as shown in Table 8-2 on page 217
pokeps pokeps is the name of a printer definition in the Printer Inventory

Table 8-1 summarizes the PIDU commands and the function of each command.

Table 8-1 Summary of PIDU commands

PIDU commands	Function of the commands
create	Create a new object
delete	Delete an object
display	Display the attributes of an object
dump	Dump all objects
export	Generate create commands for all certain criteria
force-create	Create an object and replace an object of the same name and in the same object class if it exists

PIDU commands	Function of the commands
list	List all objects or only objects
modify	Modify attributes of an object
rename	Rename an object

Note: You can abbreviate command names by using enough characters to make the command name unique. For example, you can abbreviate *display* as *di*.

8.3.1 PIDU command object classes

Most PIDU commands require that you identify the object class of the Inventory object. Table 8-2 summarizes how the PIDU program classifies Inventory objects.

Table 8-2 *Classes of Objects used by the pidu command*

Object class	Description of the object
allocation	An allocation component
fsa	An FSA definition
netspool-eof-rules	A NetSpool end-of-file component
netspool-options	A NetSpool options component
printer	A printer definition
printer-pool	A printer pool definition
printway-fss	An FSS definition for an IP PrintWay functional subsystem
printway-options	An IP PrintWay options component
processing	A processing component
protocol	A protocol component
psf-fss	An FSS definition for a PSF for OS/390 functional subsystem

Figure 8-3 on page 218 shows most of the object classes. So when a command refers to an object class, it refers to the definitions that are only in that object class. The object classes that are shown in Figure 8-3 on page 218 are as follows:

- printer, allocation, processing, netspool-options, netspool-eof-rules, printway-options, and protocol

A0PIPDC1 IP PrintWay Printer Definition		
Command ==> _____		
Printer definition name . _____		
Description . _____		(extend)
Location . . . _____		(extend)
Section	Component name (enter to list)	Custom values (enter to customize)
Allocation	=> _____	=> _____
Processing	=> _____	=> *
NetSpool options	=> _____	=> _____
NetSpool end-of-file	=> _____	=> _____
IP PrintWay options	=> _____	=> _____
Protocol	=> _____	=> *
_ Use DEST, CLASS, and FORMS for IP PrintWay printer selection		
NetSpool LU name . _____ LU classes . . . _____		(extend)

Figure 8-3 IP PrintWay Printer Definition panel showing object classes

8.3.2 Using PIDU commands

The PIDU commands, shown in Table 8-1 on page 216, have the following operand structure:

command objectclass name attribute

Where:

command A PIDU command shown in Table 8-1 on page 216.

objectclass The class of the object that you want to create. Valid values are: allocation, fsa, netspool-eof-rules, netspool-options, printway-fss, psf-fss, printer, printer-pool, printway-options, processing, and protocol. See Table 8-2 on page 217 for a description of these object classes.

name A name to identify the object. Objects in the same object class cannot have the same name. Also, a printer pool definition (printer-pool object class) cannot have the same name as a printer definition (printer object class).

When you create a printer definition, printer-pool definition, or component, specify any combination of 1 to 17 letters (a-z, A-Z), numbers (0 to 9), and special characters (such as \$, #, @, ., -, =, and /). Blank characters are not allowed. The name is case sensitive.

When you create an FSS or FSA definition, this name must match the name used to define the FSA or FSS to JES. Lowercase letters are converted to uppercase.

If the name contains special characters, enclose the name in single or double quotation marks.

attribute One or more attributes and values. Separate attributes with spaces.

Examples of PIDU command file input

The **create** command creates a printer definition, component, printer-pool definition, FSS definition, or FSA definition in the Printer Inventory, as shown in Table 8-2 on page 217. The Inventory object must not already exist in the Printer Inventory. If an Inventory object does exist and you want to modify it, use the **force-create** command. It performs the same function as the create command except that if an object of the same name and in the same object class exists, it is replaced.

You may edit PIDU commands into a HFS file or an MVS data set, for example a data set named **pidu.command**.

MVS data set example

To create a new object or printer definition for a printer named poketest, and using the PIDU command **create**, define the printer definition shown in Figure 8-4 into an HFS file, for example **pidu.command**, or into an MVS data set; for example, **ROGERS.PIDU.COMMAND**.

```
create printer poketest
location="Room 2c-16 Bldg 8" description="3130 Postscript printer"
printer-codepage=ISO8859-1 filters={ text -> aopfiltr.so }
printer-type=ip-printway print-queue-name=afccu2 printer-ip-address =
9.12.2.4 forms=STD output-class=J destination=POKETEST
document-formats-supported = { line postscript text pcl }
duplex=yes lpr-restrict-ports=yes ;
```

Figure 8-4 MVS data set copy of the PIDU create command to define a printer definition

Using the data from the MVS data set to create the new printer definition, enter the command on the OMVS shell command line, as shown in Figure 8-5.

```
-----
ROGERS @ SC43: />pidu //pidu.command
AOP062I printer poketest was created.
ROGERS @ SC43: />
```

==>

Figure 8-5 Using the pidu command to create a printer definition

HFS file example

In this example, you create the data set as a file in the HFS and name it **pidu.command** as follows:

```
BROWSE -- /data/pidu.command ----- Line 00000000 Col 001 080
Command ==>                                                                    Scroll ==> HALF
***** Top of Data *****
create printer poketest
location="Room 2c-16 Bldg 8" description="3130 Postscript printer"
printer-codepage=ISO8859-1 filters={ text -> aopfiltr.so }
printer-type=ip-printway print-queue-name=afccu2 printer-ip-address =
9.12.2.4 forms=STD output-class=J destination=POKETEST
document-formats-supported = { line postscript text pcl }
duplex=yes lpr-restrict-ports=yes ;
***** Bottom of Data *****
```

Figure 8-6 HFS copy of pidu create command to define a printer definition

Using a copy of the data set from the HFS in a directory **/u/rogers/data**, enter from the OMVS shell command line, as shown in Figure 8-7 on page 219.

```
ROGERS @ SC43: />pidu /u/rogers/data/pidu.command
AOP062I printer poketest was created.
ROGERS @ SC43: />

==>
```

Figure 8-7 Using the pidu command to create a printer definition

Usage considerations

You could type the create printer examples interactively in the shell session. However, a better way to create inventory entries is to first create an entry or entries using the ISPF application panels. Then, use the PIDU **entries** command to export the entries to a file.

To avoid having to type a complete Printer Inventory entry, do the following:

- ▶ Use the PIDU export command to export all the printer entries to a file. In the example in Figure 8-8, the filename is **print.export**. Notice that we had 51 entries in the Printer Inventory.
- ▶ Now you can edit members of this file to create a new file for a new entry you want to add to the Printer Inventory without having to do all the typing. You just need to change the parameters required for the new definition.

```
ROGERS @ SC43:/u/rogers>pidu -c 'export print.export printer ; '  
AOP065I 51 definitions were exported to print.export.  
ROGERS @ SC43:/u/rogers>ls -al  
total 304  
drwx----- 2 ADMIN SYS1 8192 Jun 22 12:49 .  
drwxr-xr-x 14 RC43 SYS1 8192 Mar 25 11:02 ..  
-rwxr-xr-x 1 ADMIN SYS1 979 Feb 29 1996 .profile  
-rw----- 1 ADMIN SYS1 29 Mar 1 1996 .sh_history  
-rw-r--r-- 1 RC43 SYS1 84543 Feb 22 18:03 Sc.pdf  
-rw-r--r-- 1 RC43 SYS1 43393 Jun 22 12:49 print.export  
ROGERS @ SC43:/u/rogers>  
===>
```

INPUT

Figure 8-8 Using the export command to create a file of printer entries

Now when you edit **print.export**, as partially shown in Figure 8-9 on page 221, you can choose the options you need to define another printer by selecting a member to use as a skeleton.


```

#=====
# Export of inventory AOP1 by ROGERS
# Fri Jun 22 12:49:48 2001
#=====
#-----
# printer color
#-----
create printer color
  protocol-type = ipp
  location = 2C-16
  description = "Infoprint color 8"
  printer-type = ip-printway
  retry-limit = 1
  retry-time = 0000:00:10
  failure-retention-period = 0000:01:00
  successful-retention-period = 0000:00:00
  normal-output-disposition = purge
  hold = no
  forms = STD
  output-class = J
  burster-trimmer-stacker = no
  destination = COLOR
  document-formats-supported = {
    line
    postscript
    text
    ....

```

Figure 8-9 The beginning of the file `print.export`

Example of multiple PIDU commands

To specify two PIDU list commands on the same input line with the `-c` option, enter:

```
pidu -c "list printer; list printer-pool;"
```

PIDU command considerations

When using the PIDU commands, consider the following rules:

- ▶ You must create components *before* you can create any printer definitions that include these components.
- ▶ You must create printer definitions *before* you create any printer pool definitions that list these printer definitions.
- ▶ If you repeat the same attribute in a statement, the PIDU command uses the *last* specification.
- ▶ A printer definition can include one or more components. For example, a printer definition can include one allocation component and one processing component. When a printer definition includes a component, the attributes specified for the component apply to the printer definition as well. To include a component, use the include-object attributes, such as `include-allocation` and `include-processing`.
- ▶ When you create a printer definition and include components, you can override any attribute that is specified in a component. To do this, specify the same attribute on the create statement for the printer definition. The attribute you specify on the create statement applies only to the printer definition; the value of the attribute in the component is not changed.

- If an attribute is specified in a printer definition, and you later change the value for that attribute in an included component, the value in the printer definition remains unchanged.

Create an allocation component example

This example creates an allocation component named **printway**. This allocation component might be included in all printer definitions for printers to be managed by the same IP PrintWay FSA.

```
pidu -c 'create allocation printway
description="JES allocation values for IP PrintWay"
output-class=K ; '
```

The attributes set for this component are:

description This attribute describes the component.

output-class This attribute specifies the output class that is the JES work-selection criterion for the IP PrintWay FSA; in this example, the output class is K. NetSpool and Print Interface will allocate output data sets on the JES spool in class K.

Create a processing component for PCL printers example

This example creates a processing component named **pcl_printer** that contains attributes suitable for printers that can print PCL files.

```
pidu -c 'create processing pcl_printer
description          = "Generic PCL options"
printer-codepage     = ISO8859-1
document-formats-supported = { text pcl modca-p line-data }
filters              = { text -> aopfiltr.so modca-p -> "afp2pcl.dll -c us"
                        line-data -> "afp2pcl.dll -c us" }
resubmit-for-filtering = yes ; '
```

The attributes set in this component are:

description This attribute describes the component.

printer-codepage This attribute specifies ASCII code page ISO8859-1. Specify an ASCII code page so that Print Interface converts EBCDIC data to ASCII before writing data to the JES spool.

document-formats-supported This attribute specifies that the printer accepts text, PCL, MO:DCA-P, and S/390 line data. The printer accepts MOD:CA-P and line data because a transform is associated with this data format in the filters attribute.

filters This attribute requests that Print Interface use the following filters:

aopfiltr.so filter for text data. IBM recommends that you specify this filter for text data. aopfiltr.so is provided with Infoprint Server.

afp2pcl.dll filter for MO:DCA-P (AFP) data and line data. This filter transforms AFP and line data to PCL data.

resubmit-for-filtering This attribute requests that IP PrintWay submit batch output to Print Interface so that AFP data can be transformed to PCL data using the **afp2pcl.dll** filter.

Additional examples of PIDU commands are in the file */usr/lpp/Printsrv/samples/sample.pidl*.

```

#-----
# Create a printway-options component for PCL landscape 17 chars per inch
#-----
create printway-options pcl_land_17cpi
    description          = "PCL Landscape 17cpi"
    retry-limit          = 5
    retry-time           = 5:00
    failure-retention-period = forever
    document-header      = "<esc>E<esc>&l10<esc>(s17H"
    document-trailer     = "<esc>E"
;

#-----
# Create a protocol component for the lpr protocol
#-----
create protocol lpr_options
    protocol-type        = lpr
    print-queue-name     = PASS
;

#-----
# Create a processing component for a PCL printer
#-----
create processing pcl_printer
    description          = "Generic PCL options"
    printer-codepage     = iso8859-1
    document-formats-supported = { text pcl }
    filters = {
        text -> aopfiltr.so
    }
;

```

Figure 8-10 Samples from /usr/lpp/Printsrv/samples/sample.pidl

Note: See *z/OS Infoprint Server Operation and Administration*, S544-5744 for additional examples of the create PIDU command and the other PIDU commands.

8.4 Running the pidu command as a batch job

You can run the **pidu** command as a batch job by using either the AOPBATCH program or the BPXBATCH utility program. IBM recommends using the AOPBATCH program; see “AOPBATCH utility program” on page 54.

To use AOPBATCH to run the **pidu** command, IBM provides sample JCL in the AOPPIDU member of the SYS1.SAMPLIB for running the **pidu** command using the AOPBATCH utility.

```

//AOPPIDU JOB , 'pidu'
/*-----
/* Sample JCL for running pidu in batch.
/* This JCL will use the AOPBATCH utility to run the pidu program
/* to process OS/390 Infoprint Server inventory definitions.
/* Replace hlq with the appropriate value. Use the aopmigr or
/* aopmigrx job to create the hlq.invdefs data set. You may edit
/* the definitions in the data set as needed.
/* STDIN may also point to a sequential dataset containing other
/* Printer Inventory Definition Language (PIDL) statements.
/*-----
//PIDU      EXEC PGM=AOPBATCH,PARM='pidu'
//STDIN     DD DSN=hlq.INVDEFS,DISP=SHR
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
/* STDENV may point to a data set containing environment variables.
/*      Builtin values will work for the default installation.
//STDENV    DD DSN=environment,DISP=SHR

```

Figure 8-11 Sample job AOPPIDU from SYS1.SAMPLIB

PARM= keyword on the EXEC statement

On the EXEC statement PARM= keyword parameter, you can specify the **pidu** command:

```
pidu [-qv] [-c "command ... ;"] ... [filename]..
```

- ▶ Example of an inline command:

```
PARM='pidu "-c list printer ;"'
```

- ▶ Example of a command in a data set:

```
PARM='pidu "'//''vaini.u.cntl(PID)''"'
```

STDIN DD statement

In both cases the STDIN DD statement can be omitted. When the PARM= field includes only the **pidu** command, but no input command designation, the command input comes from the source defined by the STDIN DD statement.

Note: You must use either the *-c* or the *filename* option on the PARM= input. The *filename* option has to be enclosed in double quotation marks when it contains slash (/) characters to ensure that the all parameter data is passed to the pidu instead of being interpreted as program parameters and Language Environment run-time options.

Using BPXBATCH to run the pidu command

If you use the BPXBATCH utility program to run the pidu command, you must always set the PATH, LIBPATH, and NLSPATH environment variables, even if your installation installed Infoprint Server files in default locations. Specify the AOPCONF environment variable if the Infoprint Server configuration file is not in */etc/Printsrv/aopd.conf* or in *\$HOME/.aopconf*.

8.5 Adding a printer definition summary

The PIDU program can be useful for creating a large number of entries in the Printer Inventory. For example, you could use the following steps:

- To help you specify valid PIDU **create** statements, first use the ISPF panels to create a prototype Inventory object and include components.
- Use the **export** command to create a base printer definition file.
- Use ISPF edit or **oedit** to repeat the desired definitions. Modify the definitions as appropriate.

Note: When you create a printer definition and include components, you can override any attribute that is specified in a component. To do this, specify the same attribute on the create statement in a printer definition. The attribute you specify on the create statement applies *only* to that printer definition; the value of the attribute in the component is not changed.

- You must create components *before* you create any printer definitions that include these components.
- You must create printer definitions *before* you create any printer pool definitions that list these printer definitions.
- If you repeat the same attribute in a create statement, the **pidu** command uses the last specification.
- Use the PIDU **force-create** or **create** command to update printer definitions, components, printer-pool definitions, FSS definitions, and FSA definitions in the Printer Inventory.

8.6 Backing up and restoring the Printer Inventory

You should back up the Printer Inventory on a regular basis. IBM recommends that you use the Infoprint Server Printer Inventory Definition Utility (PIDU) **export** command to back up the Printer Inventory to a file. To use PIDU, you must be a member of the AOPADMIN group and you must have READ access to the AOPADMIN resource profile in the FACILITY class.

Do not use ordinary copy commands to create a backup copy of the Printer Inventory because the copy might contain inconsistent data that makes the Printer Inventory unusable.

8.6.1 PIDU export command

You can use the PIDU **export** command to either export all objects in the Printer Inventory, or only those objects that meet specified conditions. You can use the **export** command to back up the Printer Inventory. Also, the statements in the output file can be used as input to the **pidu** command on another z/OS or OS/390 system.

The **export** command writes a create statement for each exported object to an output file. The output file does not need to exist before issuing the command; however, if the file already exists, the file is replaced.

Export objects in the Printer Inventory to a file as follows:

```
export filename [objectclass [where condition [and|or condition]... ] ];
```

export command examples

- This example, issued from the OMVS shell command line, exports all objects in the Printer Inventory to a HFS file called **inventory.export** and in a specified directory in the HFS.

```

ROGERS @ SC43: />cd /u/rogers
ROGERS @ SC43:/u/rogers>pidu -c "export inventory.export ; "
AOP065I 64 definitions were exported to inventory.export.
ROGERS @ SC43:/u/rogers>
===>

```

INPUT

Figure 8-12 Command to back up the Printer Inventory

- This example exports all objects in the Printer Inventory to an MVS data set.

```
pidu -c "export //'ROGERS.INVENTORY.EXPORT\' ; "
```

Note: In order to specify a fully-qualified data set name using the **pidu** command, you have to *escape* the quote so it doesn't get removed by UNIX System Services. The correct syntax for the command, therefore, has a back slash in front of each single quote.

- Because you can put conditions on what Printer Inventory entries can be exported, this example exports all printer definitions that contain attribute **output-class = K**, either in the printer definition or in an included component, to file **classk.export**.

```
pidu -c 'export classk.export printer where output-class=K ; '
```

- As a second example for conditional exporting, this exports selected FSA definitions in the Printer Inventory to file **psffsa.export**. The where predicate specifies that only FSA definitions used by PSF be exported.

```
pidu -c 'export psffsa.export fsa
where fsa-type=psf-tcpip or fsa-type=psf-channel or fsa-type=psf-sna ; '
```

Daily backup

In this example, to back up the Printer Inventory at 3 AM each day to a file **/var/Printsrv/inventory.backup.yymmdd**, where *yymmdd* is the date of the backup, follow these steps:

- Create a file to contain a short shell script, for example **/u/userid/do.backup**, where *userid* is your login user ID.

If you set the **PATH**, **NLSPATH**, and optional **AOPCONF** environment variables with the values required by Infoprint Server in either **/etc/profile** or **/u/userid/.profile** or both, add the following line in these files:

```
extension=$(date + "%y%m%d")
```

The following command exports the Printer Inventory and should be placed into the **do.backup** file.

```
pidu -c "export /var/Printsrv/inventory.backup.$extension;"
```

- Set the execute permission bit for the **/u/userid/do.backup** file. For example:

```
chmod +x /u/userid/do.backup
```

- Use the **crontab** command with the **-e** option to insert the following line into the crontab entry:

```
00 03 * * * /u/userid/do.backup
```

When the cron daemon runs the commands in your crontab entry, it does not first run your .profile to establish environment variables. For more information about the crontab command and the cron daemon, refer to *z/OS UNIX System Services Command Reference*, SA22-7802.

Restore of backup

To restore the Printer Inventory from a backup copy, follow these steps:

- ▶ Log in to z/OS UNIX System Services as root (an z/OS UNIX superuser).
- ▶ Stop NetSpool and IP PrintWay.
- ▶ To stop Infoprint Server daemonsFrom the z/OS UNIX command line, issue the **aopstop** command; or, from an operator console, issue **S AOPSTOP** command.
- ▶ Move the **/var/Printsrv/master.db** and **/var/Printsrv/jestoken.db** files to another directory. Save these files until you have restored the Printer Inventory, restarted Infoprint Server, and run with the restored Printer Inventory for a few days.
- ▶ Issue the **aopstart** command to restart Infoprint Server daemons.

Note: You cannot run the **pidu** command successfully if the Printer Inventory daemon (**aopd**) is not running.

- ▶ Issue the **pidu** command to restore the Printer Inventory. The following example shows how to restore the Printer Inventory from file **/var/Printsrv/inventory.backup.yymmdd**. This command uses the UNIX redirection symbols (> for stdout and 2> for stderr) to redirect command output to file **pidu.output**, and command errors to file **pidu.errors**:

```
pidu /var/Printsrv/inventory.backup.yymmdd >pidu.output 2>pidu.errors
```
- ▶ Issue the **aopstart** command to restart Infoprint Server Printer Inventory.
- ▶ Restart NetSpool and IP PrintWay.

Note: When you restore the Printer Inventory, no print jobs are lost. However, information in the Inventory about jobs that were submitted through Print Interface is lost. This means that users, including SAP R/3 users, are not notified when these jobs are completed, and users cannot query the status of these jobs.



Infoprint Server and SAP R/3

This chapter describes the capability to print data sets from SAP R/3 systems to the Infoprint Server support provided by the Print Interface.

It covers the following:

- ▶ Printing on SAP R/3 systems
- ▶ Implementing printers in z/OS Infoprint Server
- ▶ Setting up printing in SAP R/3

9.1 Printing on SAP/R3 systems

There are several ways to print output on SAP R/3 systems. Depending on whether you have high volume printing or desktop printing, for example, you might choose between local output to a host spool or output to SAPLPD on your desktop machine.

The local host spool access methods relevant to our configuration are:

- L** The SAP spool process uses host commands to print output that it saved in the host filesystem before. The host commands for printing and status report are saved in the instance parameters `rspo/host_spool/print` and `rspo/host_spool/query`.
- E** This mode can be used for printers defined in an Output Management System (OMS) that are suitable for connection to the SAP R/3 system. OMS then allows the user to track and manage print jobs that are already released from the SAP R/3 spool.
- P** This mode is used to define device pools that allow printing to be done to several devices at the same time.
- F** This mode is used to spool output to a spool program on the user's workstation via the dialog process the user is connected to. This spool program can be, for example, SAPLPD on Windows or `lpd` on UNIX workstations.

With a spool process on the S/390 host, you can take advantage of the printing capabilities of Infoprint Server for z/OS by defining your printers in either mode L or mode E. We followed both approaches in order to document the installation process.

9.2 Implementing printers in z/OS Infoprint Server

A printer can be defined in the Printer Inventory as an IP PrintWay printer, or a Print Services Facility (PSF) printer, or as any printer controlled by JES, as shown in Figure 9-1 on page 231. If you define a printer as an IP PrintWay printer, its output is spooled to JES and from there transferred by IP PrintWay to a printer supporting one of the following protocols: LPR-to-LPD, IPP, or direct-socket printing.

PSF printers write output in Advanced Function Printer (AFP) format to JES spool; from there the output is sent to an AFP printer device. If the output you want to print happens to be in a format other than AFP, the PSF printer definition allows you to name transform filters. IBM Infoprint Server Transforms for z/OS provides filters that allow your data format to be Printer Control Language (PCL), PostScript (PS), Portable Document Format (PDF), or SAP Output Text Format (OTF).

To print to IBM AFP printers on the z/OS system, the following software products are required:

- ▶ Infoprint Server Transforms (5697-F51). The base feature provides the SAP-to-AFP transform at no extra cost to Infoprint Server customers.
- ▶ PSF V3R1.0 for OS/390 (5655-B17), or later.

As shown in Figure 9-1 on page 231, when the SAP R/3 application server runs on z/OS, the Print Interface OMS receives print and status requests, and the Callback daemon provides immediate notification of job events. The Print Interface Callback daemon can return notification of job events to SAP R/3 application servers running on other SAP R/3 systems, provided the SAP spool work process runs on the z/OS system.

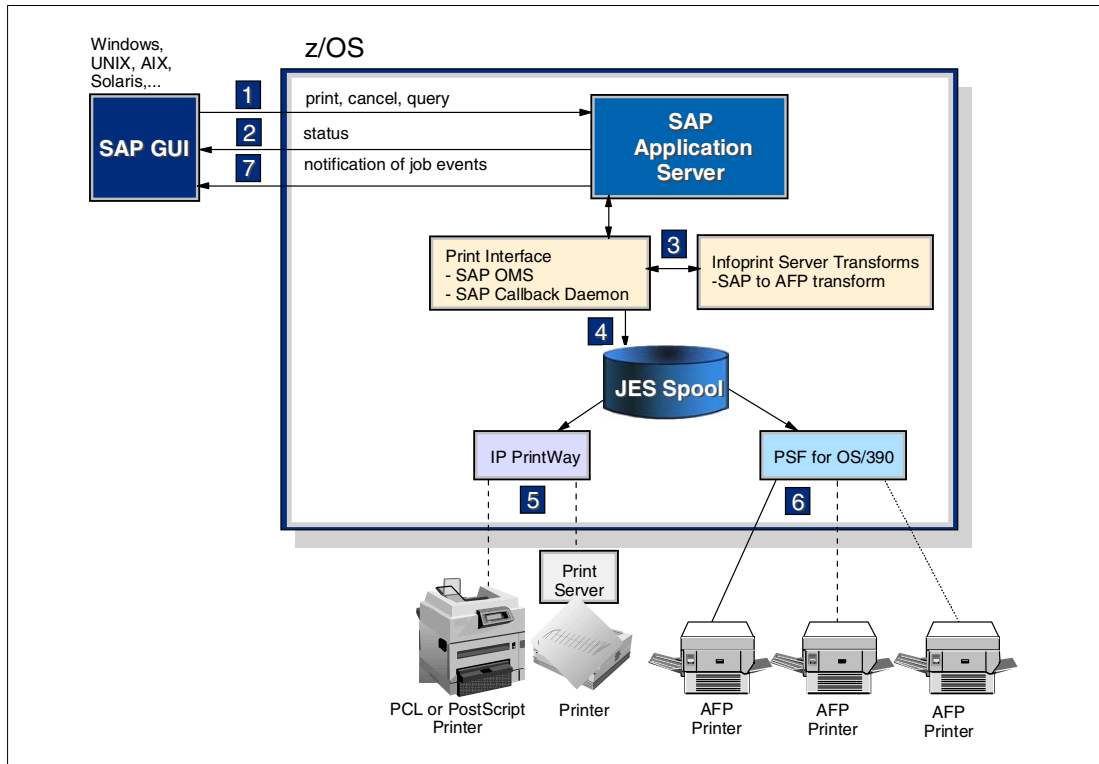


Figure 9-1 Overview of SAP/R3 and Infoprint Server

9.2.1 Implementing the PSF printer

The PSF printer we used in our installation to test printing from SAP R/3 is configured to support plain text format, AFP, PCL, PS, PDF, and OTF. The physical printer that is the ultimate destination of the output is an IBM 3130.

In order to configure the printer, we start the Infoprint Server Printer Inventory Manager, select **Add** on the first panel and **PSF for OS/390** in the following dialog. On the printer definition panel, which you can see in Figure 9-2, we entered the printer definition name. In this panel, custom values for allocation and processing can be set by moving the cursor to the appropriate row in the custom values column and pressing Enter.

```

AOPIPDC2                                PSF for OS/390 Printer Definition
Command ==> _____

Printer definition name . pokeall_____
Description . PSF printer for SAP/R3_____ (extend)
Location. . . 2C-16_____ (extend)

Section                                Component name          Custom values
Allocation                            => _____          => _____
Processing                            => _____          => *
NetSpool options                       => _____          => _____
NetSpool end-of-file                   => _____          => _____

NetSpool LU name . _____ LU classes . . . . . (extend)

```

Figure 9-2 PSF printer definition panel

In the allocation panel (not shown here), we set the CLASS parameter to U and the DEST parameter to POK3130E; these are the appropriate values for our installation. Our settings for the processing options are shown in Figure 9-3. We selected all input formats and inserted the appropriate transform filters. For all other settings, we used the defaults.

AOPIPDRR

Processing

Command ==> _____

Printer definition name . pokeall _____

More: +

Document code page . . _____

Printer code page. . . IBM-1047 _____

Print Interface supported data formats and associated filters:

Data format:

Filter:

/ Line data _____ (extend)

/ MO:DCA-P _____ (extend)

/ PostScript ps2afp.dll _____ (extend)

/ Text _____ (extend)

/ PCL pcl2afp.dll _____ (extend)

/ PDF ps2afp.dll _____ (extend)

/ SAP sap2afp.dll _____ (extend)

/ Other _____ (extend)

_ Resubmit for filtering

Conversion between SCS and Line:

Margins: Top _____ Bottom _____ Left _____ Right _____

Line length _____ Page length _____

Tabs: Vertical _____ (extend)

Horizontal _____ (extend)

IP PrintWay Line-to-Text Conversion:

/ Pagination

Margins: Top _____ Bottom _____

Page height 58 _____

Figure 9-3 PSF printer definition: part of Custom Values/Processing panel

9.2.2 Implementing the IP PrintWay printer

The physical printer we used was an IBM 3130. In this case, it provided a queue named afccu2 which was configured to accept input in plain text, PCL, and PS. It communicates with hosts using the LPR/LPD protocol.

In Figure 9-4, the entry panel to the IP PrintWay definition is displayed. It differs from the PSF panel in Figure 9-2 on page 231 by offering additional options.

AOPIPDC1

IP PrintWay Printer Definition

Command ==> _____

Printer definition name . pokepl _____

Description . SAP/R3 IP PrintWay printer _____ (extend)

Location. . . 2C-16 _____ (extend)

Section

Component name
(enter to list)

Custom values
(enter to customize)

Allocation ==> _____ ==> *

Processing ==> _____ ==> *

NetSpool options ==> _____ ==> _____

NetSpool end-of-file ==> _____ ==> _____

IP PrintWay options ==> _____ ==> *

Protocol ==> _____ ==> *

_ Use DEST, CLASS, and FORMS for IP PrintWay printer selection

NetSpool LU name . _____ LU classes _____ (extend)

Figure 9-4 IP PrintWay printer definition table

232 z/OS Infoprint Server Implementation

The only values we set in the Allocation/Custom values-panel (which is not displayed here) were CLASS = J and FORMS = STD. Note that these parameters link to JES definitions for the CLASS and could be different in other installations. The class used must be one specified for IP PrintWay printing in JES initialization. For more detail see *z/OS Infoprint Server Customization*, S544-5745.

The changes we made to the IP PrintWay processing panel are displayed in Figure 9-5. We selected only the data formats that were defined in the queue afccu2. The filter aopfiltr.so is only used to prepare text for output on an ASCII printer.

AOPIPDRR		Processing	
Command ==> _____			
Printer definition name . <u>pokepl</u> _____			
			More: +
Document code page . . _____			
Printer code page. . . <u>ISO8859-1</u> _____			
Print Interface supported data formats and associated filters:			
Data format:	Filter:		
<input type="checkbox"/> Line data			(extend)
<input type="checkbox"/> M0:DCA-P			(extend)
<input checked="" type="checkbox"/> PostScript			(extend)
<input checked="" type="checkbox"/> Text	<u>aopfiltr.so</u>		(extend)
<input checked="" type="checkbox"/> PCL			(extend)
<input type="checkbox"/> PDF			(extend)
<input type="checkbox"/> SAP			(extend)
<input type="checkbox"/> Other			(extend)
<input type="checkbox"/> Resubmit for filtering			
Conversion between SCS and Line:			
Margins: Top	Bottom	Left	Right
Line length		Page length	
Tabs: Vertical			(extend)
Horizontal			(extend)
IP PrintWay Line-to-Text Conversion:			
<input checked="" type="checkbox"/> Pagination			
Margins: Top	Bottom		
Page height	<u>58</u>		

Figure 9-5 IP PrintWay: part of the Processing panel

The panel shown in Figure 9-6 on page 234 is used for protocol settings. The protocol is lpr by default; we inserted the printer IP address and the queue name and selected to restrict the ports used to 721-731. z/OS uses these ports when communicating with the remote printer.

AOPIPDIL		LPR Protocol	
Command ==> _____			
Printer definition name . <u>pokepl</u> _____			
Printer IP address . <u>9.12.2.4</u>		(extend)	
Print queue name . . <u>afccu2</u>		(extend)	
LPR Processing Options:			
Mode	<u>2</u>	1. Control file first	2. Control file last
		3. Stream	4. Remote PSF
<input checked="" type="checkbox"/> Restrict ports			
<input checked="" type="checkbox"/> Print banner page			
Banner class . .	_____		(extend)
Banner job name	_____		(extend)
Filename	_____		
Indent	_____		
Owner	_____		
Print function . . .	<u>f</u>		(extend)
Title	_____		
Width	_____		
User options	_____		
			(extend)

Figure 9-6 IP PrintWay: part of the Protocol panel

With these settings and a properly set up Infoprint Server, we were able to print files from within OMVS, issuing commands such as:

```
lp -d pokepl <filename>
```

9.3 Setting up printing in SAP R/3

To be able to use Infoprint Server with OMS, make sure that APAR OW43022 is installed in your system. We now describe the setup for R/3 to be able to print using host spool access methods **L** and **E**, as described in “Printing on SAP/R3 systems” on page 230.

9.3.1 Setting up Infoprint Server as an OMS

There are two executables in `/usr/lpp/Printsrv/bin`. They are `aopsapd` and `aopoms`.

`aopsapd` is the callback daemon responsible for updating job status in the SAP R/3 system. It expects to find a configuration file called **aopsapd.conf** in `/etc/Printsrv`, for which a model can be found in `/usr/lpp/Printsrv/samples`. Figure 9-7 on page 235 shows our adaptation of this file. The daemon can connect to several SAP R/3 systems, which then all have to be defined in the configuration file.

Software dependencies

To use the SAP OMS, the following software products and PTFs are required:

- ▶ Infoprint Server PTF UW68519 and PTF UW73315
- ▶ The SAP R/3 Application Server for z/OS, an SAP product

The SAP R/3 Application Server for z/OS spool work process must run on the same z/OS system as Infoprint Server.

```

# aopsapd.conf
#
# Sample configuration for the Infoprint Server SAP OMS V1.0 included with
# Infoprint Server for OS/390. One system entry is required for each R/3
# application server system to which the callback daemon is to connect to.
# The initial-target attribute names the server to which the callback
# daemon will initially connect for the system. This initial connection
# will be used to automatically retrieve configuration information for the
# Infoprint Server LOMs configured on that system.
#
# Although the initial-target may be a non-OS/390 server, it is necessary
# that the initialization instance for the ROMS, and the tasking target
# for the LOMs all be located on an OS/390 server where the Infoprint
# Server is running.
#
#-----
system SC42
    initial-target = WTSC420E_RED_00
    client = 000
    userid = sapres6
    password = basis
    language = EN
;

```

Figure 9-7 /etc/Printsrv/aopsapd.conf

For Logical Output Management System (LOMS) using callback, a dialog user logs on to the R/3 System. This user updates the status of print requests directed to printers managed by an external output management system (using the host spool access method E). The dialog user defined for the client that uses callback need only have SAP R/3 authorization S_XMI_XOM_A. The user must be defined in **aopsapd.conf**.

As aopsapd.conf contains a nonencrypted password, the file should be made readable only to z/OS UNIX user redadm, who issues the **aopsapd** command.

The **aopsapd** command is stopped by the command **aopstop**, but not started by the **aopstart** command. It can be started in one of the following ways:

- ▶ Manually.
- ▶ Automatically by the SAP R/3 system as configured in the ROMS.
- ▶ It is started whenever status updates occur for a job and aopsapd is not yet started.

The **aopoms** command is the interface which the R/3 system uses to communicate with the OMS in Infoprint Server in order to:

- ▶ Submit output requests
- ▶ Cancel output requests
- ▶ Synchronously get status of output requests
- ▶ Synchronously get status of output devices

It is also used when the job status is tracked using polling rather than callback.

Another step in this process is to make an EBCDIC version of librfc available, either by putting it in **/usr/lpp/Printsrv/lib**, or by setting up LIBPATH to contain the location of librfc that comes with R/3 4.6B. We chose the first approach.

9.3.2 Host spool access method L

Configuration of a printer in SAP R/3 using host spool access method L is straightforward and does not need the implementation of an OMS.

We used transaction *spad* and, after changing to edit mode, clicked **Output Devices**. This led to a list of already configured devices. We chose **Output Device -> Create** from the menu and entered into the following panels the values that are displayed in Figure 9-8 to Figure 9-10 on page 237.

The screenshot displays the SAP 'Spool Administration: Output Device (Display)' window. The 'DeviceAttributes' tab is active, showing the following fields and values:

- Device type: SAPGOF_E : Generic output format EBCDIC
- Spool server: wtsc42oe_RED_00
- Device class: Standard printer
- Authorization group: (empty)
- Model: (empty)
- Location: (empty)
- Message: (empty)

The 'Lock printer in SAP System' checkbox is unchecked. The status bar at the bottom indicates 'RED (1) (000) wtsc42oe INS'.

Figure 9-8 SAP printer definition: Device Attribute

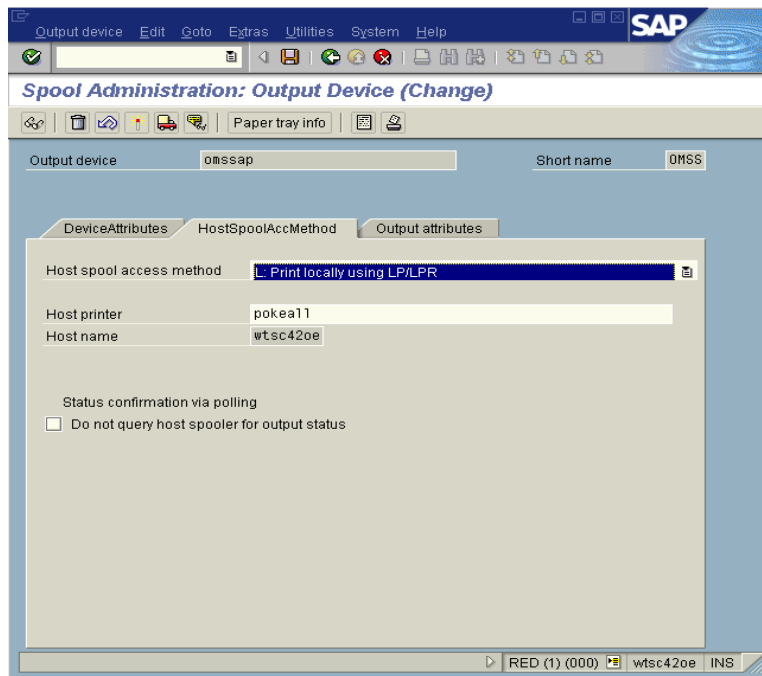


Figure 9-9 SAP printer definition: Host Spool Access Method

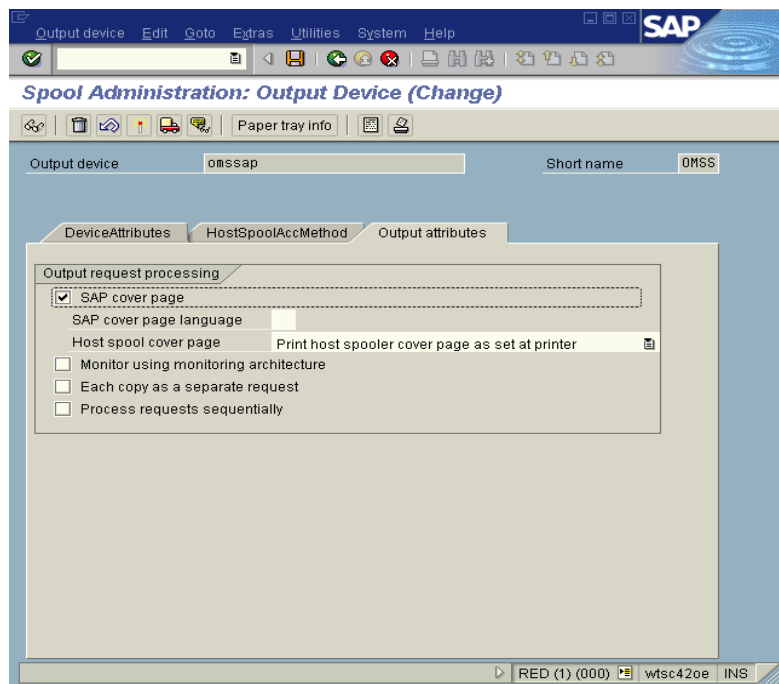


Figure 9-10 SAP printer definition: Output attributes

Specifying the device type SAPGOF_E tells SAP R/3 to create output in Output Text Format (OTF). In this case, we choose L for the Host Spool Access Method and the PSF-defined printer pokea11.

Having the system create both SAP R/3 and host spool cover pages turns out to be useful when you test printing, since the SAP R/3 page has the job ID reported in transaction sp01 on it, and the host spool banner page has the JES spool job ID printed on it.

Table 9-1 lists the device types you can select in your definition of an SAP R/3 printer. We tested these on the printers we defined on the host. Only the printing of the SAP R/3 generic output format (SAPGOF) was not successful on pokelp due to the lack of an appropriate driver.

Table 9-1 SAP device types tested on our printers

SAP device type	pokeall	pokelp
POST2	X	X
HPLJ4	X	X

9.3.3 Host spool access method E

This mode can be used for printers defined in an OMS that are suitable for connection to the SAP R/3 system. OMS then allows the user to track and manage print jobs that are already released from the SAP R/3 spool

After performing the tasks described in “Setting up Infoprint Server as an OMS” on page 234, the next thing we do in order to use our printers with access method E is to set up both an ROMS and an LOMS within SAP R/3.

There is a model configuration file called infoprint.oms that you can find in */usr/lpp/Printsrv/samples*. This file should be checked to be certain the settings reflect your configuration and that the paths for the binaries are correct. Edit the file to make any changes appropriate to your installation.

Then, we call transaction spad and choose from the menu **Utilities -> For output management system -> Import**. Enter the configuration file and select **Execute import** as shown in Figure 9-11 on page 239, then click **execute**.

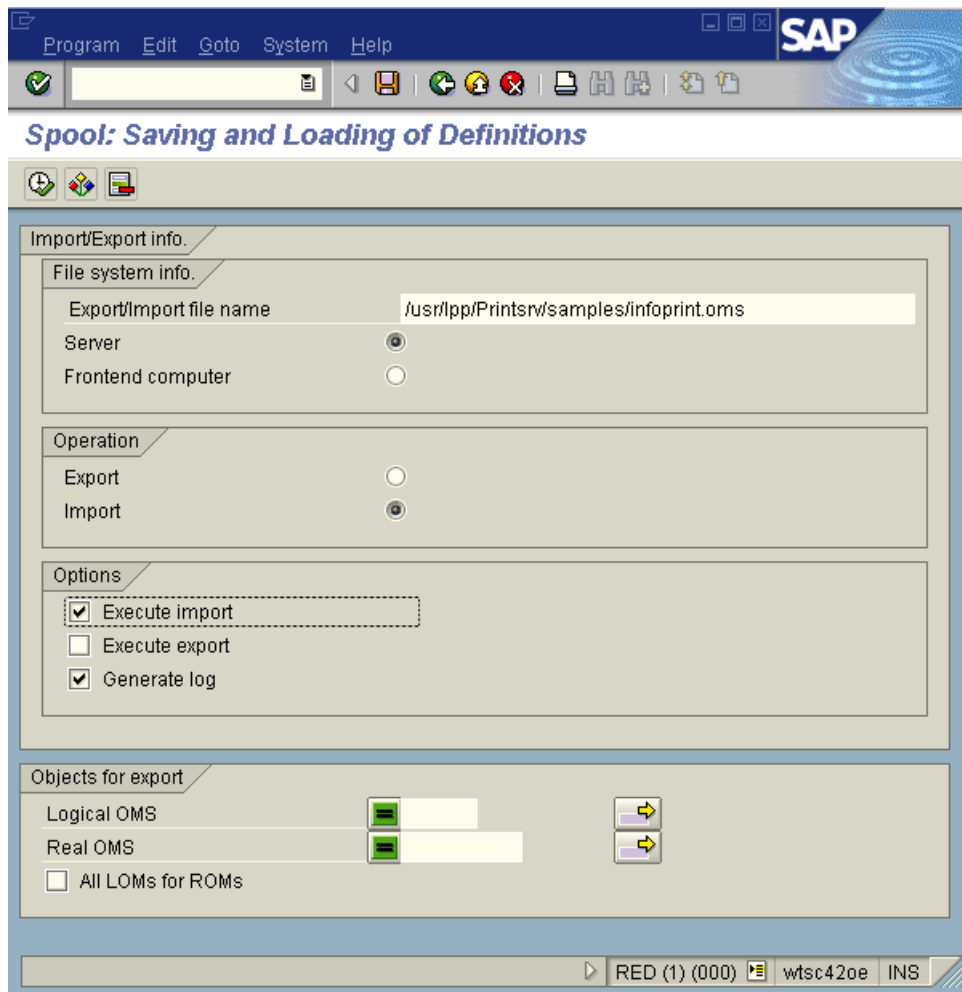


Figure 9-11 Import of the OMS configuration

After the import of the file, the panels for the real OMS (ROMS) and the logical OMS (LOMS) must be checked for their settings. For that, we click either the Extended administration button or the full administration button in transaction spad and select the tag **Output management systems**. From there, we can select the configured ROMS and LOMS.

Figure 9-12 on page 240 shows the ROMS configuration panel of our system. No initialization instance or initialization command need to be entered here.

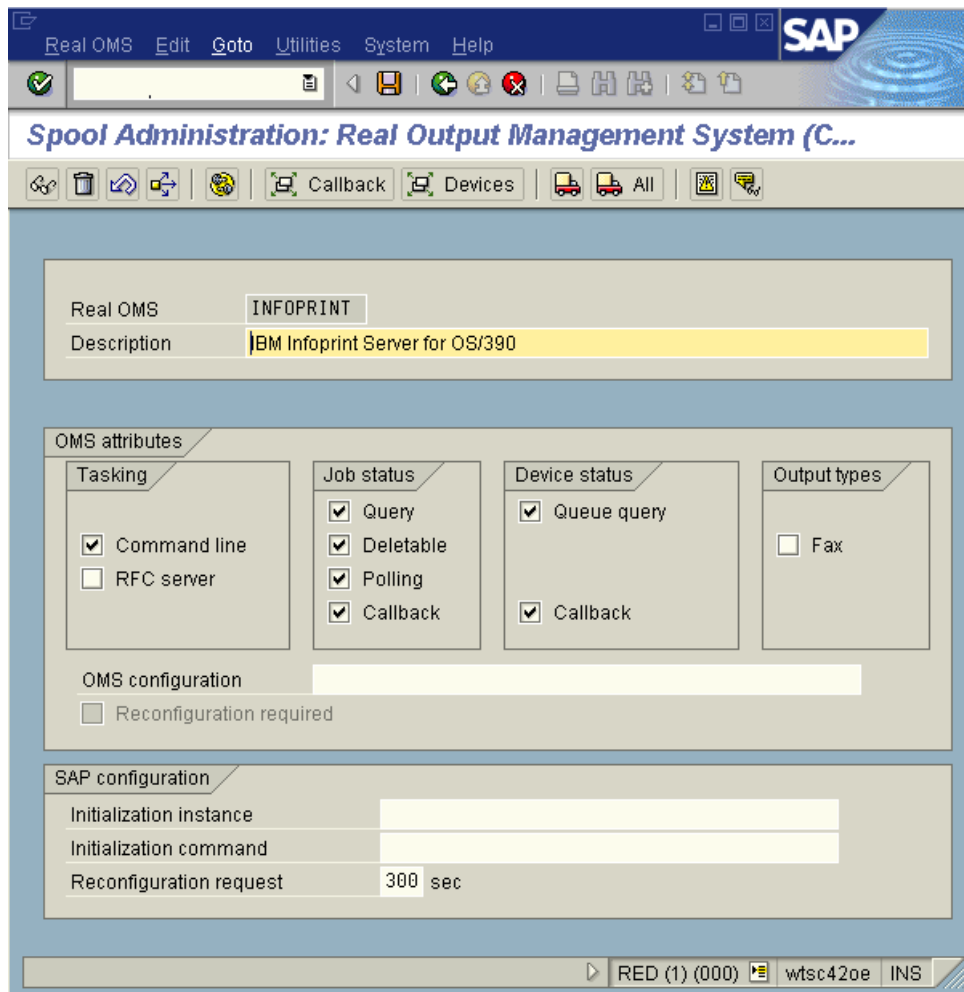


Figure 9-12 ROMS configuration panel

After the ROMS, the LOMS has to be configured. On the SAP R/3 configuration panels shown in Figure 9-13 on page 241 and Figure 9-14 on page 241, the tasking target and target for callback are both set to be the z/OS server, since all printing commands will be issued on z/OS UNIX and the callback daemon will be started on z/OS UNIX. For testing, we set the event report level to All available information. On the OMS configuration panel, nothing has to be changed.

The last items to be checked in the OMS setup process are the commands that SAP R/3 issues on the printing host. To perform this check, we pressed F6 when one of the LOMS panels shown in Figure 9-13 on page 241 was open; we then selected the operating system. Next, we checked that the paths to the executable **aopoms** are correct. The command path field should be left blank.

Note: We have been told to expect a change in this area to be implemented in a PTF whose number is not yet available. Then the command path should be /usr/lpp/Printsrv/bin/ (note the trailing /), which will mean it is no longer necessary to use the complete path in the commands; you will simply use aopoms.

For printer setup, we chose one of the printers we defined before for access method L and changed it (using the panel displayed in Figure 9-9 on page 237) to E: external management system. This creates a new field on the panel called Logical OMS, where we selected **Infoprint Server with callback**.

Logical OMS

IPS001

Description

Infoprint Server with callback

SAP configuration

OMS configuration

Real OMS

IBM Infoprint Server for OS/390

Tasking target

wtsc42oe_RED_00

Target for callback

wtsc42oe_RED_00

Command group

Event report level

All available information

Tasking

Commands

Devices

Queue query

Callback

Delete events

Jobs

Query

Can be deleted

Status dialog boxes

Callback

Polling

Figure 9-13 First LOMS configuration panel

Logical OMS

IPS001

Description

Infoprint Server with callback

SAP configuration

OMS configuration

OMS callback cache

Print requests

Send period

60

Sec

Number of events

100

OMS devices

Send period

60

Sec

Number of events

100

Restart after failure of callback target

Interval

120

Sec

OMS configuration

Reconfiguration of device list

Figure 9-14 Second LOMS configuration panel

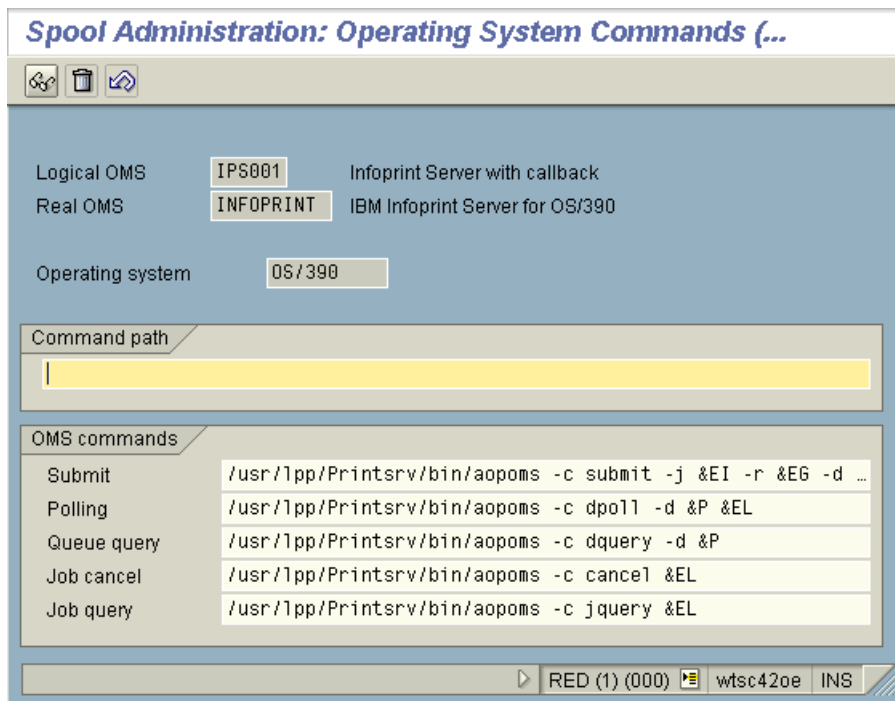


Figure 9-15 LOMS Operating System Commands panel

From the SAP GUI, users make print, cancel, job query, and device query requests to the SAP R/3 Application Server. These print requests specify the name of an SAP R/3 output device defined to the SAP R/3 system. The SAP administrator associates each output device with a printer definition in the Infoprint Server Printer Inventory.

For a Queue query request, the Print Interface SAP OMS returns the status of a print job or a list of print jobs that the printer is processing.

For a Submit request, Print Interface detects the data format of the input document and performs different processing, such as data stream transforms, depending on the type of data received.

As print jobs complete, successfully or unsuccessfully, the Print Interface SAP Callback daemon sends notification back to the SAP R/3 system.



Dynamic IP PrintWay message exit sample

This appendix provides an example of a IP Printway message exit, and shows how to use the MVS dynamic exit facility to transform a static exit to a dynamic exit.

The dynamic exit facility is a set of MVS services that allow an installation dynamically to add one or more exit routines to an exit point, delete an exit routine for an exit point, change the state of an exit routine, change the attributes of an exit point, and undefine an implicitly defined exit point. The dynamic exit facility actions on exits take place immediately and do *not* require the exit owner application to be restarted.

The dynamic exit facility is implemented by:

- ▶ The EXIT statement of the PROGxx parmlib member. The PROGxx EXIT statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, operators can use PROG=xx to specify the particular PROGxx parmlib member the system is to use. During normal processing, operators can use the SET PROG=xx command to set a current PROGxx parmlib member.
- ▶ The SETPROG EXIT operator command. This command performs the same functions as the EXIT statement of the PROGxx parmlib member.
- ▶ The CSVDYNEX macro.

Two assembler modules make up the sample IP PrintWay message exit:

- ▶ ANFUXMSG is a dynamic exit stub that exploits the MVS dynamic exit facility services to define the actual IP Printway message exit routine called UXMSGANF. Refer to Example on page 244.
- ▶ UXMSGANF is the actual IP Printway message exit routine that reissues all messages passed to the exit with WTO such that the messages are written to the hardcopy log (SYSLOG and/or OPERLOG). Refer to Example on page 247.

Note: The modules contain code on an AS IS basis.

ANFUXMSG - dynamic exit stub

Using the IP PrintWay supplied user exit, ANFUXMSG, you can have all IP PrintWay messages that are displayed in the IP PrintWay message data set sent to the SYSLOG data set with the code shown in Example A-1:

Example: A-1 IP PrintWay ANFUXMSG dynamic exit stub

```
ANFUXMSG TITLE 'IP PRINTWAY ANFUXMSG DYNAMIC EXIT STUB'
ANFUXMSG CSECT
ANFUXMSG AMODE 31
ANFUXMSG RMODE ANY
        YREGS

*
* PrintWay calls the same Message exit for all data sets. You can
* install different Message exits for different IP PrintWay FSSs.
*
* To install an exit, compile and link-edit the exit as follows:
*
* - Name the exit:
*
*   - For the Message exit, use the name ANFUXMSG.
*
* - Link-edit the exit with the RENT and AMODE 31 attributes into
*   an AFP authorized library.
*
* Identify the library containing an exit in either:
*
* - A STEPLIB statement in the IP PrintWay startup procedure.
*
*   You can provide a different exit routine for each IP PrintWay
*   FSS by using a unique STEPLIB for each startup procedure.
*
* - A TASKLIB, STEPLIB, or JOBLIB statement concatenated with
*   SYS1.LINKLIB.
*
* This dynamic exit stub drives the actual IP PrintWay Message Exit
* ANFUXMSG.
*
* The actual exit - called UXMSGANF - should be added through
* parmlib PROGxx member or be defined through the SETPROG operator
* command:
*
*   PROGxx:
*       EXIT ADD
*           EXITNAME(ANFUXMSG)
*           MODNAME(UXMSGANF)           /* <== Change */
*           STATE(ACTIVE)
*           JOBNAME=PRINTW*           /* <== Change */
*
*   - or -
*
*       SETPROG EXIT,ADD,EXITNAME=ANFUXMSG,MODNAME=UXMSGANF,
*           STATE=ACTIVE,JOBNAME=PRINTW*
*
* At entry to the Message exit, the registers contain the following:
*
* Register 1  Pointer to the address of the ANFUEXTTP control block.
* Register 13 Address of an 18-word save area for saving the caller's
*             registers.
* Register 14 Return address.
```



```

* Register 15 Entry-point address.
*
        SAVE (14,12),,ANFUXMSG_&SYSDATE._&SYSTIME
        LR R12,R15          Set base reg
        USING ANFUXMSG,R12
        LR R3,R1            Save XTP pointer pointer
        L R2,0(,R1)         Get address of the ANFUEXTP
        USING ANFUEXTP,R2
        OI XTPMSGFL,XTPM2MDS Issue message to message data set
        LA R0,WSLENGTH      GET LENGTH OF WORKING STORAGE
        GETMAIN R,LV=(0)     GETMAIN WORKING STORAGE
        LR R4,R1            ADDRESS OF WS TO R4
        LR R6,R1            ADDRESS OF WS TO R6
        LA R5,WSLENGTH      GET LENGTH OF WORKING STORAGE
        SLR R7,R7           CLEAR PATTERN REGISTER
        MVCL R4,R6          CLEAR WORKING STORAGE
        ST R1,8(,R13)       SAVE ADDR OF NEW SAVE IN OLD
        ST R13,4(,R1)       SAVE ADDR OF OLD SAVE IN NEW
        LR R13,R1           SETUP NEW SAVE/WS
        USING WS,R13        TELL ASSEMBLER

*
* QUERY EXIT
*
        MVC ERRMSGF,=CL6'QUERY'
        XC LRSNCODE,LRSNCODE
        XC LRETCODE,LRETCODE
        CSVDYNEX REQUEST=QUERY,EXITNAME=LEX,QTYPE=CALL,          C
                RETCODE=LRETCODE,RSNCODE=LRSNCODE,WORKAREA=WSDYNQRY, C
                MF=(E,WSDYNEX)
        NC LRSNCODE,=AL4(CSVDYNEXRSNCODEMASK) AND OFF EXTRA BITS
        CLC LRETCODE,=AL4(CSVDYNEXRC_OK)
        BE CALLROUT        EXIT AND MODULE AVAILABLE
        CLC LRETCODE,=AL4(CSVDYNEXRC_WARN)
        BNE ERRWTO
        CLC LRSNCODE,=AL4(CSVDYNEXRSNNOMODULES)
        BE RETURN00        NO MODULES IN EXIT ... END
        CLC LRSNCODE,=AL4(CSVDYNEXRSNQUERYNOTFOUND)
        BE DEFROUT        EXIT NOT DEFINED ..... GO DEFINE IT
        CLC LRSNCODE,=AL4(CSVDYNEXRSNIMPLICITLYDEFINED)
        BE DEFROUT        EXIT ONLY IMPLICIT ... GO DEFINE IT
        B ERRWTO

*
* DEFINE EXIT
*
DEFROUT DS OH
        MVC ERRMSGF,=CL6'DEFINE'
        XC LRSNCODE,LRSNCODE
        XC LRETCODE,LRETCODE
        CSVDYNEX REQUEST=DEFINE,EXITNAME=LEX,AMODE=31,PERSIST=IPL, C
                FASTPATH=NO,RETCODE=LRETCODE,RSNCODE=LRSNCODE, C
                MF=(E,WSDYNEX)
        CLC LRETCODE,=AL4(CSVDYNEXRC_OK)
        BE CALLROUT        ONLY RC=00 ACCEPTED
        B ERRWTO

*
* CALL ACTUAL EXIT MODULE
*
CALLROUT DS OH
        LR R1,R3           Restore original parm reg
        MVC ERRMSGF,=CL6'CALL'

```

```

XC    LRSNCODE,LRSNCODE
XC    LRETCODE,LRETCODE
MVC   LRUBBITS,=X'40040000'    PASS R1 AND R13 TO EXIT MOD
L     R10,4(,R13)
ST    R13,LRUBR13              STORE OUR SAVE AREA FOR EXIT MOD
LA    R10,20(,R10)
MVC   LRUBR1(L'LRUBR1),4(R10)  STORE OLD R1 FOR EXIT
XC    LNEXTTOKEN,LNEXTTOKEN    INITIALIZE NEXT TOKEN

*
CSVDYNEX REQUEST=CALL,EXITNAME=LEX,FASTPATH=NO,          C
        NEXTTOKEN=LNEXTTOKEN,RUB=LRUB,RETINFO=LAST,      C
        RETAREA=LRETAREA,RETLEN=AL4(RETALEN),            C
        RETCODE=LRETCODE,RSNCODE=LRSNCODE,               C
        MF=(E,WSVDYNEX)
NC    LRSNCODE,=AL4(CSVDYNEXRSNCODEMASK) AND OFF EXTRA BITS
CLC   LRETCODE,=AL4(CSVDYNEXRC_OK)
BE    SETRC                      THE END, GET R15 FROM MODULE
CLC   LRETCODE,=AL4(CSVDYNEXRC_WARN)
BNE   ERRWTO
CLC   LRSNCODE,=AL4(CSVDYNEXRSNNOMODULES)
BE    RETURN00
B     ERRWTO

*
                                THIS EXIT IS JUST DUMMY
ERRWTO DS    0H
MVC   ERRMSGX,LEX
MVC   ERRMSG,=C' - '
LA    R2,LRETAREA
USING EXRET,R2
TM    EXRETFLAGS,EXRETABEND      DID MODULE ABEND?
BZ    ERRWTX
MVC   ERRMSGB(ERRMSGE-ERRMSGB),=CL20' ABEND      '
UNPK  ERRMSGB+7(9),EXRETABENDCODE(5)
TR    ERRMSGB+7(8),X2C-C'0'
MVI   ERRMSGB+7+8,C' '
B     ERRWTY
DROP  R2
ERRWTX DS    0H
MVC   ERRMSGB,=C' RC = '
UNPK  ERRMSGR(L'ERRMSGR+1),LRETCODE+3(2)
TR    ERRMSGR(L'ERRMSGR),X2C-C'0'
MVC   ERRMSGC,=C' RSN = '
UNPK  ERRMSGS(L'ERRMSGS+1),LRSNCODE+2(3)
TR    ERRMSGS(L'ERRMSGS),X2C-C'0'
ERRWTY DS    0H
MVC   ERRMSGT,=AL2(ERRMSGE-ERRMSGX)
MVC   ERRWTOL(ERRWTOL-ERRWTOL),EWTOL
LA    R2,ERRMSGT
WTO   TEXT=(2),MF=(E,ERRWTOL)
B     RETURN00
SETRC DS    0H
LA    R2,LRETAREA
USING EXRET,R2
TM    EXRETFLAGS,EXRETABEND      DID MODULE ABEND?
BNZ   ERRWTO
L     R3,EXRETCODE              INSERT RETURN CODE FROM EXIT MODULE
B     RETURN                    RETURN TO FREEMAIN, R3 CONTAINS RC
RETURN00 DS    0H
SLR   R3,R3                     RESET R3
RETURN DS    0H
LA    R0,WSLENGTH              GET LENGTH OF WS

```

```

LR      R1,R13                ADDRESS OF WS TO R1
L       R13,4(,R13)           GET ADDRESS OF PREVIOUS SAVE
DROP    R13                   TELL ASSEMBLER
FREEMAIN R,LV=(0),A=(1)       FREE WORKING STORAGE
LR      R15,R3                SET RETURN CODE
RETURN  (14,12)              RETURN TO CALLER - NO RC

*          Data Areas          *****
LEX      DC    CL16'ANFUXMSG'
X2C      DC    C'0123456789ABCDEF'
EWTOL    WTO    TEXT=,MF=L
EWTOLE   DS     F
          LTORG

*          Dsects              *****
WS        DSECT                WORKING STORAGE DSECT
DSSA      DS     18F           FIRST WORD OF SAVEAREA
DYNMODEL  CSVDYNEX MF=(L,WS)
WSDYNQRY  DS     CL512
RETALLEN  EQU    L'EXRETHDR+L'EXRETFIN
LRETAREA  DS     (RETALLEN)CL1
LNEXTTOKEN DS    D
LRETCODE  DS     F
LRSNCODE  DS     F
LRUB      DS     0XL12
LRUBBITS  DS     BL.32
LRUBR1    DS     A
LRUBR13   DS     A
ERRMSGT   DC     AL2(ERRMSG-ERRMSGX)
ERRMSGX   DC     CL16' '
ERRMSGGA  DC     C' - '
ERRMSGF   DC     CL6' '
ERRMSGB   DC     C' RC = '
ERRMSGR   DC     CL2' '
ERRMSGC   DC     C' RSN = '
ERRMSGS   DC     CL4' '
ERRMSGGE  DC     C' '
ERRWTOL   WTO    TEXT=,MF=L
ERRWTOLE  DS     F
          DS     0D
WSLENGTH  EQU    *-WS
DROP      R12
PRINT     NOGEN
ANFUXTTP
CSVEXRET
END  ANFUXMSG

```

UXMSGANF - IP PrintWay ANFUXMSG exit proper

Following is the code for the ANFUXMSG exit.

Example: A-2 UXMSGANF - IP PrintWay ANFUXMSG exit proper

```

UXMSGANF TITLE 'IP PRINTWAY ANFUXMSG DYNAMIC EXIT CODE'
UXMSGANF CSECT
UXMSGANF AMODE 31
UXMSGANF RMODE ANY
          YREGS

```

- *
- * PrintWay calls the same Message exit - ANFUXMSG - for all data sets.
- * The sample ANFUXMSG exit defines through the CSVDYNEX macro this

```

* UXMSGANF exit.
*
* The Message exit allows you to either (1) suppress or (2) modify
* the ID or text of any message that IP PrintWay issues to the
* PrintWay message-log data set, including messages created by
* another PrintWay exit.
*
* This sample exit does not change the input message at all. Rather,
* it reissues the messages through a single line or multiline WTO with
* ROUTCDE=(7), MCSFLAG=(HRDCPY),and DESC=(6). A single line WTO is
* used when the message text length does not exceed 125 bytes.
* When the message text length is greater than 125 bytes, a multiline
* WTO is used.
*
* At entry to the Message exit, the registers contain the following:
*
* Register 1   Pointer to the address of the ANFUEXTP control block.
* Register 13  Address of an 18-word save area for saving the caller's
*              registers.
* Register 14  Return address.
* Register 15  Entry-point address.
*
* Note: The Dynamic Exit Stub ANFUXMSG has turned XTPMSGFL.XTPM2MDS
* bit on (i.e. write messages to message data set)
*
* You can linkedit a dynamic exit routine into a library by:
*
* - Linkediting it into a data set that is made part of the PLPA,
*   MLPA, or FLPA at IPL-time
*
* - Linkediting it into a data set that is part of the LNKLST
*   concatenation
*
* - Linkediting it into IEANUCOx
*
* - Linkediting it into any PDS/PDSE and naming that data set using
*   the DSNAMES option of:
*
*   - The SETPROG EXIT command
*   - The EXIT ADD statement of a PROGxx parmlib member
*   - The ADD request of the CSVDYNEX macro.
*
* The PDS/PDSE data set must be cataloged, but does not need to be
* APF-authorized.
*
      SAVE (14,12),,UXMSGANF_&SYSDATE._&SYSTIME
      LR   R12,R15              Set base reg
      USING UXMSGANF,R12
      L    R2,0(,R1)           Get address of the ANFUEXTP
      USING ANFUEXTP,R2
      OI   XTPMSGFL,XTPM2MDS    Issue message to message data set
      LA   R0,PRIVATE-PRIVAT
      GETMAIN R,LV=(0)
      LR   R3,R1
      USING PRIVAT,R3
      XR   R4,R4
      ICM  R4,3,XTPMSGFLN       Length of the message
      BZ   RETURN
      L    R5,XTPMSGP           Address of the message
      CH   R4,=AL2(L'PRIVATC)   More than SL-WTO can handle

```

```

      BH    SPLITIT                Yes - Do ML-WTO
* ----- Issue Single line WTO
      MVC   PRIVATS,WTO$L
      STH   R4,PRIVATB
      BCTR  R4,0
      MVC   PRIVATC(*-*),0(R5)    Copy message
      EX    R4,*-6
      TR    PRIVATC(*-*),TRCHA    Eliminate non-printable chars
      EX    R4,*-6
      WTO   TEXT=PRIVATB,MF=(E,PRIVATS)
      B     DONEHERE
* ----- Issue Multiline WTO(s)
SPLITIT DS    0H
      MVC   PRIVATM,WTO$M
      SLR   R10,R10                Init connect id
      LR    R7,R5                  Copy message address
      SLR   R6,R6                  Set R6 to..
      BCTR  R6,0                   .-1
      LA    R9,70(,R7)             Point at E.O. max split size
SPLITMOR DS    0H
      CLI   0(R9),C' '             Good point to split message?
      BE    SPLITWTO
      BXH   R9,R6,SPLITMOR
SPLITWTO DS    0H
      LR    R1,R9                  Compute split..
      SR    R1,R7                  .text length
      BP    SPLITMVC
      LA    R9,70(,R7)             Point at E.O. max split
      B     SPLITWTO
SPLITMVC DS    0H
      STH   R1,PRIVATB
      BCTR  R1,0
      MVC   PRIVATC(*-*),0(R7)    Copy message (up to 70 bytes)
      EX    R1,*-6
      TR    PRIVATC(*-*),TRCHA    Eliminate non-printable chars
      EX    R1,*-6
      SH    R4,PRIVATB             Calculate remaining message length
      BP    SPLITNOE              Some left - Not MLWTO end line
      OI    PRIVATF,X'10'         Set "E" on top of "D"
SPLITNOE DS    0H
      MVC   PRIVATD,PRIVATB        Save original WTO length
      CLI   PRIVATC,C' '          Strip first blank from the WTO?
      BNE   SPLITNOB              No - WTO
      LH    R1,PRIVATB             Adjust WTO..
      BCTR  R1,R0                  .text..
      STH   R1,PRIVATB             ..length
      BCTR  R1,R0
      MVC   PRIVATC(*-*),PRIVATC+1 Strip
      EX    R1,*-6
SPLITNOB DS    0H
      WTO   TEXT=PRIVATB,CONNECT=(R10),MF=(E,PRIVATM)
      LR    R10,R1                 Save connect id
      MVC   PRIVATB,PRIVATD        Restore original WTO length
      LTR   R4,R4
      BNP   DONEHERE
      LH    R1,PRIVATB             Get split length
      AR    R7,R1                 Point at next message split
      LA    R9,70(,R7)             Point at E.O. max split
      CH    R4,=H'70'             Split size > max MLWTO line?
      BH    SPLITMOR              Yes - Keep splitting

```

```

        LA      R9,0(R4,R7)          Issue WTO for the..
        B       SPLITWTO             .last part of the split
        DROP    R2
DONEHERE DS      0H
        LR      R1,R3
        LA      R0,PRIVATE-PRIVAT
        FREEMAIN R,LV=(0),A=(1)
RETURN   DS      0H
        RETURN  (14,12)
*
        DROP    R3
        LTORG
WTOSL    WTO     TEXT=,ROUTCDE=(7),MCSFLAG=(HRDCPY),DESC=(6),MF=L
WTOSLE   DS      0H
WTOML    WTO     TEXT=((0,D)),ROUTCDE=(7),MCSFLAG=(HRDCPY),DESC=(6),MF=L
WTOMLE   DS      0H
*
TRCHA    DC      CL256' '
        ORG     TRCHA+X'4A'
        DC      X'4A4B4C4D4E4F50'
        ORG     TRCHA+X'5A'
        DC      X'5A5B5C5D5E5F6061'
        ORG     TRCHA+X'6A'
        DC      X'6A6B6C6D6E6F'
        ORG     TRCHA+X'79'
        DC      X'797A7B7C7D7E7F'
        ORG     TRCHA+C'a'
        DC      C'abcdefghi'
        ORG     TRCHA+C'j'
        DC      C'jklmnopqr'
        ORG     TRCHA+X'A1'
        DC      X'A1',C'stuvwxyz'
        ORG     TRCHA+X'CO'
        DC      X'CO',C'ABCDEFGH'I'
        ORG     TRCHA+X'DO'
        DC      X'DO',C'JKLMNOPQR'
        ORG     TRCHA+X'E0'
        DC      X'E0'
        ORG     TRCHA+C'S'
        DC      C'STUVWXYZ'
        ORG     TRCHA+C'O'
        DC      C'O123456789'
        ORG
*
PRIVAT   DSECT
PRIVATS  DS      XL(WTOSLE-WTOSL)
        ORG     PRIVATS
PRIVATM  DS      XL(WTOMLE-WTOML)
PRIVATF  EQU     *-4,2,C'X'
        ORG
PRIVATD  DS      H
PRIVATB  DS      H
PRIVATC  DS      CL125
PRIVATE  DS      OD
        ANFUEXT
        IKJTCB
        END    UXMSGANF

```



Sample IP PrintWay user exits

This appendix provides a sample job separator page exit that can be used to produce the same type of banner page produced by JES2 or JES3.

It contains the following:

- ▶ A sample ANFUXBD2 begin data set exit
- ▶ An example of the header or banner page produced
- ▶ A sample ANFUCRC2 record exit

Sample ANFUXBD2 begin data set exit

Example B-1 is a sample exit that creates a job separator page. It is a modification of sample exit ANFUXBD2 from SYS1.SAMPLIB.

The modifications are to extract the banner page block letter job name, programmer name, and room information from OUTPUT JCL when available. If the OUTPUT JCL information is not available, the XTP (exit parameter list) data is used instead. (This allows room information to be provided in a JES3 environment as well.)

The special support for the LPR Protocol 'Filename' and 'Owner' "/" (which was translated to # pound sign) is there as well.

Example: B-1 Sample exit for a job separator page

```
ANFUXBD2 TITLE 'Begin Data Set Exit'
***** START OF SPECIFICATIONS *****
* MODULE NAME = ANFUXBD2
* DESCRIPTIVE NAME = Begin Data Set exit
* COPYRIGHT= 5695040 (c) COPYRIGHT IBM CORPORATION 2000
* STATUS = VERSION 1, RELEASE 1, LEVEL 0
* FUNCTION = Begin Data Set exit for putting out banner page
* This exit produces:
* 1. A separator/banner with all the information that
* is also produced with a JES2 or PSF/MVS attached
* printer, see below.
* The output of this exits is as follows:
* start banner, which looks like:
*
* +-----+
* |
* | JOBNAME in big blockletters
* |
* |
* | PROGRAMMER NAME in big blockletters
* | (first 8 characters)
* |
* |
* | ROOM in big blockletters
* |
* |
* | ***START***START***START***START***START**
* | *
* | * JOBNAME: JOBID= USERID:
* | * SYSOUTCLASS: DESTINATION: PRINTER:
* | * PRINTTIME: PRINTDATE: SYSTEMID:
* | *
* | * OUTPUTGRP:
* | * {SEGMENTID:}
* | * TITLE:
* | * NAME:
* | * ROOM:
* | * BUILDING:
* | * DEPARTMENT:
* | * ADDRESS:
* | *
* | *
* | ***START***START***START***START***START**
* |
* | +-----+
* | +-----+
```



```

*          |      Application data          |      *
*          |                               |      *
*          /                               /      *
*          /                               /      *
*          |                               |      *
*          +-----+
* NOTES =
*      ANSI defined Printer Control Characters:
*      Blank = Space 1 line
*      0      = Space 2 line
*      -      = Space 3 line
*      +      = Suppress space
*      1      = Skip to line 1 on new page
* DEPENDENCIES = none
* MODULE TYPE = PROCEDURE
*   PROCESSOR = Assembler
*   ATTRIBUTES =
*       REENTRANT
*       AMODE(31)
*       RMODE(ANY)
* PATCH LABEL = none
* ENTRY POINT =
*       ANFUXBD2
* LINKAGE =
*       LOAD ANFUXBD2
*       CALL ANFUXBD2(addr(ANFUExTP))
*       calling module passes address of Common Parameter
*       area.
* INITIAL VERSION: July 2000
* CHANGE ACTIVITY =
***** END OF SPECIFICATIONS *****
ANFUXBD2 CSECT ,                Establish the csect
ANFUXBD2 AMODE ANY              It addresses all storage
ANFUXBD2 RMODE ANY              It can reside anywhere
                                Establish temporary          X
                                addressability to module
                                B      BDSINIT                Branch around copyright
ANFUXBD2 MODID BR=NO            information eyecatcher info
                                DC    C' This module has been developed '
                                DC    C'by Jan de Rover, IBM Printing '
                                DC    C'Systems Division (PSD) Amsterdam '
                                DC    C'in the Netherlands '
                                DC    C'This code is not considered Type 1 '
                                DC    C'code of IBM. Also it has not been '
                                DC    C'submitted to any formal IBM test. '
                                DC    C'Distribution is on an "as is" '
                                DC    C'basis without any warranty either '
                                DC    C'expressed or implied. '
                                PRINT ON,GEN
*****
*      ANFUXBD2 Module entry point
*****
BDSINIT DS    OH
        STM    R14,R12,12(R13)    Save callers registers
        LR     R12,R15            Move base register to R12
        DROP   R15                Done with this register
        USING  ANFUXBD2,R12        R12 is base reg for program code
        L      R4,0(R1)           Get address of parm area
        USING  ANFUExTP,R4         R4 is base for parm area
        ST     R13,XTPSAVE+4      Chain save areas

```

```

        LA    R15,XTPSAVE          Get my save area
        ST    R15,8(R13)          Point callers save area to it
        LR    R13,R15            Point R13 to my save area
*****
*      Get storage for work area      *
*****
        SLR   R2,R2              Get a zero
        L     R9,XTPWORK1        Get working storage addr
        CR    R9,R2              Has work area been gotten?
        BNE   BDSDOIT            Yes, skip this
        LA    R2,BUFWRKL         Get length of DSECT
        GETMAIN RU,LV=(2),LOC=ANY Get program storage area
BDSGOTIT DS    OH
        ST    R1,XTPWORK1        Save my work area
        LR    R9,R1              Put it in the correct reg
*****
*      Actual work - call to get block characters      *
*****
BDSDOIT DS    OH
        USING BUFWRK,R9
        USING IAZJSPA,R11
        L     R11,XTPJSPAP        Get address of jspa area
        TM    XTPDSFLG,XTPDSJOB   First data set in job?
        BNO   GETOUT              No, don't produce banner
        TM    XTPRCFLG,XTPRDFST   First call for data set?
        BNO   ENDINIT             No, don't do init stuff
*****
*      First call only routine      *
*****
        MVI   BDSWORKD,0          State is need first userid line
        MVI   BDSWORKE,0          SWB error flag
        BAL   R10,SJFRETR         Read SWBs
*-----*
*      First blockletter = JOBNAME      *
*-----*
        MVC   SWBUIDN,=CL16' '    Init UID save field
        MVC   SWBJBNM,=CL16' '    Init job name save field
        CLC   JSPAJBNM(2),=CL2'# ' Resubmit special translation?
        BNE   HAVEJNM             BIN - No problem
*      Extract UID and JOBNAME from TITLE (JES node.uid.jobname...)
        USING SJTRKEYL,R2         Base KEYLIST
        ICM   R2,B'1111',WRKKYLSL Load KEYLIST address
        BZ    HAVEJNM             Branch if NO KEYLIST
        LA    R2,KYLSTTL(,R2)     Addr of TITLE key
        ICM   R1,B'1111',SJTRTPAD Load TITLE TU address
        BZ    HAVEJNM             Branch if NO TITLE
        LA    R1,DOCNTENT-DOCNUNIT(,R1) Addr. of TEXT UNIT
        USING DOCNTFLD,R1
        LA    R14,DOCNTPRM        Load TU text address
        LH    R15,DOCNTLEN        Length of TU text
        DROP  R1,R2
        LA    R0,10               Find "." loop count
FINDJNM DS    OH                  Skip 'node.'
        CLI   0(R14),C' '         Anything?
        BE    HAVEJNM             BIN
        CLI   0(R14),C'.'         Period?
        BE    FINDUIDS            BIY
        LA    R14,1(,R14)         Point at next char
        BCT   R0,FINDJNM          Keep finding a "."
        B     HAVEJNM             No "."

```

FINDUIDS	DS	OH	Take 'uid'
	LA	RO,8	Find "." loop count
	LA	R15,SWBUIDN	Point at UID save field
FINDUID	DS	OH	
	CLI	1(R14),C'.'	Period?
	BE	FINDJNMS	BIY - Done with UID
	MVC	O(1,R15),1(R14)	Copy jobname char
	LA	R14,1(,R14)	Point at next char in
	LA	R15,1(,R15)	Point at next char out
	BCT	RO,FINDUID	Keep copying
	MVC	SWBUIDN,=CL16' '	Reset UID save field
	B	HAVEJNM	Did 8 chars
FINDJNMS	DS	OH	Take 'jobname'
	LA	RO,9	Find "." loop count
	LA	R15,SWBJBNM	Point at job name field
FINDJNML	DS	OH	
	CLI	2(R14),C'.'	Period?
	BE	HAVEJNMM	BIY - Done with JOBNAME
	MVC	O(1,R15),2(R14)	Copy jobname char
	LA	R14,1(,R14)	Point at next char in
	LA	R15,1(,R15)	Point at next char out
	BCT	RO,FINDJNML	Keep copying
	MVC	SWBJBNM,=CL16' '	Reset job name save field
HAVEJNMM	DS	OH	
	MVC	XTPBSTRG,SWBJBNM	Copy job name for conversion
	B	HAVEJNMD	Done with JOBNAME
HAVEJNM	DS	OH	
	MVC	XTPBSTRG,JSPAJBNM	Copy job name for conversion
HAVEJNMD	DS	OH	
	MVI	XTPBIFLG,X'00'	Default formatting stuff
	OI	XTPBIFLG,XTPBPRFM	Narrow letters
	OI	XTPBIFLG,XTPBLJST	On left margin
	OI	XTPBIFLG,XTPBFRST	First call for the string

*	End first call only routine		*

*	Build the blockletters for the banner		*

ENDINIT	DS	OH	
	CLI	BDSWORKD,1	Done with last line of 1e block
	BH	DOSECOND	Yes, do 2nd blockletter
*	Call the block letter routine		
	ST	R4,WRKSAVE	Save addr of parm
	LA	R1,WRKSAVE	
	L	R15,=V(ANFUBLK)	Get address of block builder
	BALR	R14,R15	
	CLI	BDSWORKD,0	First line first block?
	BH	DOREMDR	No, do remainder of block
*	MVI	XTPBOUTP,X'F1'	Page eject carriage control
	MVI	XTPBOUTP,C' '	Space 1 line carriage control
	MVI	BDSWORKD,1	Finished first 1e blk line
DOREMDR	DS	OH	
	TM	XTPBOFLG,XTPBLAST	Last one of this block?
	BNO	PRINTIT	No, don't do next blk init
	MVI	BDSWORKD,2	Saw last rec,
*			prepare for 2nd blktr

*	Second blockletter = PROGRAMMER NAME (first 8 pos)		*

	USING SJTRKEYL,R2	Base KEYLIST
	ICM R2,B'1111',WRKKYLIST	Load KEYLIST Address
	BZ DONONAME	Branch if no KEYLIST
	LA R2,KYLSTNM(,R2)	Locate Addr. of NAME
*		key in the KEYLIST
	ICM R1,B'1111',SJTRTPAD	Load NAME TU address
	BZ DONONAME	Branch if NO NAME
	LA R1,DOCNTENT-DOCNUNIT(,R1)	Addr of text unit
	USING DOCNTFLD,R1	Copy..
	LA R14,DOCNTPRM	.NAME..
	LA R15,8	..from..
	CH R15,DOCNTLEN	...OUTPUT..
	BNH *+4+4JCL..
	LH R15,DOCNTLENwhen..
	SH R15,=H'1'available
	BM DONONAME	
	MVC XTPBSTRG+(0),0(R14)	Copy for conversion
	EX R15,*-6	
	B DOSECBLK	Branch to PRT_NAME
	DROP R2,R1	Drop KEYLIST
DONONAME	DS OH	If name not in the JCL
	MVC XTPBSTRG(8),SWBUIDN	Put in requestor user ID
	CLI SWBUIDN,C' '	Have a name from OUTPUT JCL?
	BNE DOSECBLK	BIY
	MVC XTPBSTRG,JSPJPNAM	Conversion 2nd blk
DOSECBLK	DS OH	

	OI XTPBIFLG,XTPBFRST	First call for the string
	B PRINTIT	

*	Prepare to print the second block letter	*

DOSECOND	DS OH	
	CLI BDSWORKD,3	Done with last line of 2nd block
	BH DOTHIRD	Yes, do 3e block
*	Call the block letter routine	
	ST R4,WRKSAVE	Save addr of parm
	LA R1,WRKSAVE	
	L R15,=V(ANFUBLK)	Get address of block builder
	BALR R14,R15	
	CLI BDSWORKD,2	First line second block?
	BH DOREMDR2	No, do remainder of second blk
	MVI XTPBOUTP,X'F0'	Skip line carriage control
	MVI BDSWORKD,3	Finished first line second blk
DOREMDR2	DS OH	
	TM XTPBOFLG,XTPBLAST	Last one of this block?
	BNO PRINTIT	No, don't do next line init
	MVI BDSWORKD,4	Finished second block

*	Third blockletter = ROOM from jobstm	*

	MVC XTPBSTRG(8),=CL16' '	2 leading blanks
	USING SJTRKEYL,R2	Base KEYLIST
	ICM R2,B'1111',WRKKYLIST	Load KEYLIST address
	BZ DONOROOM	Branch if no KEYLIST
	LA R2,KYLSTRM(,R2)	Locate ROOM addr in KEYLIST
	ICM R1,B'1111',SJTRTPAD	Load ROOM TU Address
	BZ DONOROOM	Branch If no ROOM
	LA R1,DOCNTENT-DOCNUNIT(,R1)	Addr of TEXT UNIT
	USING DOCNTFLD,R1	Copy..

```

        LA    R14,DOCNTPRM      .ROOM..
        LA    R15,8              ..from..
        CH    R15,DOCNTLEN      ...OUTPUT..
        BNH   *+4+4             ....JCL..
        LH    R15,DOCNTLEN      ....when..
        SH    R15,=H'1'         .....available
        BM    DONOROOM
        MVC   XTPBSTRG(0),0(R14) Copy for conversion
        EX    R15,*-6
        CLI   XTPBSTRG,C' '      Anything from OUTPUT JCL?
        BNE   DOROOMOK          Have a ROOM
        DROP  R2,R1              Drop KEYLIST
DONOROOM DS    OH               If room not in the JCL
        MVC   XTPBSTRG+2(4),JSPJRMNO Copy for conversion
DOROOMOK DS    OH               If room not in the JCL
*-----*
SETUNKWN DS    OH
        OI    XTPBIFLG,XTPBFRST First call for the string
        B     PRINTIT           Write last line sec block
*****
*       Prepare to print the third block letter       *
*****
DOTHIRD DS    OH
        CLI   BDSWORKD,5        Done with last line of 3e block
        BH    DETAIL0           Yes, do next
*       Call the block letter routine
        ST    R4,WRKSAVE        Save addr of parm
        LA    R1,WRKSAVE
        L     R15,=V(ANFUBLK)   Get address of block builder
        BALR  R14,R15
        CLI   BDSWORKD,4        First line second block?
        BH    DOREMDR3          No, do remainder of third blk
        MVI   XTPBOUTP,X'F0'    Skip line carriage control
        MVI   BDSWORKD,5        Finished first line third blk
DOREMDR3 DS    OH
        TM    XTPBOFLG,XTPBLAST Last one?
        BNO   PRINTIT           No, don't do next line init
        MVI   BDSWORKD,6        Finished third block
        B     PRINTIT           Write last line third block
*****
*       Prepare to print detail records                *
*       The branch tabel below MUST be keep in the same order *
*       to prevent formating errors                    *
*       Every routine at the designated label will set the *
*       BDSWORKD switch. That switch will be used when this *
*       module is called again.                        *
*****
DETAIL0 SLR    R3,R3             Get a zero
        ICM   R3,1,BDSWORKD     Get the footprint
        SLL   R3,2              Multiply with 4 for B-tabel
        B     BTABEL(R3)        Go to the correct rtn
BTABEL  DS    OF
        DC    6F'0'             Routines already done
        B     DODETAIL          BDSWORKD=6
        B     DOBOXL1           BDSWORKD=7
        B     DOBOXL2           BDSWORKD=8
        B     DOBOXL3           BDSWORKD=9
        B     DOBOXL4           BDSWORKD=10
        B     DOBOXL5           BDSWORKD=11
        B     DOBLANK1          BDSWORKD=12

```

```

B      DOBOXL6      BDSWORKD=13
B      DOBSEGM      BDSWORKD=14
B      DOBOXL7      BDSWORKD=15
B      DOBOXL8      BDSWORKD=16
B      DOBOXL9      BDSWORKD=17
B      DOBOXLA      BDSWORKD=18
B      DOBOXLB      BDSWORKD=19
B      DOBOXLC      BDSWORKD=20
B      DOBOXLD      BDSWORKD=21
B      NOADDR       BDSWORKD=22
B      DOBOXLF      BDSWORKD=23
B      SKIPSPAC     BDSWORKD=24
B      SKIPSPAC     BDSWORKD=25
B      SPACE1L      BDSWORKD=26
B      DOLAST       BDSWORKD=27
*****
*      End of Branch tabel      *
*****
*****
*      First space 1 line      *
*****
DODETAIL DS      OH
        MVI      WRKCC,C' '      Space 1 line
        LA      R5,1
        ST      R5,XTPERLEN
        ST      R9,XTPERPTR
        OI      XTPRCFLG,XTPRCTRN      Translate it to ASCII
        OI      XTPRCFLG,XTPRCCC      Assume it has carriage control
        OI      XTPRCFLG,XTPRCEXT      Print the record
        MVI      BDSWORKD,7
        B      GETOUT
*****
*      Obtain time and date (year 2000 ready)      *
*      Routine called if BDSWORKD=7      *
*****
DOBOXL1 DS      OH
        XC      WRKY2000(16),WRKY2000      Clear
        TIME    DEC,WRKY2000,LINKAGE=SYSTEM,MF=(E,TIME1)
        L      R0,WRKY2000+8      Point to date OYYYYDDD
        SLL     R0,4      RO=YYYYDDDDO
        SLR     R1,R1      R1=00000000
        IC      R1,=X'0F'      R1=0000000F
        OR      R0,R1      RO=YYYYDDDF
        ST      R0,WRKDATE      Save date packed decimal
        L      R0,WRKY2000      Point to time HHMMSSST
*****
*      Adjust time for am/pm      *
*****
        LA      R2,WRKTIME      Get address of work area
        LA      R1,WRKAMPM      Get address of AM/PM workarea
        MVC     WRKAMPM(2),AM      Set AM/PM to AM
        CL      R0,=X'12000000'      Test for zero hours
        BL      PMORNING      Branch if AM
        MVI     0(R1),C'P'      Change from AM to PM
        SL      R0,=X'12000000'      Subtract twelve hours
PMORNING ST      R0,0(,R2)      Store adjusted time
        CLI     0(R2),X'00'      Test for zero hours
        BNE     PADJERR      Br if not to test adj err
        MVI     0(R2),X'12'      Convert zero to twelve
PADJERR  TM      0(R2),X'08'      Test for adjustment errors

```

```

        BZ    PEDTIME                Branch if no error
        NI    0(R2),X'09'            Correct for binary substr. err
PEDTIME DS    OH
*****
*          Unpack hours minutes seconds          *
*****
        MVI   WRKTH,X'0C'            Reset low order 2 bytes
*                                     with sign for packed decimal
        UNPK  WRKUTIME(7),WRKTIME(4) Unpack time
        CLI   WRKHR,X'F0'            If hour has leading zero
        BNE   NEXTJ
        MVC   WRKHR(1),BLANK         Change zero to blank
*****
*          Obtain month, day and year            *
*****
NEXTJ   LA    R1,JULTABEL            Addressability to Julian table
        MVC   WRKJTBL(48),DAYTBL(R1) Copy table for leap year
*                                     adjustment
        MVC   WRKWORK+4(4),WRKDATE   Obtain date from saved area
        TM    WRKWORK+5,X'01'         test
        BO    NOLEAPYR               for
        TM    WRKWORK+5,X'12'         leap
        BM    NOLEAPYR               year
        MVI   WRKJTBL+4,29            Adjust feb for leap year
NOLEAPYR MVC   WRKYY(4),=X'20202020' Place pattern for edit
        ED    WRKYY(4),WRKWORK+4     Edit the year
        MVC   WRKWORK(6),ZEROES      Reset all but julian date
        SLR   R0,R0                  Clear for ic
        CVB   1,WRKWORK              Convert to binary day
        LA    2,WRKJTBL-4            Address od date conversion
*                                     table
SEARCH  SLR   R1,R0                  convert
        LA    R2,4(,R2)              julian day
        IC    R0,0(,R2)              to
        CLR   R0,R1                 standard day
        BL    SEARCH
        CVD   1,WRKWORK              Convert to decimal day
        UNPK  WRKDD(2),WRKWORK+6(2)  Unpack the day
        OI    WRKDD+1,X'F0'          Insure sign nibble
        MVC   WRKMMM(3),1(R2)        Set ebcidic alpha month
*****
*          Start buiding the Box now:            *
*          Build and print top line              *
*****
TOPLINE DS    OH
        BAL   R14,CLRLINE            Clear printline
        MVC   BOXLINE(10),STARTAG     Move in START tag
        MVC   BOXLINE+10(L'BOXLINE-10),BOXLINE Propogate chars
        MVI   BDSWORKD,8
        B     GETOUT
*****
*          Build and print 1 blank line          *
*          (Frame characters in box cols 01 & 78) *
*          Routine called if BDSWORKD=8          *
*****
DOBOXL2 DS    OH
        BAL   R14,CLRLINE            Clear printline
        MVI   BDSWORKD,9
        B     GETOUT
*****

```

```

*          Build and print JOB NAME + JOB ID + USERID line          *
*          Routine called if BDSWORKD=9                             *
*****
DOBOXL3  DS      OH
          BAL      R14,CLRLINE          Clear printline
          MVC      BOXDESC,LBJOBNAM     Put JOB NAME label
          MVC      BOXINFO(8),JSPAJBNM  Put JOBNAME into box
          CLC      JSPAJBNM(2),=CL2'# ' Resubmit special translation?
          BNE      DOBOXL3T
          CLI      SWBJBNM,C' '         Have a spare name?
          BE       DOBOXL3T             BIN
          MVC      BOXINFO(8),SWBJBNM   Put JOBNAME into box
DOBOXL3T DS      OH
          MVC      BOXDESC2,LBJOBID     Put JOB ID label
          MVC      BOXINFO2(8),JSPAJBID Put JOBID into box
          MVC      BOXDESC3,LBUSERID    Put USER NAME label
          MVC      BOXINFO3(8),XTPUSRID  Put USER NAME
          CLC      XTPUSRID,=C'+++++++' Unprintable id from system?
          BNE      DOBOXL3A
          MVC      BOXINFO3(8),=C'UNKNOWN ' Put in unknown string
DOBOXL3A DS      OH
          CLC      BOXINFO3(2),=CL2'# ' Resubmit translation special?
          BNE      DOBOXL3B
          CLI      SWBUIDN,C' '         Have a spare name?
          BE       DOBOXL3B             BIN
          MVC      BOXINFO3(8),SWBUIDN   Put in requestor user ID
DOBOXL3B DS      OH
          MVI      BDSWORKD,10
          B        GETOUT
*****
*          Build and print SYSOUT CLASS + DEST + PRT line          *
*          Routine called if BDSWORKD=10                           *
*****
DOBOXL4  DS      OH
          BAL      R14,CLRLINE          Clear printline
          MVC      BOXDESC,LBSYSCL      Put SYSOUT CLASS label
          MVC      BOXINFO(1),XTPCLASS  Put SYSOUT CLASS
          OC       BOXINFO(16),=C116' ' Upper case
          MVC      BOXDESC2,LBDEST      Put DESTINATION label
          MVC      BOXINFO2(8),XTPDEST  Put DESTINATION
          OC       BOXINFO2,=C116' '    Upper case
          MVC      BOXDESC3,LBPRNAME    Put PRINTER NAME label
          MVC      BOXINFO3(8),JSPADEVN Put device NAME
          OC       BOXINFO3,=C116' '    Upper case
          MVI      BDSWORKD,11
          B        GETOUT
*****
*          Build and print TIME + DATE + SYSTEMID (of printer)     *
*          Routine called if BDSWORKD=11                           *
*****
DOBOXL5  DS      OH
          BAL      R14,CLRLINE          Clear printline
          MVC      BOXDESC,LBPRTIME     Put PRINT TIME label
          CLC      WRKHR(1),BLANK        If hour has leading blank
          BNE      HROKAY               NO - then branch
          MVI      WRKHR,X'F0'          Change blank to zero
HROKAY   MVC      BOXHR(2),WRKHR        Set current hour
          MVI      BOXTS1,X'7A'         Set time separator
          MVC      BOXMIN(2),WRKMIN     Set current minute
          MVI      BOXTS2,X'7A'         Set time separator

```



```

MVC BOXSEC(2),WRKSEC          Set current second
MVC BOXAMP(2),WRKAMP          Set current AM/PM
MVC BOXDESC2,LBPRDATE        Put PRINT DATE label
MVC BOXDDD(2),WRKDD          Set print DAY
MVC BOXMMM(3),WRKMMM          Set print MONTH
MVC BOXYYY(4),WRKYY          Set print YEAR
USING JMR,R7                  Base JMR
L R7,JSPAJMR                  Load address of JMR
MVC BOXDESC3,LBSYSTEM        Put SYSTEM label
MVC BOXINFO3(4),JMRCPUID      Put SYSTEM ID
MVI BDSWORKD,12
B GETOUT
DROP R7                        Drop Address. to JMR
*****
* Build and print 1 blank line *
* (Frame characters in box cols 01 & 78) *
* Routine called if BDSWORKD=12 *
*****
DOBLANK1 DS OH
BAL R14,CLRLINE              Clear printline
MVI BDSWORKD,13
B GETOUT
*****
* Build and print OUTPUT GROUP line *
* Routine called if BDSWORKD=13 *
*****
DOBOXL6 DS OH
BAL R14,CLRLINE              Clear printline
MVC BOXDESC,LBOUTGRP          Put OUTPUT GROUP label
MVC BOXJNAME(8),JSPJGRP1      Put OUTPUT GRP NODE1
MVI BOXGSEP1,C'.'            Insert separator char
LH R0,JSPJGRP1                Get JOE ID1
CVD R0,WRKJIDEC               Convert to pkd decimal
UNPK BOXJID1,WRKJIDEC+5(3)    Unpack NODE2
OI BOXJZON1,X'F0'             Insure positive sign
MVI BOXGSEP2,C'.'            Insert separator char
LH R0,JSPJGRP2                Get JOE ID2
CVD R0,WRKJIDEC               Convert to pkd decimal
UNPK BOXJID2,WRKJIDEC+5(3)    Unpack GRP NODE3
OI BOXJZON2,X'F0'             Insure positive sign
MVI BDSWORKD,14
B GETOUT
*****
* Build and print SEGMENT ID line, if present, else blank line *
* Called if BDSWORKD=14 *
*****
DOBSEGM DS OH
BAL R14,CLRLINE              Clear printline
USING JSPEXT,R7              Base JSPA extension
TM JSPAFLG1,JSPA1EXT          Does JSPA ext. exist?
BZ PROUTPUT                   NO--Branch
LR R7,R11                     Load JSPA address
AH R7,JSPALEN                 Add JSPA length
CLC JSPCESEG,XZERO            Is SYSOUT segmented?
BE PROUTPUT                   NO---Omit SEGMENT ID
MVC BOXSGLBL(13),LBSEGID      Put SEGMENT ID label
L R14,JSPCESEG                 Load R14 SEGMENT#
CVD R14,DBLWORD                Convert to pkd dec.
CP DBLWORD,PZERO              Is SEGMENT# < 0?
BL INVSEG                     YES--put invalid SEG#

```

```

        CP      DBLWORD,MAXSEG#           Is SEGMENT# > 99999?
        BNH     PRSEG#                     NO---put SEGMENT#
INVSEG   MVC     BOXSGINF(5),=C'*****'   Set invalid SEGMENT#
        B       PROUTPUT                   GO print this line
PRSEG#   DS      OH
        UNPK    BOXSGINF,DBLWORD+5(3)     Put SEGMENT# into box
        OI      BOXSGINF+4,X'F0'          Insure positive sign
PROUTPUT DS      OH
        MVI     BDSWORKD,15
        B       GETOUT
*****
*        Build and print TITLE line                      *
*        Routine called if BDSWORKD=15                    *
*****
DOBOXL7  DS      OH
        BAL     R14,CLRLINE               Clear printline
        MVC     BOXDESC,LBTITLE           Put TITLE label
        USING   SJTRKEYL,R2              Base KEYLIST
        ICM     R2,B'1111',WRKKYLIST      Load KEYLIST address
        BZ      PRTITLE                   Branch if NO KEYLIST
        LA      R2,KYLSTTL(,R2)           Locate addr. of TITLE
*                                                key in the KEYLIST
        ICM     R1,B'1111',SJTRTPAD       Load TITLE TU address
        BZ      PRTITLE                   Branch if NO TITLE
        LA      R1,DOCNTENT-DOCNUNIT(,R1) Addr. of TEXT UNIT
        BAL     R8,MOVETU                 Get TITLE data
        DROP    R2                        Drop KEYLIST
*                                                from the TU
PRTITLE  DS      OH
        MVI     BDSWORKD,16
        B       GETOUT
*****
*        Build and print NAME line                      *
*        Routine called if BDSWORKD=16                    *
*****
DOBOXL8  DS      OH
        BAL     R14,CLRLINE               Clear printline
        MVC     BOXDESC,LBNAME           Put NAME label
        USING   SJTRKEYL,R2              Base KEYLIST
        ICM     R2,B'1111',WRKKYLIST      Load KEYLIST Address
        BZ      NONAME                    Branch if no KEYLIST
        LA      R2,KYLSTNM(,R2)           Locate Addr. of NAME
*                                                key in the KEYLIST
        ICM     R1,B'1111',SJTRTPAD       Load NAME TU address
        BZ      NONAME                    Branch if NO NAME
        LA      R1,DOCNTENT-DOCNUNIT(,R1) Addr of text unit
        BAL     R8,MOVETU                 Put NAME data from TU
        B       PRT_NAME                  Branch to PRT_NAME
        DROP    R2                        Drop KEYLIST
NONAME   DS      OH
*                                                If name not in the JCL
*                                                then use the name in
*                                                the JSPA
        MVC     BOXINFO(L'JSPJPNAM),JSPJPNAM Set from JOB stmt
PRT_NAME DS      OH
        MVI     BDSWORKD,17
        B       GETOUT
*****
*        Build and print ROOM line                      *
*        Routine called if BDSWORKD=17                    *
*****

```

```

DOBOXL9 DS    OH
        BAL    R14,CLRLINE          Clear printline
        MVC    BOXDESC,LBROOM        Put ROOM label
        USING  SJTRKEYL,R2           Base KEYLIST
        ICM    R2,B'1111',WRKKYLIST Load KEYLIST address
        BZ     NOROOM                Branch if no KEYLIST
        LA     R2,KYLSTRM(,R2)       Locate Addr of ROOM
*                                     Key in the KEYLIST
        ICM    R1,B'1111',SJTRTPAD   Load ROOM TU Address
        BZ     NOROOM                Branch If no ROOM
        LA     R1,DOCNTENT-DOCNUNIT(,R1) Addr of TEXT UNIT
        BAL    R8,MOVETU             Put ROOM data from TU
        B      PRT_ROOM              Branch to PRTRROOM
        DROP   R2                    Drop KEYLIST
NOROOM  DS    OH                    If room not in the JCL
*                                     then use the room in
*                                     the JSPA
        MVC    BOXINFO(L'JSPJRMNO),JSPJRMNO Set from JOB stmt
PRT_ROOM DS    OH
        MVI    BDSWORKD,18
        B      GETOUT
*****
*      Build and print BUILDING line                                     *
*      Routine called if BDSWORKD=18                                     *
*****
DOBOXLA DS    OH
        BAL    R14,CLRLINE          Clear printline
        MVC    BOXDESC,LBBLDG        Put BLDG label
        USING  SJTRKEYL,R2           Base KEYLIST
        ICM    R2,B'1111',WRKKYLIST Load KEYLIST address
        BZ     PRTBLDG              Branch if no KEYLIST
        LA     R2,KYLSTBL(,R2)       Locate addr of BLDG
        ICM    R1,B'1111',SJTRTPAD   Load BLDG TU Address
        BZ     PRTBLDG              Branch if no BUILDING
        LA     R1,DOCNTENT-DOCNUNIT(,R1) Addr of TEXT UNIT
        BAL    R8,MOVETU             Put BLDG data from TU
        DROP   R2                    Drop KEYLIST
PRTBLDG DS    OH
        MVI    BDSWORKD,19
        B      GETOUT
*****
*      Build and print DEPARTMENT line                                   *
*      Routine called if BDSWORKD=19                                   *
*****
DOBOXLB DS    OH
        BAL    R14,CLRLINE          Clear printline
        MVC    BOXDESC,LBDEPT        Put DEPT label
        USING  SJTRKEYL,R2           Base KEYLIST
        ICM    R2,B'1111',WRKKYLIST Load KEYLIST address
        BZ     PRTDEPT              Branch if no KEYLIST
        LA     R2,KYLSTDP(,R2)       Locate addr of DEPT
        ICM    R1,B'1111',SJTRTPAD   Load DEPT TU address
        BZ     PRTDEPT              Branch if no DEPT
        LA     R1,DOCNTENT-DOCNUNIT(,R1) Addr of TEXT UNIT
        BAL    R8,MOVETU             Put DEPT data from TU
        DROP   R2                    Drop KEYLIST
PRTDEPT DS    OH
        MVI    BDSWORKD,20
        B      GETOUT
*****

```

```

*                BUILD AND PRINT ADDRESS LINES                *
*                -----                *
*    The address can be from 1 to 4 lines long. Any lines    *
*    that are unused must be printed as blank lines, but the *
*    label "ADDRESS:" must appear on the first line even if   *
*    no address was specified on the OUTPUT JCL.              *
*    A blank line is printed after the four ADDRESS lines.    *
*    If the SWBTUREQ macro returned an error, the ADDRESS     *
*    line is not printed. An error message is printed         *
*    in the four lines that would have contained the ADDRESS. *
*    Routine called if BDSWORKD=20                             *
*****
DOBOXLC DS    OH
        CLI    SWBERR,C'Y'          SWBTUREQ error?
        BNE    ADDRLBL             NO - GO print
*****
*    Print the SWBTUREQ error msg                               *
*****
        BAL    R14,CLRLINE          Clear printline
        CLI    BDSWORKE,0
        BH     ADDLINE2
        MVC    BOXMSG1(40),LBMSG     Set error msg text
        MVI    BDSWORKE,1
        B      GETOUT
ADDLINE2 DS    OH
        CLI    BDSWORKE,1
        BH     ADDLINE3
        MVC    BOXRCLB,LBRC          Set return code label
        MVC    BOXRC(4),SWBRC        Set SWBTUREQ
        MVI    BDSWORKE,2
        B      GETOUT
ADDLINE3 DS    OH
        CLI    BDSWORKE,2
        BH     ADDLINE4
        MVC    BOXRSLB,LBRS          Set REASON CODE label
        MVC    BOXRS(4),SWBRS        Set SWBTUREQ reason code
        MVI    BDSWORKE,3
        B      GETOUT
ADDLINE4 DS    OH
        CLI    BDSWORKE,3
        BH     BOTBLNKS
        MVI    BDSWORKE,4
        B      GETOUT
BOTBLNKS DS    OH
        CLI    BDSWORKE,4
        BH     SKIPSPAC
        MVI    BDSWORKE,5
        MVI    BDSWORKD,25
        B      GETOUT              Printit
*****
*    Print the ADDRESS label                                    *
*****
ADDRLBL DS    OH
        MVC    WRKADNUM(2),XZERO     Zero number of lines
        BAL    R14,CLRLINE          Clear printline
        MVC    BOXDESC,LBADDR        Put ADDRESS label
        USING  SJTRKEYL,R2           Base KEYLIST
        ICM    R2,B'1111',WRKKYLIST Load KEYLIST address
        BZ     NOADDR                Branch if no KEYLIST
        LA     R2,KYLSTAD(,R2)       Locate addr of ADDR

```

```

*                                Key in the KEYLIST
      ICM  R1,B'1111',SJTRTPAD    Load ADDR TU address
      BZ   NOADDR                 Branch if no ADDRESS
      LH   R7,DOCNTNUM-DOCNUNIT(,R1) Number of lines used
*                                for ADDRESS info
      CH   R7,=H'4'               Is number > 4 ?
      BNH  SAVENUML              NO---Save number of
*                                lines
      LA   R7,4                   YES---Set to MAX of 4
SAVENUML DS   OH
      STH  R7,WRKADNUM           Save number of lines
      LTR  R7,R7                 Number > zero?
      BZ   NOADDR               NO---print blank lines
      DROP R2                   Drop KEYLIST
*****
*      LOOP to Build/Print Multiple Line ADDRESS      *
*****
      LA   R1,DOCNTENT-DOCNUNIT(,R1) Addr of TEXT UNIT
      USING DOCNTFLD,R1
ADDRLOOP DS   OH
      LR   R6,R1                 Save the TU address
      BAL  R8,MOVETU             Put ADDR data from TU
      ST   R6,WRKSAVR6
      ST   R7,WRKSAVR7
      MVI  BDSWORKD,21
      B    GETOUT               Printit
*****
*      Routine called if BDSWORKD=21      *
*****
DOBOXLD DS   OH
      L    R6,WRKSAVR6
      L    R7,WRKSAVR7
      LR   R1,R6                 Restore TU address
      LH   R15,DOCNTLEN          Bump down to next
      LA   R1,L'DOCNTLEN(R15,R1) Address length/data pair
      BAL  R14,CLRLINE           Clear printline
      BCT  R7,ADDRLOOP           Loop again if more
*                                data
      DROP R1                   Drop addressability
      MVI  BDSWORKD,22
*****
*      LOOP to Build/Print Multiple Blank ADDRESS Lines      *
*      Routine called if BDSWORKD=22      *
*****
NOADDR  DS   OH
      LA   R6,4                 Load MAX # of lines
      LH   R7,WRKADNUM          Restore # of address
*                                lines used
      SR   R6,R7                Find number unused
*                                lines
      LA   R6,1(,R6)            Add 1 for extra blank line
BLKLINE DS   OH
      ST   R6,WRKSAVR6
      MVI  BDSWORKD,23
      B    GETOUT               Printit
*****
*      Routine called if BDSWORKD=23      *
*****
DOBOXLF DS   OH
      L    R6,WRKSAVR6

```

```

        BAL R14,CLRLINE          Clear printline
        BCT R6,BKLINE            LOOP if more to print
        MVI BDSWORKD,24
        B SKIPSPAC              Skip space line
*****
*      Build and print bottom line                                     *
*      Routine called if BDSWORKD=24+25                             *
*****
SKIPSPAC DS    0H
        BAL R14,CLRLINE          Clear printline
        MVC BOXLINE(10),STARTAG  Move in START tag
        MVC BOXLINE+10(L'BOXLINE-10),BOXLINE Propagate chars
        MVI BDSWORKD,26
        B GETOUT                Printit
*****
*      Space 1 line                                                 *
*      Routine called if BDSWORKD=26                               *
*****
SPACE1L DS    0H
        MVI WRKCC,C'1'          Space 1 line
        LA R5,1
        ST R5,XTPERLEN
        ST R9,XTPERPTR
        OI XTPRCFLG,XTPRCTRN    Translate it to ASCII
        OI XTPRCFLG,XTPRCCC     Assume it has carriage control
        OI XTPRCFLG,XTPRCEXT    Print the record
        MVI BDSWORKD,27
        B GETOUT
*****
*      Epilogue                                                     *
*****
*      Prepare to print the record                                   *
*****
PRINTIT DS    0H
        LA R5,XTPBOUTP          Get record address
        ST R5,XTPERPTR          Put it in XTP area
        SLR R5,R5               Get a zero
        IC R5,XTPBLENG          Get record length
        ST R5,XTPERLEN          Set length to print
        OI XTPRCFLG,XTPRCTRN    Translate it to ASCII
        OI XTPRCFLG,XTPRCCC     Assume it has carriage control
        OI XTPRCFLG,XTPRCEXT    Print the record
        B GETOUT
*****
*      Prepare to print the last record with page eject           *
*      Routine called if BDSWORKD=27                               *
*****
DOLAST OI XTPRCFLG,XTPRLAST     Don't call again
        LA R5,XTPBOUTP          Get record address
        ST R5,XTPERPTR          Put it in XTP area
        MVI XTPBOUTP,X'F1'      Page eject carriage control
        LA R5,1                 Get length of record
        ST R5,XTPERLEN          Set length to print
        OI XTPRCFLG,XTPRCTRN    Translate it to ASCII
        OI XTPRCFLG,XTPRCCC     Assume it has carriage control
        OI XTPRCFLG,XTPRCEXT    Print the record
*****
*      Return to caller                                             *
*****

```

```

GETOUT  DS    OH
        L      R13,XTPSAVE+4      Get address of caller save area
        LM     R14,R12,12(R13)    Restore caller registers
        BR     R14                Return to MVS
*****
*  FUNCTION:                      *
*      This routine will retrieve the JCL keywords:                      *
*          * ADDRESS              *
*          * BUILDING            *
*          * DEPARTMENT          *
*          * TITLE               *
*          * ROOM                *
*          * NAME                *
*      and saves them in storage variables. The keywords                *
*      are retrieved using the Scheduler JCL Facility                    *
*      SWBTUREQ macro.                                                  *
*      Storage located at the end of PRTBUF:                            *
*          * SWBTUREQ Parameter List                                    *
*          * SWBTUREQ Work Area                                        *
*          * Keylist                                                  *
*          * List of SWBTU pointers                                    *
*      Storage located in the area pointed to by OUTTUWS:              *
*          * SWBTUREQ OUTPUT AREA                                    *
*  OTHER CONSIDERATIONS: none                                          *
*****
*      Fill in the SWBTUREQ RETRIEVE parameter list, IEFSJTRP.        *
*****
SJFRETR DS    OH
        XC     SJTRP(SJTRLGTH),SJTRP  Clear parameter list
        MVC    SJTRID,=A(SJTRCID)    Assign function
        MVI    SJTRVERS,SJTRCVER     Assign version number
        LA     R1,SJTRLGTH           Set parameter list
        STH    R1,SJTRLEN            length
*****
*      The Work Area is a 1K work area in PRTBUF that is used as      *
*      a work area by the SWBTUREQ macro.                              *
*****
        LA     R1,SWBTUWS           Set work area
        ST     R1,SJTRSTOR          address
        LA     R1,L'SWBTUWS         Set work area
        STH    R1,SJTRSTSZ          length
        LA     R1,1                Indicate only ONE
        STH    R1,SJTRSWBN          SWBTU pointer
        LA     R1,SBTLAREA          Set SWBTU addr. list
        ST     R1,SJTRSWBA          address (SJTRSBTL)
        LA     R1,6                Indicate six keys in
        STH    R1,SJTRKIDN          key list
        LA     R1,KEYLIST           Set key list
        ST     R1,SJTRKIDL          address
        ST     R1,WRKKYLST          Save keylist address
*****
*      This is a 1K Text Unit Output Area where the parameters are    *
*      returned by the SWBTUREQ macro.                                  *
*****
        LA     R1,OUTTUWS           Set output area
        ST     R1,SJTRAREA          address
        LA     R1,L'OUTTUWS         Set output area
        STH    R1,SJTRSIZE          length
*****

```

```

*          Initialize the SWBTU pointer list          *
*****
        USING SJTRSBTL,R7          Establish
        LA     R7,SBTLAREA          addressability
        XC     SBTAREA(L'SBTAREA),SBTLAREA  Clear SWBTU list
*****
*          Set the address of the area containing the SWBTU data          *
*          from the JCL for the SWBTUREQ macro.                          *
*****
        MVC    SJTRSTUP,XTPSWBTP    Set input SWBTU ptr
        DROP   R7
*****
*          Insert the keys for all of the ESS                          *
*          parameters being used into the key list.                      *
*****
$JTRKEYL DS    OH
        USING SJTRKEYL,R7          Establish
        L      R7,WRKKYLIST          addressability
        XC     SJTRKEYL(6*SJTRKLEN),SJTRKEYL  Clear key list
        LA     R1,DOTITLE            Request
        STH    R1,SJTRKYID+KYLSTTL    TITLE key
        LA     R1,DONAME              Request
        STH    R1,SJTRKYID+KYLSTNM    NAME key
        LA     R1,DOROOM              Request
        STH    R1,SJTRKYID+KYLSTRM    ROOM key
        LA     R1,DOBUILD              Request
        STH    R1,SJTRKYID+KYLSTBL    BUILDING key
        LA     R1,DODEPT              Request
        STH    R1,SJTRKYID+KYLSTDP    DEPARTMENT key
        LA     R1,DOADDRES            Request
        STH    R1,SJTRKYID+KYLSTAD    ADDRESS key
        DROP   R7
*****
*          INVOKE the SWBTUREQ REQUEST=RETRIEVE Macro                    *
*          -----                                                        *
*          Set up R1 to point to a word of storage that                  *
*          contains the address of the parameter list, IEFSJTRP.        *
*****
        MVI    SWBERR,C'N'          Initialize to no
*                                     SWBTUREQ error
        LA     R1,SJTRP              Address of
        ST     R1,WRKPLPTR            the SWBTUREQ
        LA     R1,WRKPLPTR            parameter list
        SWBTUREQ REQUEST=RETRIEVE    INVOKE the Macro
        ST     R15,SWBRXC             Save RC
        ST     R0,SWBRXS              Save RSN
        CH     R15,=H'8'              Does R0 have the RSN?
        BH     *+4+6                  BIY
        MVC    SWBRXS,SJTRREAS        Save RSN for RC < 12
        CH     R15,=H'4'              Check return code
        BL     SJFRETU                GOOD retrieval
*****
*          No keys matched indicates that none of the JCL keywords      *
*          were specified on the OUTPUT JCL. No error message is        *
*          printed. Instead the keywords are just left blank on the     *
*          header sheet.                                                 *
*****
        CLC    SJTRREAS,=A(SJTRNOKY)  No keys matched?
        BE     SJFRETU                YES--Go to TOPLINE
*****

```



```

*      If the SWBTUREQ returns an error, the detail box is      *
*      still printed.  An error message is printed on the      *
*      ADDRESS line.                                           *
*      NOTE: This exit should not receive any errors from      *
*            the SWBTUREQ.  This code is mainly supplied      *
*            for diagnostic purposes when changing the exit.    *
*****
REQ_BAD DS    OH
        CVD    R15,DBLWORD          Convert to pkd dec.
        MVC    FULLWORD,DBLWORD+4    Move packed ret. cde
        UNPK    DBLWORD,FULLWORD      Unpack return code
        MVZ    DBLWORD+7(1),DBLWORD+6 Correct the sign
        MVC    SWBRC(4),DBLWORD+4     Save return code
        L      R1,SJTRREAS            Load reason code
        CVD    R1,DBLWORD             Convert to pkd dec.
        MVC    FULLWORD,DBLWORD+4     Move packed reas. cde
        UNPK    DBLWORD,FULLWORD      Unpack reason code
        MVZ    DBLWORD+7(1),DBLWORD+6 Correct the sign
        MVC    SWBRS(4),DBLWORD+4     Save reason code
        MVI    SWBERR,C'Y'            Indicate a SWBTUREQ
*                                     error occurred and
*                                     a msg is required
SJFRETI DS    OH
        BR     R10                    Back to caller
*****
*      CLRLINE - Clears the separator page line and            *
*               inserts the box frame characters.              *
*      FUNCTION:                                              *
*      This subroutine is called to clear the separator       *
*      page line.  This is necessary so as to avoid           *
*      printing detail box information left over from          *
*      previous processing with the current line.              *
*      LINKAGE:                                              *
*      Accessed via BAL using the label as the entry address   *
*      and register 14 as the return address.                  *
*      INPUT:                                                  *
*               The separator page line/box buffer (BUFWRK).   *
*      OUTPUT:                                                 *
*               The separator page line is cleared and the frame *
*               characters inserted in columns 1 and 78 of the  *
*               detail box.                                     *
*               XTPERPTR - Address of the separator page line   *
*               XTPERLEN - Length of the separator page line   *
*               XTPRCFLG=XTPRCTRN - Translate it to ASCII      *
*               XTPRCFLG=XTPRCCC - Assume it has carriage control *
*               XTPRCFLG=XTPRCEXT - Print the record            *
*      REGISTER USAGE:                                         *
*      REG      VALUE ON ENTRY      VALUE ON EXIT            *
*      R1-R7     N/A                 Unchanged                *
*      R8        N/A                 Destroyed                 *
*      R9-R13    N/A                 Unchanged                 *
*      R14       Return address      Unchanged                 *
*      R15       N/A                 Unchanged                 *
*      RETURN CODES:                                           *
*      None                                              *
*      OTHER CONSIDERATIONS:                                   *
*      None                                              *
*****
CLRLINE DS    OH
        MVI    WRKCC,C' '            Set CC to write

```

```

*                                     1 line
*      MVI    PAGELINE,C' '          Clear 1st printline
*                                     char
*      MVC    PAGELINE+1(L'PAGELINE-1),PAGELINE  Propagate
*                                     spaces
*      MVI    BOXCOL1,C'*'          Insert box
*      MVI    BOXCOL78,C'*'         frame character
*      ST     R9,XTPERPTR           Store record address
*      MVC    XTPERLEN(4),ESSLEN     Get length of record
*      OI     XTPRCFLG,XTPRCTRN      Translate it to ASCII
*      OI     XTPRCFLG,XTPRCCC       Assume it has carriage control
*      OI     XTPRCFLG,XTPRCEXT      Print the record
*      BR     R14                   Return to caller
*****
*      MOVETU - Move the TU text from the TU output area to the
*               detail line DSECT.
*
*      FUNCTION:
*      This subroutine is called to move the TU text from the
*      TU output area to the detail line DSECT area. Since
*      TUs are variable length (up to sixty characters long),
*      the detail line is padded with blanks on the right after
*      the move is performed.
*
*      LINKAGE:
*      Accessed via BAL using the label as the entry address
*      and register 8 as the return address.
*      INPUT:
*      R1 - Address of TU length/parameter pair
*      OUTPUT:
*      The text from the TU parameter is copied into the
*      detail line DSECT area ( BOXINFO ).
*
*      REGISTER USAGE:
*
*      REG      VALUE ON ENTRY      VALUE ON EXIT
*      R0        N/A                 Destroyed
*      R1        Length/Parameter Pair Destroyed
*      R2-R7     N/A                 Unchanged
*      R8        Return Address      Unchanged
*      R9-R13    N/A                 Unchanged
*      R14-R15   N/A                 Destroyed
*
*      RETURN CODES:
*      None
*
*      OTHER CONSIDERATIONS:
*      None
*****
      USING DOCNTFLD,R1
MOVETU  LA     R14,DOCNTPRM          Load TU text address
        LH     R15,DOCNTLEN         Length of TU text
        ICM    R15,B'1000',=CL16' ' Set pad char to blank
        LA     R0,BOXINFO           Set up the
        LA     R1,L'BOXINFO         receiving field
        MVCL   R0,R14              Move the text
        BR     R8                   Return to caller
        DROP   R1
*****
*      Constants and Literals
*****
*      ANSI defined Printer Control Characters:
*
*      Blank = Space 1 line
*      0      = Space 2 line
*      -      = Space 3 line
*      +      = Suppress space

```

```

*          1          = Skip to line 1 on new page          *
*****
JULTABEL DS      OH
           DC      AL1(31),CL3'JAN',AL1(28),CL3'FEB'    JULIAN
           DC      AL1(31),CL3'MAR',AL1(30),CL3'APR'    DAY
           DC      AL1(31),CL3'MAY',AL1(30),CL3'JUN'    AND
           DC      AL1(31),CL3'JUL',AL1(31),CL3'AUG'    MONTH
           DC      AL1(30),CL3'SEP',AL1(31),CL3'OCT'    CONVERSION
           DC      AL1(30),CL3'NOV',AL1(255),CL3'DEC'    TABLE
*****
*          Misc constants and equates          *
*****
XZERO     DC      F'0'
*ESSLLEN  DC      F'133'
* Because of portrait next variable is 78 + 1 (CC)
ESSLLEN   DC      F'79'
ZEROES    DC      X'000000000000'
BLANK     DC      X'40'
AM        DC      CL2'AM'
START     DC      CL5'START'
STARTAG   DC      CL10'**START**'
MAXSEG#   DC      PL8'99999'
PZERO     DC      PL8'00000'
DAYTBL    EQU     0
*****
*          All labels that will appear in the detailbox are listed below*
*****
LBJOBID   DC      CL(L'BOXDESC)'JOBID:'
LBSEGID   DC      CL(L'BOXDESC)'SEGMENT ID:'
LBJOBNAM  DC      CL(L'BOXDESC)'JOB NAME:'
LBUSERID  DC      CL(L'BOXDESC)'USERID:'
LBSYSCL   DC      CL(L'BOXDESC)'SYSOUT CLASS:'
LBOUTGRP  DC      CL(L'BOXDESC)'OUTPUT GROUP:'
LBTITLE   DC      CL(L'BOXDESC)'TITLE:'
LBDEST    DC      CL(L'BOXDESC)'DESTINATION:'
LBNAME    DC      CL(L'BOXDESC)'NAME:'
LBROOM    DC      CL(L'BOXDESC)'ROOM:'
LBBLDG    DC      CL(L'BOXDESC)'BUILDING:'
LBDEPT    DC      CL(L'BOXDESC)'DEPARTMENT:'
LBADDR    DC      CL(L'BOXDESC)'ADDRESS:'
LBPRTIME  DC      CL(L'BOXDESC)'PRINT TIME:'
LBPRDATE  DC      CL(L'BOXDESC)'PRINT DATE:'
LBPRNAME  DC      CL(L'BOXDESC)'PRINTER:'
LBSYSTEM  DC      CL(L'BOXDESC)'SYSTEM ID:'
LBMSG     DC      CL(L'BOXMSG)'EXIT ERROR -- SWBTUREQ MACRO FAILED'
LBRC      DC      CL(L'BOXRCLB)'RETURN CODE:'
LBRS      DC      CL(L'BOXRSLB)'REASON CODE:'
*****
*          Ltorg pool          *
*****
          LTORG
*****
*          Key list equates          *
*****
KYLSTTL   EQU     0*SJTRKLEN          Title key
KYLSTNM   EQU     1*SJTRKLEN          Name key
KYLSTRM   EQU     2*SJTRKLEN          Room key
KYLSTBL   EQU     3*SJTRKLEN          Building key
KYLSTDP   EQU     4*SJTRKLEN          Dept key
KYLSTAD   EQU     5*SJTRKLEN          Address key

```

```

*****
*           Equates for registers 0-15           *
*****
YREGS
*****
*           ANFUXBD2 work buffer                 *
*****
BUFWRK  DSECT
WRKCC   DS    CL1           Carriage control
*WRKDATA DS    CL132        Data line
* Because of portrait next variable is 78
WRKDATA DS    CL78          Data line
WRKARORG DS    OCL1         Work area ORG
BUFPR   EQU   WRKDATA
ORG     WRKDATA
*****
*           Work area                           *
*****
*           Begin separator page line & detail box mapping *
*****
ORG     WRKDATA           Separator page line &
*                               detail box
*PAGELINE DS    OCL132
* Because of portrait next variable is 78
PAGELINE DS    OCL78
* Because of portrait next variable is comment
*           DS    CL26           Left margin
BOXLINE  DS    OCL78
*
*           COL           DESCRIPTION
*           (relative)
BOXCOL1  DS    CL1           1           Frame Character
           DS    CL1           2           Blank
BOXDESC  DS    CL13          3-15        Line Description
           DS    CL1           16         Blank
BOXINFO  DS    CL60          17-76       Line Information
           DS    CL1           77         Blank
BOXCOL78 DS    CL1           78         Frame Character
* Because of portrait next variable is comment
*           DS    CL26           Right margin
ORG     BOXINFO+13
BOXDESC2 DS    CL12          31-42       Line Description
           DS    CL1           43         Blank
BOXINFO2 DS    CL8           44-51       Line Information
           DS    CL5           52-56       Blanks
BOXDESC3 DS    CL11          56-67       Line Description
           DS    CL1           68         Blank
BOXINFO3 DS    CL8           69-76       Line Information
ORG     BOXLINE+55          Segment area
BOXSGLBL DS    C'SEGMENT ID: ' 57-68       Segment ID label
BOXSGINF DS    CL5           69-73       Segment ID Numb.
ORG     BOXINFO             Output Group area
BOXJNAME DS    CL8           18-25       JOE name
BOXGSEP1 DS    CL1           26          Group separator
BOXJID1  DS    OCL5          27-31       JOE ID 1
           DS    CL4           27-30
BOXJZON1 DS    CL1           31          Byte field for MVZ
BOXGSEP2 DS    CL1           32          Group separator
BOXJID2  DS    OCL5          33-37       JOE ID 2
           DS    CL4           33-36
BOXJZON2 DS    CL1           37          Byte field for MVZ

```

	ORG	BOXINFO	Print Time Area
BOXTIME	DS	OCL11	18-28
BOXHR	DS	CL2	18-19 Print Hour
BOXTS1	DS	CL1	20 Separator
BOXMIN	DS	CL2	21-22 Print Minute
BOXTS2	DS	CL1	23 Separator
BOXSEC	DS	CL2	24-25 Print Second
	DS	CL1	26
BOXAMPM	DS	CL2	27-28 AM / PM
	ORG	BOXINFO+26	Print Date Area
BOXDATE	DS	OCL9	28-36 Printing Date
BOXDDD	DS	CL2	28-29 Printing Day
	DS	CL1	30
BOXMMM	DS	CL3	31-33 Printing Month
	DS	CL1	34
BOXYYY	DS	CL4	35-38 Printing Year
	ORG	BOXINFO	SWBTUREQ Error Msg Area
BOXMSG	DS	OCL40	18-57
BOXMSG1	DS	CL40	18-57 Static msg text
	ORG	BOXINFO	
BOXRCLB	DS	CL12	18-29 RETURN CDE label
	DS	CL1	30
BOXRC	DS	CL4	31-34 SWBTUREQ
*			return code
	ORG	BOXINFO	
BOXRSLB	DS	CL12	18-29 REASON CDE label
	DS	CL1	30
BOXRS	DS	CL4	31-34 SWBTUREQ
*			reason code

*	End separator page line & detail box mapping		*

*	Work Areas		*

WRKAREAS	EQU	WRKARORG	
	ORG	WRKARORG	
	DS	OF	
WRKDATE	DS	CL4	Current date - packed
WRKTIME	DS	CL4	Current time - packed
	ORG	WRKTIME	
	DS	CL3	Hours minutes seconds
WRKTH	DS	CL1	Tenths and hundredths
WRKUDATE	DS	CL8	Current date - unpacked
	ORG	WRKUDATE	
WRKMMM	DS	CL3	Month
WRKDD	DS	CL2	Day
WRKYY	DS	CL4	Year
	DS	CL1	Reserved
WRKAMPM	DS	CL2	Current AM/PM
WRKUTIME	DS	CL7	Current time - unpacked
	ORG	WRKUTIME	
WRKHR	DS	CL2	Hour
WRKMIN	DS	CL2	Minute
WRKSEC	DS	CL2	Second
	DS	CL1	Sign byte
	DS	CL1	Reserved
WRKJID1	DS	F	Work area for JOE id 1
WRKJID2	DS	F	Work area for JOE id 2
	DS	OD	Alignment

WRKJIDEC	DS	CL8	Work area JOE id to decimal
WRKJID1Z	DS	CL8	Work area JOE id 1 to zoned
WRKJID2Z	DS	CL8	Work area JOE id 2 to zoned
WRKWORK	DS	CL8	Work area for conversion
WRKJTBL	DS	CL48	Julian conversion table
WRKY2000	DS	CL16	Time SVC work area
TIME1	TIME	LINKAGE=SYSTEM,MF=L	Year 2000 ready
	DS	10F	Alignment
WRKSAVR6	DS	F	Work area save reg 6
WRKSAVR7	DS	F	Work area save reg 7
WRKSAVE	DS	F	Work area save reg
BDSWORKD	DS	8BL1	
BDSWORKE	DS	8BL1	

*	SWBTUREQ declares		*

	DS	OF	
WRKADNUM	DS	H	Number of ADDRESS lines
WRKPLPTR	DS	F	Address of SWBTUREQ
*			parm list
WRKKYLIST	DS	F	Keylist address
*			(SJTRKEYL)
OUTTUWS	DS	CL1024	SWBTUREQ Output Area
FULLWORD	DS	F	Full word work area
DBLWORD	DS	D	Double word work area
SWBRC	DS	F	SWBTUREQ return code
SWBRS	DS	F	SWBTUREQ reason code
SWBRCSX	DS	F	SWBTUREQ return code
SWBRXSX	DS	F	SWBTUREQ reason code
SWBERR	DS	CL1	SWBTUREQ error = "Y"
SWBJBNM	DS	CL8	Job Name Save
	DS	H	
SWBUIDN	DS	CL8	User ID Save
	DS	H	
	DS	OF	
KEYLIST	DS	CL64	SJTRKEYL Area
SBTLAREA	DS	CL16	SJTRSBTL Area
SWBTUWS	DS	CL1024	SWBTUREQ Work Area
TESTLEN	EQU	*-BUFWRK	
BUFWRKL	EQU	3000	

*	Even though the SWBTUREQ parameter list is invoked		*
*	with DSECT=NO, there are still DSECTS in the macro.		*
*	Therefore do NOT attempt to add DCs or DSs after this		*
*	macro that are not part of a DSECT.		*

*	Total length of the work buffer (BUFWRK) is BUFWRKL.		*
*	This includes also DC and DS in the next macro: IEFSJTRP		*
*	IEFSJTRP consists (with DSECT=NO) of the parm list		*
*	and two DSECTS: SJTRSBTL and SJTRKEYL.		*
*	The SWBTUREQ parm list must be part of the work buffer (BUFWRK)		*

	IEFSJTRP DSECT=NO		SWBTUREQ parm. list

*	End of ANFUXBD2 work buffer		*

	TITLE 'Dsects'		
	IEFDOTUM		Text unit mapping
	IEFDOKEY		OUTPUT key mapping
	IEFSJTRC		SWBTUREQ return codes

ANFUEXTP	
IEFJMR	
IAZJSPA LIST=YES	
CVT DSECT=YES	Required for SWBTUREQ
IEFJESCT	Required for SWBTUREQ
END ANFUXBD2	End of ANFUXBD2 module

Sample header page from ANFUXBD2

Example 9-1 Sample header page from the supplied exit in this appendix

```

VV  VV  AAAAAA  IIIIII  N    NN  IIIIII
VV  VV  AAAAAAAA  IIIIII  NN   NN  IIIIII
VV  VV  AA   AA   II   NN   NN   II
VV  VV  AA   AA   II   NNN  NN   II
VV  VV  AA   AA   II   NNNN  NN   II
VV  VV  AAAAAAAA  II   NN  N  NN   II
VV  VV  AAAAAAAA  II   NN  NN  NN   II
VV  VV  AA   AA   II   NN  N  NN   II
VV  VV  AA   AA   II   NN  NNNN  II
VV  VV  AA   AA   II   NN   NNN  II
VVVV  AA   AA  IIIIII  NN   NN  IIIIII
VV  AA   AA  IIIIII  NN   N  IIIIII

VV  VV  AAAAAA  IIIIII  N    NN  IIIIII  KK   K  AAAAAA  IIIIII
VV  VV  AAAAAAAA  IIIIII  NN   NN  IIIIII  KK   KK  AAAAAAAA  IIIIII
VV  VV  AA   AA   II   NN   NN   II   KK  KK  AA   AA   II
VV  VV  AA   AA   II   NNN  NN   II   KK  KK  AA   AA   II
VV  VV  AA   AA   II   NNNN  NN   II   KK  KK  AA   AA   II
VV  VV  AAAAAAAA  II   NN  N  NN   II   KKKK  AAAAAAAA  II
VV  VV  AAAAAAAA  II   NN  NN  NN   II   KKKK  AAAAAAAA  II
VV  VV  AA   AA   II   NN  N  NN   II   KK  KK  AA   AA   II
VV  VV  AA   AA   II   NN  NNNN  II   KK  KK  AA   AA   II
VV  VV  AA   AA   II   NN   NNN  II   KK  KK  AA   AA   II
VVVV  AA   AA  IIIIII  NN   NN  IIIIII  KK   KK  AA   AA  IIIIII
VV  AA   AA  IIIIII  NN   N  IIIIII  KK   K  AA   AA  IIIIII

CCCCC  UU   UU  BBBBBB  EEEEEEE  22222  EEEEEEE  000000  11
CCCCCCC  UU   UU  BBBBBB  EEEEEEE  2222222  EEEEEEE  00000000  111
CC   CC  UU   UU  BB   BB  EE   22   22  EE   00   00  1111
CC   UU   UU  BB   BB  EE   22   EE   00   00  11
CC   UU   UU  BB   BB  EE   22   EE   00   0  00  11
CC   UU   UU  BBBBBB  EEEEEEE  22   EEEEEEE  00  0  00  11
CC   UU   UU  BBBBBB  EEEEEEE  22   EEEEEEE  00  0  00  11
CC   UU   UU  BB   BB  EE   22   EE   00  0  00  11
CC   UU   UU  BB   BB  EE   22   EE   000  00  11
CC   CC  UU   UU  BB   BB  EE   22   EE   000  00  11
CCCCCCC  UUUUUUU  BBBBBB  EEEEEEE  2222222  EEEEEEE  00000000  1111111
CCCCC  UUUUU  BBBBBB  EEEEEEE  2222222  EEEEEEE  000000  1111111

**START*****START*****START*****START*****START*****START*****START*
*
* JOB NAME:      VAINI      JOBID:      JOB14825      USERID:      VAINI      *
* SYSOUT CLASS:  J          DESTINATION: PASSTHRU      PRINTER:      PRTWAY      *
* PRINT TIME:    08:15:29 PM  PRINT DATE:  21 MAY 2001  SYSTEM ID:  SC43      *
*
* OUTPUT GROUP:      .00000.00000      *
*
* TITLE:      *
* NAME:      Vainikainen J      *
* ROOM:      Cube2E01      *
* BUILDING:    008      *
* DEPARTMENT:  ITS0      *
* ADDRESS:     POUGHKEEPSIE      *
*
**START*****START*****START*****START*****START*****START*****START*

```



Sample ANFUXRC2 user exits

This appendix provides two samples to replace ANFUXRC2, the Record exit.

It contains the following:

- ▶ A sample user exit to do a page eject
- ▶ A sample exit to replace every special character with PCL5 strings

Sample user exit

Example: C-1 Sample to page eject when output is duplexed

```

ANFUXRC2 TITLE 'Record Exit - NULL First Page First EJECT'
***** START OF SPECIFICATIONS *****
* MODULE NAME = ANFUXRC2 *
* DESCRIPTIVE NAME = Record exit *
* COPYRIGHT= 5695040 (c) COPYRIGHT IBM CORPORATION 2000 *
*-----*
* This exit is should be used with the sample job header exit *
* ANFUXBD2. Sometimes, when the output is duplexed, the first *
* data record is written on the back side of the first page. *
* This record exit attempts to fix the problem... *
* *
* Logic: *
* - Initialization stuff *
* - Get working storage which will be passed between calls *
* - Remove the first FormFeed or ASA CC 1 if found on two *
* first records in a data set *
* *
* LINKAGE = *
* LOAD ANFUXRC2 *
* CALL ANFUXRC2(addr(ANFUExTP)) *
* calling module passes address of Common Parameter *
* area. *
***** END OF SPECIFICATIONS *****
ANFUXRC2 CSECT , Establish the csect
ANFUXRC2 AMODE ANY It addresses all storage
ANFUXRC2 RMODE ANY It can reside anywhere
YREGS
SAVE (14,12),,ANFUXRC2_&SYSTIME._&SYSTIME
LR R12,R15 Move base register to R12
USING ANFUXRC2,R12 R12 is base reg for program code
L R4,0(R1) Get address of parm area
USING ANFUExTP,R4 R4 is base for parm area
ST R13,XTPSAVE+4 chain save areas
LA R15,XTPSAVE get my save area
ST R15,8(R13) point callers save area to it
LR R13,R15 point R13 to my save area
*-----*
* Get storage for work area (1 time only!) *
*-----*
ICM R3,15,XTPWORK1 get working storage addr
BNZ MAINLINE yes, skip this
LA R3,RECSTGL Get length of DSECT
GETMAIN RU,LV=(2),LOC=ANY Get program storage area
ST R1,XTPWORK1 save my work area
LR R3,R1
MAINLINE DS OH
USING RECSTG,R3
L R5,XTPJSPAP Point at JSPA
USING IAZJSPA,R5
CLC RECDSNAM,XTPDSNAM Same data set as before?
BNE NEWDS BIN
CLC RECAJBNM(L'RECAJBNM+L'RECAJBID+L'RECADEVN),JSPAJBNM
BNE NEWDS Same job? - BIN
LH R0,RECCNT Enough..
CH R0,=H'1' .records..
BH GOON ..processed....

```

```

NEWDS      B      NEWDSS
           DS      OH
           XC      RECCNT,RECCNT      Indicate new data set
           MVC      RECDSNAM,XTPDSNAM  Set data set name
           MVC      RECAJBNM(L'RECAJBNM+L'RECAJBID+L'RECADEVN),JSPAJBNM
NEWDSS     DS      OH
           ICM      R7,15,XTPPRPTR      Point at record
           BZ      GOON                  None? - BIY
           TM      XTPDSFLG,XTPDSCC      CCs?
           BO      NEWDSA
           CLC      =X'0DOC',0(R7)      CR/FF?
           BNE      GOON                  BIN
           MVI      1(R7),X'00'          NULL FF
           B        GOON
NEWDSA     DS      OH
           CLI      0(R7),C'1'          ASA CH 1?
           BNE      GOON                  BIN
           MVI      0(R7),C' '          NOP CH 1?
*-----*
*      Replace original with new record      *
*-----*
GOON       DS      OH
*-----*
*      No change, use original record        *
*-----*
           OI      XTPRCFLG,XTPRCORG      add original record
           OI      XTPRCFLG,XTPRLAST      done with this record
           B        GETOUT
*-----*
*      Return to caller                      *
*-----*
GETOUT     DS      OH
           LH      R1,RECCNT              Increment..
           LA      R1,1(R1)               .record count..
           STH      R1,RECCNT              ..for this data set
           L        R13,XTPSAVE+4          Get address of caller's save area
           RETURN   (14,12),RC=0           Restore caller's registers
           DROP     R3,R4,R5              RECSTG, XTP, JSPA
*-----*
*      Constants and Literals                *
*-----*
           LTORG
*-----*
*      Work area                            *
*-----*
RECSTG     DSECT
*           XTP      data set name
RECDSNAM   DS      CL53                  DSNAME
*           JSPA     field
RECAJBNM   DS      CL8                   JOB NAME
RECAJBID   DS      CL8                   JOB ID
RECADEVN   DS      CL8                   DEVICE NAME
*
RECCNT     DS      H
           DS      OD
RECSTGL    EQU      *-RECSTG              Length of program storage
           ANFUEXTP
           IAZJSPA LIST=YES
*-----*
*      End of Program Storage                *

```

```

*-----*
END      ANFUXRC2      End of ANFUXRC2 module

```

Sample ANFUCRC2 record exit

ANFUXRC2 is a (working) example of setting up the Language Environment 370 (LE/370) to be able to call customer HLL (COBOL) routines. This example replaces every special character with PCL5 strings.

Example: C-2

```

ANFUXRC2 TITLE 'Record Exit'
***** START OF SPECIFICATIONS *****
*
* MODULE NAME = ANFUXRC2
*
* DESCRIPTIVE NAME = Record exit
*
* COPYRIGHT= 5695040 (c) COPYRIGHT IBM CORPORATION 2000
*
* STATUS = VERSION 1, RELEASE 1, LEVEL 0
*
*-----*
*
* Logic:
*
* - Initialization stuff
* - Get our working storage which will be passed between calls for
*   one data set, one time only...
* - Initialization of the Language Environment (LE/370) necessary for
*   linking to COBOL program: PRWAY, this is the interface to DIADIA
* - If OK ----> NO: - Give errormsg in syslog with return code
*   YES:
*     | - Add original record
*     | - Exit
*   |
* - Check FORMS from JCL:
*   STD? --> YES: - Add original record
*     | - Exit
*   |
*   +--> NO: DIAPCL5? --> YES: - Set TYPE=PCL5
*     | - Proceed at @2
*     |
*     +--> NO: DIAOVLY? --> YES: - Set TYPE=OVLY
*       | - Proceed at @2
*       |
*       +--> NO: - Add org. record
*         - Exit
* - @2
* - Get working storage for passing to COBOL PRWAY program, one time
*   only ...
* - Setup parmlist for PRWAY:
*   1 fullword: @ original record
*   2 fullword: @ of length original record
*   3 fullword: @ new record
*   4 fullword: @ of length new record
*   5 fullword: @ of Call type: PCL5 or OVLY
*   6 fullword: @ of Status: 1=add, 2=replace, 3=delete, 4=call again
*   7 fullword: @ of @ of COBOL workarea

```

```

*
* - Switch to problem state for calling to LE/370
* - Link to PRWAY
* - On return for PRWAY check STATUS:
*   If 1 ADD --> YES: - Add new record and Exit
*     2 REPLACE --> YES: - Replace with new record and Exit
*     3 DELETE --> YES: - Delete the record and Exit
*     4 Call Again --> YES: - Replace with new record
*       |
*       | - ask Printway to call again for record
*       | - Set STATUS = CALL AGAIN
*       | - Exit
*     +--> NO: Error invalid STATUS returned for PRWAY
*       - Give error msg in SYSLOG
*       - Add original record
*       - Exit
*
*-----*
*
* NOTES = none
* DEPENDENCIES = none
*
* MODULE TYPE = PROCEDURE
*   PROCESSOR = Assembler
*   ATTRIBUTES =
*     REENTRANT
*     AMODE(31)
*     RMODE(ANY)
*
* PATCH LABEL = none
*
* ENTRY POINT =
*   ANFUXRC2
*
* LINKAGE =
*   LOAD ANFUXRC2
*   CALL ANFUXRC2(addr(ANFUEXTP))
*   calling module passes address of Common Parameter
*   area.
*
***** END OF SPECIFICATIONS *****
*
ANFUXRC2 CSECT ,           Establish the csect
ANFUXRC2 AMODE ANY         It addresses all storage
ANFUXRC2 RMODE ANY         It can reside anywhere
                          USING ANFUXRC2,R15      Establish temporary addressability X
                                                  to module
      B      RECINIT       Branch around copyright information
ANFUXRC2 MODID BR=NO       Eyecatcher info
      DC     C' This module has been developed '
      DC     C'by Jan de Rover, IBM Printing '
      DC     C'Systems Division (PSD) Amsterdam '
      DC     C'in the Netherlands '
      DC     C'This code is not considered Type 1 '
      DC     C'code of IBM. Also it has not been '
      DC     C'submitted to any formal IBM test. '
      DC     C'Distribution is on an "as is" '
      DC     C'basis without any warranty either '
      DC     C'expressed or implied. '
      PRINT ON,NOGEN
*

```

```

*-----*
*   ANFUXRC2 Module entry point   *
*-----*
*
RECINIT DS    OH
        STM   R14,R12,12(R13)    Save callers registers
        LR    R11,R15            Move base register to R11
        DROP  R15                Done with this register
        USING ANFUXRC2,R11       R11 is base reg for program code
        L     R4,0(R1)           Get address of parm area
        USING ANFUEXTP,R4        R4 is base for parm area
        ST    R13,XTPSAVE+4      chain save areas
        LA    R15,XTPSAVE        get my save area
        ST    R15,8(R13)         point callers save area to it
        LR    R13,R15           point R13 to my save area
*
*-----*
*   Get storage for work area (1 time only!)   *
*-----*
*
        SLR   R3,R3              get a zero
        L     R2,XTPWORK1        get working storage addr
        CR    R2,R3              has work area been gotten?
        BNE   INITLE             yes, skip this
        LA    R2,RECSTGL         Get length of DSECT
        GETMAIN RU,LV=(2),LOC=ANY Get program storage area
RECGOTIT DS   OH
        ST    R1,XTPWORK1        save my work area
        LR    R2,R1
*
INITLE  DS    OH
        USING RECSTG,R2
*
*-----*
*   Use Language Environment (LE/370) to link to Cobol program   *
*-----*
        CEELRR ACTION=INIT
*
        C     R15,=F'4'          Warning is accepted
        BNH   LEOKE              If Higher RC=4 then leave
*
*-----*
*   RC not 0 or 4 write message   *
*-----*
*
        MVC   MSG,MSG1           Move in message
        CVD   R15,RCDEC           convert RC to decimal
        UNPK  RCCODE,RCDEC+5(3)    unpack it
        OI    RCCODE+7,X'F0'      make it printable
        MVC   MSGRC,RCCODE+4      and move to WTO message
        WTO   MF=(E,MSG)
*
        B     NOCHANGE           Use original record and leave
*
*-----*
*   Language Environment oke   *
*   Check if we have to do the DIADIA stuff   *
*-----*
*
LEOKE   DS    OH

```

```

        CLC   XTPFORMS,=C'STD      '
        BE    NOCHANGE              If forms=STD then no call(DIADIA)
        CLC   XTPFORMS,=C'DIAPCL5  '
        BNE   CHKFRMS2
        MVC   RCTYPE,=C'PCL5'
        B     LEOKE1
CHKFRMS2 CLC   XTPFORMS,=C'DIAOVLY '
        BNE   NOCHANGE
        MVC   RCTYPE,=C'OVLY'
        B     LEOKE1

*
*-----*
*           Get working storage for COBOL PRWAY routine (1 time only!) *
*-----*
*
LEOKE1  DS     OH
        SLR   R7,R7                get a zero
        L     R8,XTPWORK2          get working storage addr
        CR    R7,R8                has work area been gotten?
        BNE   CBWRK1              yes, skip this
        L     R7,CBWRKLN           Get length of COBOL workarea
        GETMAIN RU,LV=(7),LOC=ANY  Get storage area
        ST    R1,XTPWORK2          save work area
CBWRK1  DS     OH
        L     R1,XTPPRPTR          Get address of org record
        ST    R1,RCORGPTR          Save it in parm list
        LA    R1,XTPPRLEN          Get Length of org record
        ST    R1,RCORGLPT          save @ Length of org record
        LA    R1,RECWORKB          Get address of new record
        ST    R1,RCNEWPTR          Save it in parm list
        LA    R1,RCNEWLEN          Get @ of new record length
        ST    R1,RCNEWLPT          Save it in parm list
        LA    R1,RCTYPE            Get address of type
        ST    R1,RCTYPEPT          Save it in parm list
        LA    R1,RCSTATUS          Get address of type
        ST    R1,RCSTATPT          Save it in parm list
        L     R1,XTPWORK2          Get COBOL workarea pointer
        ST    R1,CBWRKPTR          Save it in pointer field
        LA    R1,CBWRKPTR          Get address of pointer field
        ST    R1,RCWRKPTR          Save it in parm list
        SLR   R1,R1                get a zero to init next parms
        ST    R1,RCNEWLEN          Init new length rec length
        CLC   RCSTATUS,RCCALL      Previous call asked for 2e call?
        BE    LEAVEST              if yes, don't blank status
        ST    R1,RCSTATUS          Init status
LEAVEST MODESET MODE=PROB          Make it problem state for LE/370
        LA    R1,RECPARMS          Get address of parm area
        LINK  EP=PRWAY             Link to DIA interface program
        MODESET MODE=SUP           return to original mode
        CLC   RCSTATUS,RCDELETE    Delete record?
        BE    GETOUT
        CLC   RCSTATUS,RCREPLAC    Replace record?
        BE    REPLACE
        CLC   RCSTATUS,RCCALL      Call PRWAY again?
        BE    CALAGAIN
        CLC   RCSTATUS,RCADD        Add record?
        BE    ADDREC

*
*-----*
*           Invalid status returned from PRWAY *
*-----*

```

```

*-----*
*
MVC    MSG,MSG1           Move in message
L      R15,RCSTATUS
CVD    R15,RCDEC          convert RC to decimal
UNPK   RCCODE,RCDEC+5(3)  unpack it
OI     RCCODE+7,X'F0'     make it printable
MVC    MSGRC,RCCODE+4     and move to WTO message
WTO    MF=(E,MSG)
*
B      NOCHANGE           Use original record and leave
*
*-----*
*      Add new record
*-----*
*
ADDREC MVI    XTPRCFLG,X'00'    all flags off
      OI     XTPRCFLG,XTPRCEXT  add new record
      OI     XTPRCFLG,XTPRLAST  done with this record
      OI     XTPRCFLG,XTPRCTRN  translate it
      LA     R6,RECWORKB        get exit's buffer
      ST     R6,XTPERPTR        point parm area at it
      L      R6,RCNEWLEN        Get new record length
      ST     R6,XTPERLEN        point parm area at it
      B      GETOUT
*
*-----*
*      No change, use original record
*-----*
*
NOCHANGE MVI    XTPRCFLG,X'00'    all flags off
      OI     XTPRCFLG,XTPRCORG  add original record
      OI     XTPRCFLG,XTPRLAST  done with this record
      B      GETOUT
*
*-----*
*      Replace original with new record
*-----*
*
REPLACE MVI    XTPRCFLG,X'00'    all flags off
      OI     XTPRCFLG,XTPRCORG  add original record
      OI     XTPRCFLG,XTPRLAST  done with this record
      LA     R6,RECWORKB        get exit's buffer
      ST     R6,XTPPRPTR        point parm area at it
      L      R6,RCNEWLEN        Get new record length
      ST     R6,XTPPRLEN        point parm area at it
      B      GETOUT
*
*-----*
*      Replace original with new one and call PRWAY again
*-----*
*
CALAGAIN MVI    XTPRCFLG,X'00'    all flags off
      OI     XTPRCFLG,XTPRCORG  add original record
      LA     R6,RECWORKB        get exit's buffer
      ST     R6,XTPPRPTR        point parm area at it
      L      R6,RCNEWLEN        Get new record length
      ST     R6,XTPPRLEN        point parm area at it
      B      GETOUT
*

```



```

*-----*
*           Return to caller                               *
*-----*
GETOUT  DS    OH
        L      R13,XTPSAVE+4      Get address of caller's save area
        LM     R14,R12,12(R13)    Restore caller's registers
        BR     R14                Return to MVS
*
*-----*
*           Constants and Literals                         *
*-----*
*
MSG1    DS    OF
MSG11S  DC    AL2(MSG11L)          Test length
        DC    B'1000000001000000' MCSFLAGS
        DC    C'ANFUXRC2 Unexpected RC from LE/370 '
        DC    C'init lib retention OR invalid '
MSG11E  DC    B'0000000000000000' Descriptor codes
        DC    B'0000000000100000' Routing codes
        DC    XL2'2000'           Line type
        DC    X'00'              Area id
        DC    AL1(2)             Total number of lines
MSG12S  DC    AL2(MSG12L)          Message length
        DC    XL2'3000'           Line type
        DC    C'status from PRWAY. RC = '
MSG12I  DC    C'xxx'
        DC    C'.'
*
MSG11L  EQU  MSG11E-MSG11S          Length of wto line 1
MSG12L  EQU  *-MSG12S              Length of wto line 2
MSG12IO EQU  MSG12I-MSG11S          Offset to insert for RC
MSG1LEN EQU  *-MSG1               Total wto message length
*
RCADD   DC    F'1'                add new record
RCREPLAC DC F'2'                Replace with new record
RCDELETE DC F'3'                delete original record
RCCALL  DC    F'4'                call again
CBWRKLN DC    F'4095'            Length of COBOL workarea
        LTORG
*
*-----*
*           Work area                                       *
*-----*
*
RECSTG  DSECT
RCNEWLEN DS    F
RCTYPE   DS    F
RCSTATUS DS    F
RCWORKKB DS    CL1500            area for output record
        DS    OD
*
*-----*
*           Parm area for link to PRWAY                     *
*-----*
*
RECPARMS EQU  *
RCORGPTR DS    F                @ to original record
RCORGLPT DS    F                @ to length of original record
RCNEWPTR  DS    F                @ to new record
RCNEWLPT  DS    F                @ to length of new record

```

```

RCTYPEPT DS    F           @ to type of this call
RCSTATPT DS    F           @ to status returned by PRWAY
RCWRKPTR DS    F           @ to @ of the COBOL workarea
CBWRKPTR DS    F           @ to the COBOL workarea
*
*-----*
*           Message variables                               *
*-----*
*
MSG        DS    CL(MSG1LEN)
           DS    OD           Alignment
RCDEC      DS    PL15         To convert Retcode to decimal
           DS    OD           Alignment
RCCODE     DS    CL8          To make Retcode printable
WORK1      DS    CL4
WORK2      DS    CL10
PARM1      DS    F
* Message insert mapping
MSGMAP     ORG    MSG
           DS    CL(MSG12IO)
MSGRC      DS    CL4
*
RECSTGL    EQU    *-RECSTG      Length of program storage
           ANFUEXTP
*-----*
*           End of Program Storage                           *
*-----*
*
R0         EQU    0             Register
R1         EQU    1             equates
R2         EQU    2
R3         EQU    3
R4         EQU    4
R5         EQU    5
R6         EQU    6
R7         EQU    7
R8         EQU    8
R9         EQU    9
R10        EQU    10            used by LE/370
R11        EQU    11
R12        EQU    12
R13        EQU    13
R14        EQU    14
R15        EQU    15
           END    ANFUXRC2      End of ANFUXRC2 module

```

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

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

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

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

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

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

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

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 290.

Other resources

These publications are also relevant as further information sources:

- ▶ *z/OS Infoprint Server User's Guide*, S544-5746
- ▶ *z/OS Infoprint Server Customization*, S544-5744
- ▶ *z/OS Infoprint Server Operation and Administration*, S544-5745
- ▶ *z/OS Infoprint Server Messages and Diagnosis*, G544-5747
- ▶ *z/OS UNIX System Services Planning*, GA22-7800
- ▶ *z/OS UNIX System Services Command Reference*, SA22-7802
- ▶ *z/OS JES2 Initialization and Tuning Reference*, SA22-7533
- ▶ *z/OS JES3 Initialization and Tuning Reference*, SA22-7550
- ▶ *z/OS JES2 Initialization and Tuning Guide*, SA22-7532
- ▶ *z/OS JES3 Initialization and Tuning Guide*, SA22-7549
- ▶ *z/OS MVS JCL Reference*, SA22-7597
- ▶ *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- ▶ *z/OS MVS Product Management*, SA22-7603 (softcopy only)
- ▶ *z/OS C/C++ Programming Guide*, SC09-4765
- ▶ *z/OS Communications Server IP Configuration Guide*, SC31-8775
- ▶ *z/OS IBM Communications Server: SNA Resource Definition Reference*, SC31-8778
- ▶ *z/OS Communications Server: IP Configuration Reference*, SC31-8776
- ▶ *z/OS SecureWay Security Server RACF Security Administrator's Guide*, SA22-7683
- ▶ *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394 (softcopy only)
- ▶ *z/OS Language Environment Programming Guide*, SA22-7561
- ▶ *PSF for OS/390: Customization*, S544-5622
- ▶ *PSF for OS/390: User's Guide*, S544-5630
- ▶ *PSF for OS/390: Diagnosis*, G544-5623
- ▶ *AFP: Programming Guide and Line Data Reference*, S544-3884

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Windows Client programs:
<http://www.printers.ibm.com/R5PSC.NSF/Web/download/>
- ▶ IPP prototype download:
<http://www.printers.ibm.com/R5PSC.NSF/Web/ipp/>

How to get IBM Redbooks

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web Site

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web Site for information about all the CD-ROMs offered, updates and formats.

Index

Symbols

\$HOME/.profile 26, 28
/etc/Printsrv 32, 234
/etc/Printsrv/aopxfd.conf 186
/etc/profile 26, 27, 29, 36, 37, 45, 46
/etc/rc 49
/u/userid 26
/usr/lpp/Printsrv/bin 37, 171, 234
/usr/lpp/Printsrv/lib 235
/usr/lpp/Printsrv/samples 234, 238
/usr/lpp/Printsrv/samples/aopxfd.conf 200
/usr/lpp/Printsrv/samples/sample.pidl 222
/var/Printsrv 32
/var/Printsrv/jestoken.db 227
/var/Printsrv/master.db 227

A

AFP 230, 231
AFP Printer Drive 16
AFP Printer Driver for Windows 144
AFP Viewer plug-in 17
ANF.QUEUE 62
ANFDEAL 62
ANFMFILE initialization program 107
ANFMFILE program 62
AOP.SAOPEXEC 39
AOP_COLOR 200
AOP_RESOURCE_PATH 191
AOPADMIN group 225
AOPBATCH program 223
AOPBATCH utility program 54
aopd.conf 33, 178
aopfiltr.so 54, 72
AOPINIT 40
AOPINIT exec 39
AOPJAUTH member 47
aopoms 234, 235, 240
AOPPIDU member 223
AOPPRINT 9
AOPPRINT JCL procedure 50, 164, 183
aopsapd 234, 235
aopsapd.conf 33, 235
aopsetup command 27
aopsetup shell script 27
aopstart command 28, 29, 30, 42, 44, 46, 49, 178
AOPSTART JCL procedure 206
AOPSTART procedure 46
aopstop command 28, 44, 48
aopxfd.conf 33, 178, 179, 184, 187, 190, 208
AOPXFD_CONF 186
APAR 234
APIMIJCL program 129
ASCII printers 72
ascii-codepage 34

ASSIZEMAX 46, 189

automount facility 27

automount policy 28

B

BIND keyword 43

blank truncation 51

BPXBATCH utility program 223, 224

BPXPRMxx member 46

C

callback demon 234, 240

cancel command 170

CEE.SCEERUN 110

chmod command 181

chown command 26, 181

configuration files 33

create command 218

cron daemon 227

crontab command 226

D

DATASETPREFIX 94

DCE 159

DFS 159

Direct socket printer 7

direct sockets protocol 65

Distributed File Service 159

E

echo command 36

environment variables 36

export command 220, 225

extattr command 29

F

font-mapping 209

force-create command 218

FSA 95

FSS 95

FSSDATA parameter 118

functional subsystem 95

functional subsystem application 95

I

IBM 3130 232

ICHRIN03 28

IFAPRDxx member 177

Infoprint Server configuration files 33

Infoprint Server ISPF panels 60

Infoprint Server Transforms 72, 176

- infoprint.oms 238
- Internet Printing Protocol (IPP) 42, 151
- Internet Printing Protocol daemon 27
- IP PrintWay 230, 232
- IP PrintWay exits 112
- IP PrintWay printers 6
- IP PrintWay procedure 114
- IP PrintWay runtime options 111
- IPP client 151, 153
- IPP protocol 65, 110, 153
- IPP protocol printers 6
- IPP server 151
- IPP Server daemon 45

J

- Java 27, 29, 47
- JES blank truncation support 50
- JES page definitions 205
- job 33
- job-prefix 33, 135

L

- Language Environment run-time library 110
- LE environment 111
- LIBPATH 235
- LOMS 235, 239, 240
- lp command 170
- LPD compatibility filter 54
- lpr 233
- lpr command 150
- LPR protocol 65
- lpstat command 170
- ls command 181

M

- MAXASSIZE 46, 189
- message-log data set 62, 107
- mkdir command 26
- mount command 26

N

- NetSpool customization 124
- NetSpool message-log data set 128

O

- OBEYFILE command 43
- object class 217
- OEDIT command 179
- oedit command 36
- OMS 234, 240
- OS/390 159
- OS/390 Printer Port Monitor 16, 53
- OTF 230, 231, 237
- OUTCLASS statement 98
- OUTPUT JCL statement 116

P

- PCL 230, 231, 232
- PCL to AFP transform 186
- pcl2afp command 186
- PDF 230, 231
- PIDU 61
- pidu command 214, 223
- PIDU utility 91
- PORT statement 43
- preload.ps resource file 193
- Print Interface 9, 50
- Print Interface LPD 42
- PRINTDEV statement 85
- Printer Inventory 8, 33, 60
- Printer Inventory Definition Utility 61
- Printer Inventory Definition Utility (PIDU) 60
- Printer Inventory Definition Utility program 214
- Printer Inventory ISPF application 61
- Printer Inventory Manager 22
- PRTnnnnn statement 97
- PS 230, 231, 232
- PSF 230, 231
- PSF JCL procedure 85

R

- RCF 1179 2
- Redbooks Web Site 290
 - Contact us x
- remote transform support 177
- Request for Comments (RFC) 2
- RFC 1179 65, 114
- ROMS 239, 240

S

- SAP R/3
 - transaction sp01 238
 - transaction spad 238
- SCEERUN data set 110
- Server Message Block 159
- Server Message Block (SMB) 42
- SETOMVS command 46, 189
- SMB 159
- SNMP agent 36
- SPOOLDEF statement 98
- start-daemons 33
- start-daemons 42
- STDENV 56
- STDERR 56
- stderr 215
- STDIN 56
- STDOUT 56
- stdout 215
- STEPLIB statement 110
- SYS1.SAMPLIB 112
- SYS1.SAMPLIB members
 - ANFDEAL 62
 - ANFMIJCL 62
 - ANFQINIT 62

T

- TCP/IP considerations 94
- TCP/IP LP 42
- TCP/IP transmission protocols 94
- TCP/IP-attached printers 42, 94
- TCPIPJOBNAME 94
- transaction sp01 238
- transaction spad 238
- transform DLL 186
- Transform Manager 188
- Transform Manager Daemon (AOPXFD) 45
- transform resource files 191
- Transmission queue 114
- transmission-queue data set 62, 103

U

- UserInit resource file 192
- USERPORTS LPR option 113
- Using 215
- using the lpr command 150

V

- VBUILD definition statement 125
- VTAM APPL definition statement 125
- VTAM protocol 65
- VTAM protocol printers 6
- VTAM-controlled printers 94



Infoprint Server for z/OS Implementation

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



Redbooks

z/OS Infoprint Server Implementation

Customize Infoprint Server

Determine which components and functions you need

Sample user exits are provided

This IBM Redbook describes how to customize the Infoprint Server, an element of z/OS Version 1 Release 1, and Infoprint Server Transforms Version 1 Release 1 Modification Level 1, a separate IBM program product (5697-F51).

It is intended for system programmers and administrators responsible for customizing Infoprint Server for their installation. The reader should be familiar with z/OS UNIX System Services, TCP/IP, the Job Entry Subsystem (JES), and z/OS Job Control Language (JCL).

In this book we introduce Infoprint Server, describe how the components fit into your system, list the functions each component provides, and identify the components you need to customize in order to use each function.

We then describe how to customize each component of Infoprint Server. Sample user exits are provided as an aid in customizing your own exits.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6234-00

ISBN 0738421634